User's Manual

32

RX634 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX600 Series

Renesas Electronics
www.renesas.com

Rev.1.00 Feb 2015

Notice

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

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

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

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

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

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

(2012.4)

NOTES FOR CMOS DEVICES

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.
This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

When designing an application system that includes this LSI, take all points to note into account.
Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for the RX634 Group. Before using any of the documents, please visit our website to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Datasheet	Overview of hardware and electrical characteristics	RX634 Group Datasheet	R01DS0255EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX634 Group User's Manual: Hardware	This document
User's Manual: Software	Detailed descriptions of the CPU and instruction set	RX Family User's Manual: Software	R01US0032EJ
Application Note	Examples of applications and sample programs	—	—
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.	—	—

2. Description of Registers

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
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Contents

R01UH0495EJ0100
Rev.1.00
Feb 25, 2015

54 MHz 32-bit RX MCU with FPU, 90 DMIPS, up to 2-Mbyte flash memory, 12-bit ADC, 10-bit DAC, ELC, MPC, CEC transmission/reception, remote control signal reception

Features

■ 32-bit RX CPU core

- Max. operating frequency: 54 MHz
  Capable of 90 DMIPS in operation at 54 MHz
- Two types of multiply-and-accumulation unit (between memories and between registers)
- 32-bit multiplier (fastest instruction execution takes one CPU clock cycle)
- Divider (fastest instruction execution takes two CPU clock cycles)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- Supports the Memory Protection Unit (MPU)
- On-chip debugging circuit

■ Low power design and architecture

- Operation from a single 2.7 to 3.6 V or 4.0 to 5.5 V supply
- Four low power consumption modes

■ On-chip main flash memory, no wait states

- 54-MHz operation, 18.5-ns read cycle
- 1 to 2 Mbytes supported
- User code is programmable by on-board

■ On-chip data flash memory

- 32 Kbytes capacities
  (Number of times of reprogramming: 100,000)
- Programming/erasing as background operations (BGOs)

■ On-chip SRAM, no wait states

- 128-Kbyte size capacities
- For instructions and operands

■ DMA

- DMAC: Incorporates four channels
- DTC

■ ELC

- Module operation can be initiated by event signals without going through interrupts.
- Modules can operate while the CPU is sleeping.

■ Reset and supply management

- Power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions

- External crystal oscillator or internal PLL for operation at 8 to 20 MHz
- Internal 125-kHz LOCO
- Dedicated 125-kHz LOCO for the IWDT
- Clock frequency accuracy measurement circuit (CAC)

■ Independent watchdog timer

- 125-kHz on-chip oscillator produces a dedicated clock signal to drive IWDT operation.

■ Useful functions for IEC60730 compliance

- Oscillation-stoppage detection, frequency measurement, CRC, IWDT, self-diagnostic function for the A/D converter, etc.

PLQP0144KA-A 20 × 20 mm, 0.5-mm pitch

■ Various communications interfaces

- SCI with many useful functions (up to 13 channels)
  Asynchronous mode, clock synchronous mode, smart card interface, simplified SPI, simplified I^2C, and extended serial mode
- I^2C bus interface: Transfer at up to 400 kbps (three channels)
- RSPI for high-speed transfer (two channels)

■ CEC transmission/reception function

- CEC signals can be transmitted/received conforming to CEC standard 1.4

■ Remote control signal reception

- Two units integrated
- Four pattern waveform matching supported

■ External address space

- Buses for high-speed data transfer (max. operating frequency of 27 MHz)
- 4 CS areas (4 × 16 Mbytes)
- Multiplexed bus or separate bus are selectable per area.
- 8-, or 16-bit bus space is selectable per area

■ Up to 20 extended-function timers

- 16-bit MTU: input capture, output compare, complementary PWM output, phase counting mode
  (six channels)
- 16-bit TPU: input capture, output capture, phase counting mode (six channels)
- 8-bit TMR (four channels)
- 16-bit compare-match timers (four channels)

■ 12-bit A/D converter

- Capable of conversion within 1 µs
- Sample-and-hold circuits (for three channels)
- Three-channel synchronized sampling available
- Self-diagnostic function and analog input disconnection detection assistance function

■ 10-bit D/A converter: 2 channels

■ Register write protection can protect values in important registers against overwriting

■ Up to 114 pins for general I/O ports

- Open drain, input pull-up

■ MPC

- Multiple locations are selectable for I/O pins of peripheral functions

■ Operating temperature range

- –40 to +85°C

1.Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline.

Table 1.1 Outline of Specifications (1 / 4)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 54 MHz • 32-bit RX CPU • Minimum instruction execution time: One instruction per state (cycle of the system clock) • Address space: 4-Gbyte linear • Register General purpose: Sixteen 32-bit registers Control: Nine 32-bit registers Accumulator: One 64-bit register • Basic instructions: 73 • Floating-point instructions: 8 • DSP instructions: 9 • Addressing modes: 10 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 × 32 → 64 bits • Memory-protection unit (MPU)
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	ROM	• Capacity: 1 M/1.5 M/2 Mbytes • 54 MHz, no-wait memory access • On-board programming: 3 types Off-board programming
	RAM	• Capacity: 128 Kbytes • 54 MHz, no-wait memory access
	E2 DataFlash	• Capacity: 32 Kbytes • Number of times for programming/erasing: 100,000
MCU operating mode		Single-chip mode, on-chip ROM enabled expansion mode, and on-chip ROM disabled expansion mode (software switching)
Clock	Clock generation circuit	• Main clock oscillator, low-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • Oscillation stop detection • Measuring circuit for accuracy of clock frequency (clock-accuracy check: CAC) • Independent settings for the system clock (ICLK), peripheral module clock (PCLKB), external bus clock (BCLK), and FlashIF clock (FCLK) The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 54 MHz (at max.) Peripheral modules run in synchronization with the peripheral module clock (PCLKB): 32 MHz (at max.) Devices connected to the external bus run in synchronization with the external bus clock (BCLK): 27 MHz (at max.) The flash peripheral circuit runs in synchronization with the FlashIF clock (FCLK): 32 MHz (at max.)
Reset		RES# pin reset, power-on reset, voltage monitoring reset, watchdog timer reset, independent watchdog timer reset, deep software standby reset, and software reset
Voltage detection	Voltage detection circuit	• When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated. The detection voltage level of voltage detection circuit 0 is fixed Voltage detection circuit 1 is capable of selecting the detection voltage from 3 levels Voltage detection circuit 2 is capable of selecting the detection voltage from 3 levels
Low power consumption	Low power consumption facilities	• Module stop function • Four low power consumption modes Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode
	Function for lower operating power consumption	• Operating power control modes High-speed operating mode, low-speed operating mode 1, low-speed operating mode 2
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 178 • External interrupts: 14 (NMI, IRQ0 to IRQ12 pins) • Non-maskable interrupts: 6 (the NMI pin, oscillation stop detection interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, WDT interrupt, and IWDT interrupt) • 16 levels specifiable for the order of priority

Table 1.1 Outline of Specifications (2 / 4)

Classification	Module/Function	Description
External bus extension		• The external address space can be divided into four areas (CS0 to CS3), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS3) A chip-select signal (CS0# to CS3#) can be output for each area. Each area is specifiable as an 8-bit or 16-bit bus space The data arrangement in each area is selectable as little or big endian (only for data). Bus format: Separate bus, multiplex bus • Wait control • Write buffer facility
DMA	DMA controller (DMACA)	• 4 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	Data transfer controller (DTCa)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Interrupts • Chain transfer function
I/O ports	General I/O ports	144-pin • I/O: 114 • Input: 9 (P40 to P47, P35) • Pull-up resistors: 111 • Open-drain outputs: 114 • 5-V tolerance: Not supported
Event link controller (ELC)		• Event signals of 56 types can be directly connected to the module • Operations of timer modules are selectable at event input • Capable of event link operation for ports B and E
Multi-function pin controller (MPC)		• Capable of selecting input/output function from multiple pins
Timers	16-bit timer pulse unit (TPUa)	• (16 bits × 6 channels) × 1 unit • Maximum of 16 pulse-input/output possible • Select from among seven or eight counter-input clock signals for each channel • Supports the input capture/output compare function • Output of PWM waveforms in up to 15 phases in PWM mode • Support for buffered operation, phase-counting mode (two-phase encoder input) and cascade-connected operation (32 bits × 2 channels) depending on the channel. • Capable of generating conversion start triggers for the A/D converters • Signals from the input capture pins are input via a digital filter
	Multi-function timer pulse unit 2 (MTU2a)	• (16 bits × 6 channels) × 1 unit • Up to 16 pulse-input/output lines and three pulse-input lines are available with six 16-bit timer channels • Select from among eight or seven counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, PCLK/256, PCLK/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD) other than channel 5, for which only four signals are available. • Input capture function • 21 output compare/input capture registers • Pulse output mode • Complementary PWM output mode • Reset synchronous PWM mode • Phase-counting mode • Generation of triggers for A/D converter conversion
	Port output enable 2 (POE2a)	Controls the high-impedance state of the MTU's waveform output pins
	Programmable pulse generator (PPG)	• (4 bits × 4 groups) × 1 unit • Pulse output with the MTU output as a trigger • Maximum of 16 pulse-output possible
	8-bit timer (TMR)	• (8 bits × 2 channels) × 2 units • Select from among seven internal clock signals (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/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 • Capable of generating baud-rate clocks for SCI5, SCI6, and SCI12 • Capable of generating a receive clock for the RCR
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among six counter-input clock signals (PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192)

Table 1.1 Outline of Specifications (3 / 4)

Classification	Module/Function	Description
Timers	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Counter-input clock: IWDT-dedicated on-chip oscillator Frequency divided by 1, 16, 32, 64, 128, or 256
Communication functions	Serial communications interfaces (SCIe, SCIf)	• 13 channels (channel 0 to 11: SCIe, channel 12: SCIf) • Serial communications modes: Asynchronous, clock synchronous, and smart-card interface • 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 (SCI5, SCI6, and SCI12) • Simple IIC • Simple SPI • Master/slave mode supported (SCIf only) • Start frame and information frame are included (SCIf only)
	I²C bus interface (RIIC)	• 3 channel • Communications formats: I²C bus format/SMBus format • Master/slave selectable • Max. transfer rate: Supports the fast mode (400 Kbps)
	Serial peripheral interface (RSPI)	• 2 channels • Transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPI clock (RSPCK) 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 Choice of LSB-first or MSB-first transfer 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) • Double buffers for both transmission and reception
	CEC transmission/ reception circuit (CEC) (3-V packages only)	CEC signals can be generated and received conforming to the CEC standard, and communication states can be detected by hardware. • Serial communication can be performed conforming to the CEC standard. • The operating clock can be selected from among the PCLK, main clock, and IWDTCLK. • Any value can be set for the low-level width/bit width of the start bit and data bit during transmission and reception. • Errors and communication states can be detected by hardware. • An error handling pulse can be output when a timing error of the long bit width is detected. • Signal-free time can be counted. • Receive operation can be restarted by detecting the start bit during reception.
	Remote control signal receiver (RCR) (3-V packages only)	• Two units • Four pattern matching (header, data 0, data 1, and special data detection) • 8-byte receive buffer per unit • The operating clock can be selected from among the PCLK, main clock, IWDTCLK, and TMR.
12-bit A/D converter (S12ADb)		• 12 bits (16 channels × 1 unit) • 12-bit resolution • Minimum conversion time: 1.0 μs per channel (in operation with ADCLK at 50 MHz) • Operating modes Scan mode (single scan mode, continuous scan mode, and group scan mode) • Sample-and-hold function • Self-diagnosis for the A/D converter • Assistance in detecting disconnected analog inputs • Double-trigger mode (duplication of A/D conversion data) • A/D conversion start conditions A software trigger, a trigger from a timer (MTU), an external trigger signal, or ELC
D/A converter (DA)		• 2 channels • 10-bit resolution • Output voltage: 0 V to VREFH
CRC calculator (CRC)		• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Data Operation Circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Operating frequency		54 MHz

Table 1.1 Outline of Specifications (4 / 4)

Classification	Module/Function	Description
Power supply voltage		• 3-V package VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V • 5-V package VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V
Operating temperature		–40 to +85°C (products with wide-temperature-range spec.)
Packages		144-pin LQFP (PLQP0144KA-A)
On-chip debugging system		• E1 emulator (JTAG and FINE interfaces) • E20 emulator (JTAG interface)

Table 1.2 Comparison of Functions of Different RX634 Group Products

Functions		**RX634 Group**	
Group Products		**3-V package**	**5-V package**
External bus	External bus width	16 bits	
DMA	DMA controller	Channels 0 to 3	
	Data transfer controller	Supported	
Timers	16-bit timer pulse unit	Channels 0 to 5	
	Multi-function timer pulse unit 2	Channels 0 to 5	
	Port output enable 2	Supported	
	Programmable pulse generator	Supported	
	8-bit timer	Channels 0 to 3	
	Compare match timer	Channels 0 to 3	
	Watchdog timer	Supported	
	Independent watchdog timer	Supported	
Communication functions	Serial communications interface (SCIe)	Channels 0 to 11	
	Serial communications interface (SCIf)	Channel 12	
	I^2C bus interface	Channels 0, 1, 3	
	Serial peripheral interface	Channels 0, 1	
	CEC transmission/reception circuit (CEC)	Supported	Not supported
	Remote control signal receiver (RCR)	Channels 0, 1	Not supported
12-bit A/D converter		AN000 to AN015	
D/A converter		Channels 0, 1	
CRC calculator		Supported	
Event link controller		Supported	
Clock frequency accuracy measurement circuit (CAC)		Supported	

1.2 List of Products

Table 1.3 is a lists of products, and Figure 1.1 shows how to read the product part no., memory capacity, and package type.

Table 1.3 List of Products

Group	Part No.	Orderable Part No.	Package	ROM Capacity	RAM Capacity	Power Supply Voltage
RX634	R5F5634EYDFB	R5F5634EYDFB#30	PLQP0144KA-A	2 Mbytes	128 Kbytes	VCC = AVCC0 = 4.0 to 5.5V
	R5F5634ECDFB	R5F5634ECDFB#30	PLQP0144KA-A	2 Mbytes	128 Kbytes	VCC = AVCC0 = 2.7 to 3.6V
	R5F5634DYDFB	R5F5634DYDFB#30	PLQP0144KA-A	1.5 Mbytes	128 Kbytes	VCC = AVCC0 = 4.0 to 5.5V
	R5F5634DCDFB	R5F5634DCDFB#30	PLQP0144KA-A	1.5 Mbytes	128 Kbytes	VCC = AVCC0 = 2.7 to 3.6V
	R5F5634BYDFB	R5F5634BYDFB#30	PLQP0144KA-A	1 Mbyte	128 Kbytes	VCC = AVCC0 = 4.0 to 5.5V
	R5F5634BCDFB	R5F5634BCDFB#30	PLQP0144KA-A	1 Mbyte	128 Kbytes	VCC = AVCC0 = 2.7 to 3.6V

Note: Orderable part numbers are current as of when this manual was published. Please make sure to refer to the relevant product page on the Renesas website for the latest part numbers.

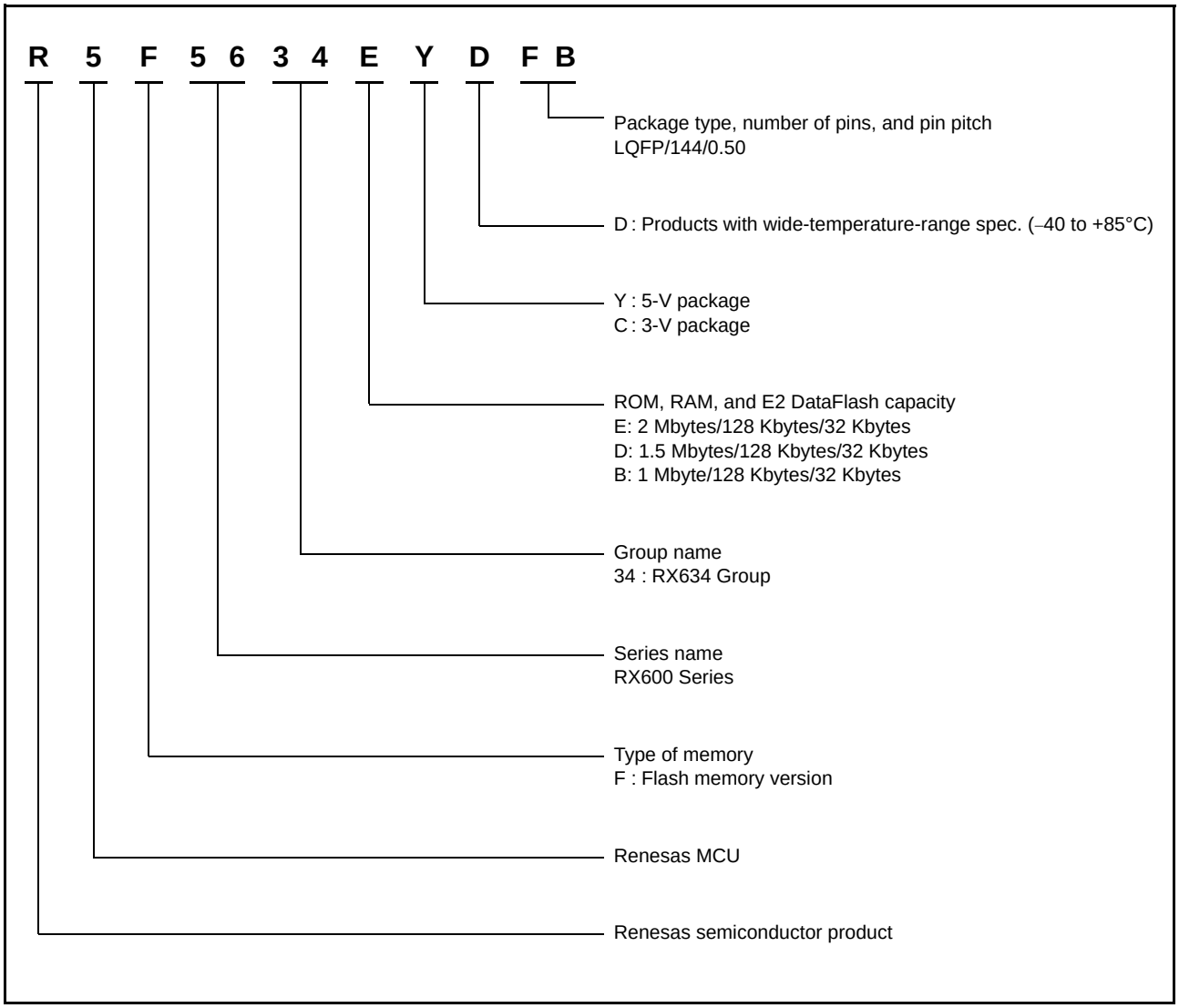
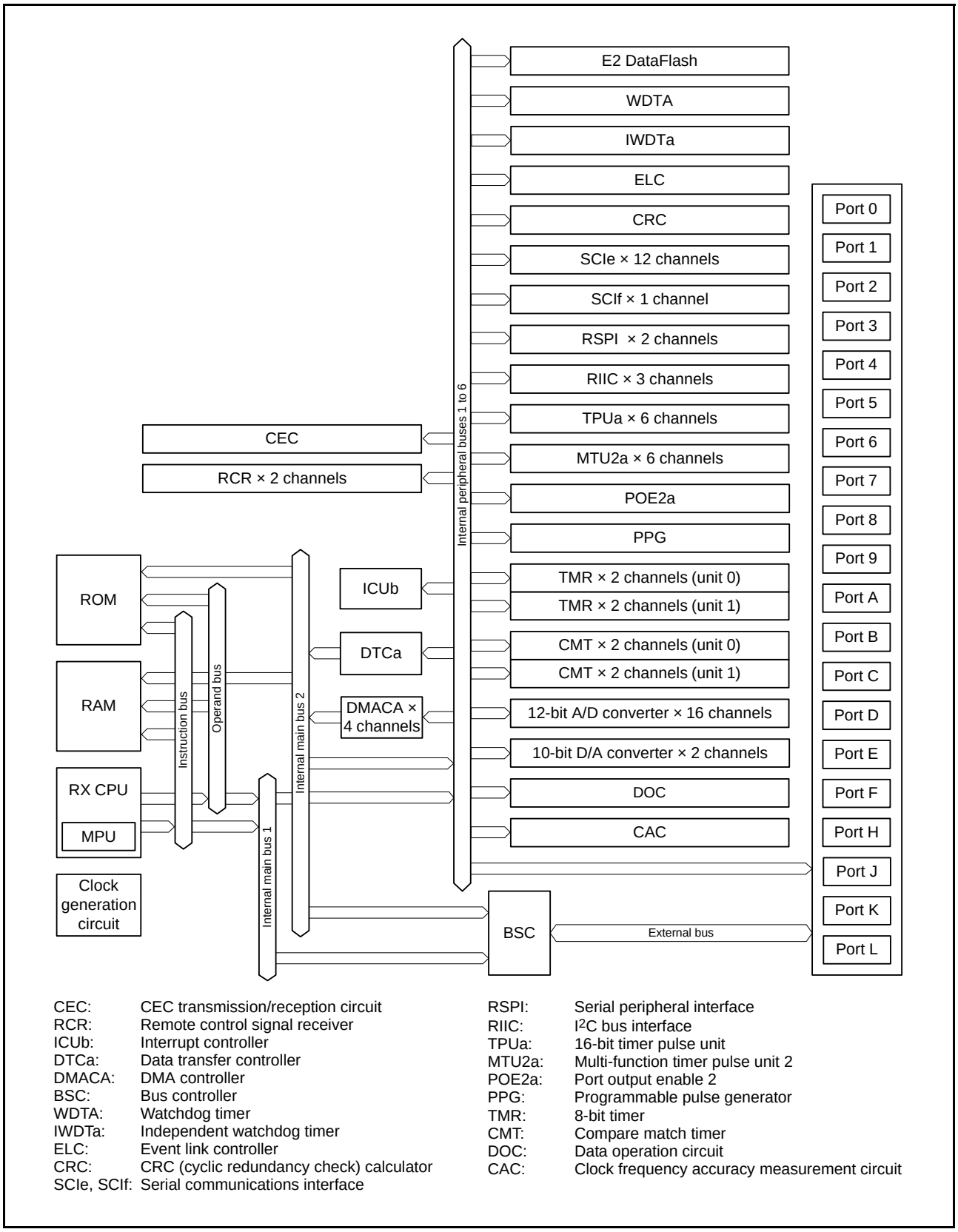


Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type

1.3 Block Diagram

Figure 1.2 shows a block diagram.



CEC: CEC transmission/reception circuit
RCR: Remote control signal receiver
ICUb: Interrupt controller
DTCa: Data transfer controller
DMACA: DMA controller
BSC: Bus controller
WDTA: Watchdog timer
IWDTa: Independent watchdog timer
ELC: Event link controller
CRC: CRC (cyclic redundancy check) calculator
SCIe, SCIf: Serial communications interface
RSPI: Serial peripheral interface
RIIC: I^2C bus interface
TPUa: 16-bit timer pulse unit
MTU2a: Multi-function timer pulse unit 2
POE2a: Port output enable 2
PPG: Programmable pulse generator
TMR: 8-bit timer
CMT: Compare match timer
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1 / 4)

Classifications	Pin Name	I/O	Description
Power supply	VCC	—	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	—	Connect this pin to the VSS pin via the 0.1 μF smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	—	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for connecting a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	BCLK	Output	Outputs the external bus clock for external devices.
Clock frequency accuracy measurement	CACREF	Input	Input for the trigger signal in measuring accuracy of the clock frequency
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin 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 onchip emulator is used, this pin should be driven high. When not used, it should be driven low.
On-chip emulator	FINEC	Input	Fine interface clock pin
	FINED	I/O	Fine interface pin
	TRST#	Input	On-chip emulator pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TRCLK	Output	This pin outputs the clock for synchronization with the trace data.
	TRSYNC#	Output	This pin indicates that output from the TRDATA0 to TRDATA3 pins is valid.
	TRDATA0 to TRDATA3	Output	These pins output the trace information.
Address bus	A0 to A23	Output	Output pins for the address.
Data bus	D0 to D15	I/O	Input and output pins for the bidirectional data bus.
Multiplexed bus	A0/D0 to A15/D15	I/O	Address/data multiplexed bus
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress.
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in single-write strobe mode.
	WR0# to WR1#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, and D15 to D8) is valid in writing to the external bus interface space, in byte strobe mode.
	BC0# to BC1#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0 and D15 to D8) is valid in access to the external bus interface space, in single-write strobe mode.
	ALE	Output	Address latch signal when address/data multiplexed bus is selected.
	WAIT#	Input	Input pin for wait request signals in access to the external space.
	CS0# to CS3#	Output	Select signals for areas 0 to 3.

Table 1.4 Pin Functions (2 / 4)

Classifications	Pin Name	I/O	Description
Interrupt	NMI	Input	Non-maskable interrupt request pin.
	IRQ0 to IRQ12	Input	Interrupt request pins.
16-bit timer pulse unit	TIOCA0, TIOCB0 TIOCC0, TIOCD0	I/O	The TGRA0 to TGRD0 input capture input/output compare output/ PWM output pins.
	TIOCA1, TIOCB1	I/O	The TGRA1 and TGRB1 input capture input/output compare output/ PWM output pins.
	TIOCA2, TIOCB2	I/O	The TGRA2 and TGRB2 input capture input/output compare output/ PWM output pins.
	TIOCA3, TIOCB3 TIOCC3, TIOCD3	I/O	The TGRA3 to TGRD3 input capture input/output compare output/ PWM output pins.
	TIOCA4, TIOCB4	I/O	The TGRA4 and TGRB4 input capture input/output compare output/ PWM output pins.
	TIOCA5, TIOCB5	I/O	The TGRA5 and TGRB5 input capture input/output compare output/ PWM output pins.
	TCLKA, TCLKB TCLKC, TCLKD	Input	Input pins for external clock signals.
Multi-function timer pulse unit 2	MTIOC0A, MTIOC0B MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/ PWM output pins.
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/ PWM output pins.
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/ PWM output pins.
	MTIOC3A, MTIOC3B MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/ PWM output pins.
	MTIOC4A, MTIOC4B MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/ PWM output pins.
	MTIC5U, MTIC5V, MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/external pulse input pins.
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.
Port output enable 2	POE0# to POE3#, POE8#	Input	Input pins for request signals to place the MTU pins in the high impedance state.
Programmable pulse generator	PO0 to PO15	Output	Output pins for the pulse signals.
8-bit timer	TMO0 to TMO3	Output	Compare match output pins.
	TMCI0 to TMCI3	Input	Input pins for external clocks to be input to the counter.
	TMRI0 to TMRI3	Input	Input pins for the counter reset.

Table 1.4 Pin Functions (3 / 4)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIe)	• Asynchronous mode/clock synchronous mode		
	SCK0 to SCK11	I/O	Input/output pins for the clock
	RXD0 to RXD11	Input	Input pins for received data
	TXD0 to TXD11	Output	Output pins for transmitted data
	CTS0# to CTS11#	Input	Input pins for controlling the start of transmission and reception
	RTS0# to RTS11#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL0 to SSCL11	I/O	Input/output pins for the I^2C clock
	SSDA0 to SSDA11	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK0 to SCK11	I/O	Input/output pins for the clock
	SMISO0 to SMISO11	I/O	Input/output pins for slave transmission of data
	SMOSI0 to SMOSI11	I/O	Input/output pins for master transmission of data
	SS0# to SS11#	Input	Chip-select input pins
Serial communications interface (SCIf)	• 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 transmit data
	SMOSI12	I/O	Input/output pin for master transmit data
	SS12#	Input	Chip-select input pin
	• Extended serial mode		
	RXDX12	Input	Input pin for data reception by SCId
	TXDX12	Output	Output pin for data transmission by SCId
	SIOX12	I/O	Input/output pin for data reception or transmission by SCId
I^2C bus interface	SCL0, SCL1, SCL3	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open-drain output.
	SDA0, SDA1, SDA3	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open-drain output.
Serial peripheral interface	RSPCKA, RSPCKB	I/O	Clock input/output pin for the RSPI.
	MOSIA, MOSIB	I/O	Input or output data output from the master for the RSPI.
	MISOA, MISOB	I/O	Input or output data output from the slave for the RSPI.
	SSLA0, SSLB0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1 to SSLA3 SSLB1 to SSLB3	Output	Output pins to select the slave for the RSPI.
CEC transmission/ reception circuit (CEC)	CECIO	I/O	Input/output pin for CEC communication data
Remote control signal receiver (RCR)	PMC0	Input	Input pin for external pulse signal
	PMC1	Input	Input pin for external pulse signal

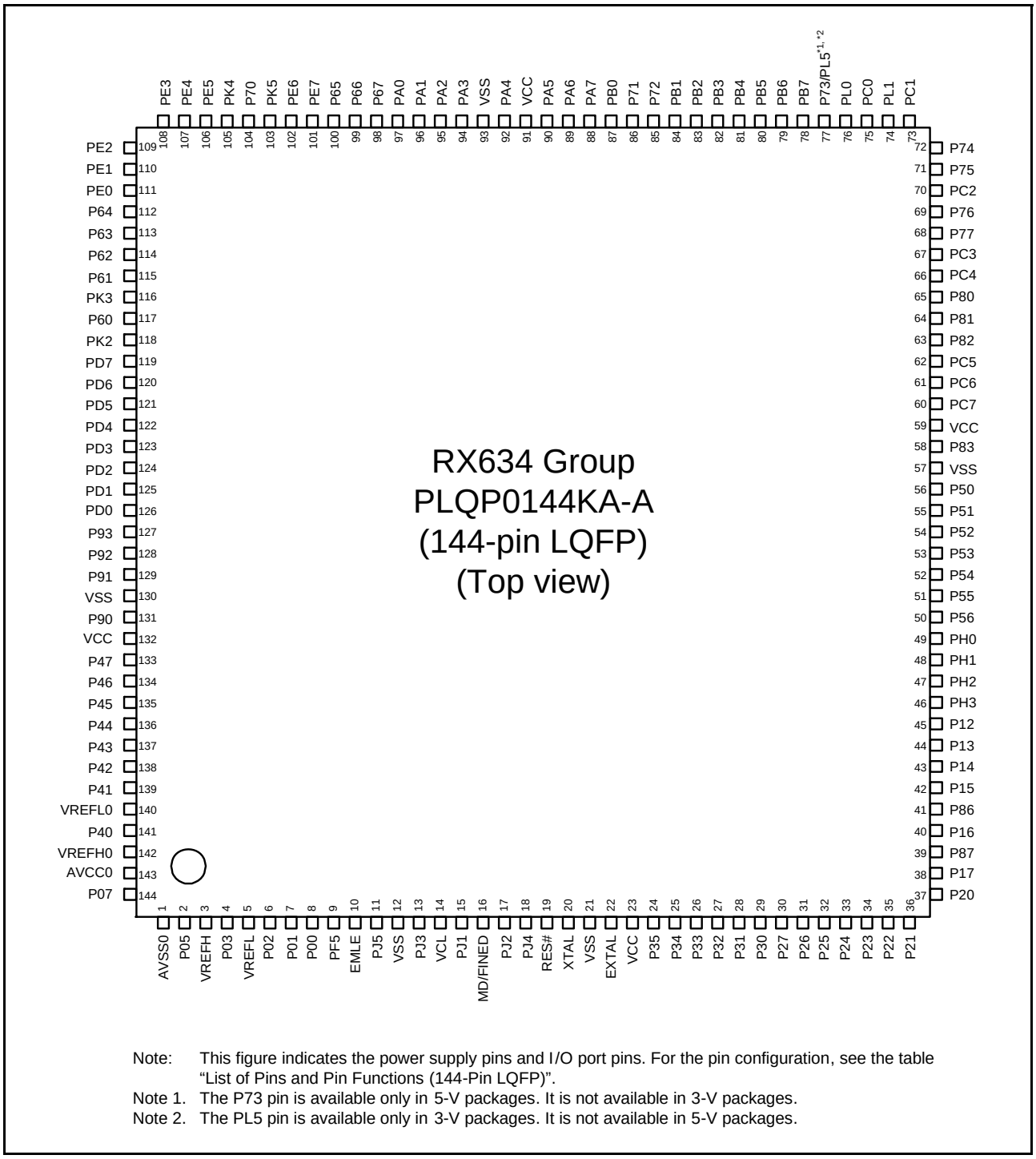
Table 1.4 Pin Functions (4 / 4)

Classifications	Pin Name	I/O	Description
12-bit A/D converter	AN000 to AN015	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pin for the external trigger signals that start the A/D conversion.
D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter.
Analog power supply	AVCC0	—	Analog voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC if the 12-bit A/D converter is not to be used.
	AVSS0	—	Analog ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	VREFH0	—	Analog reference voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC if the 12-bit A/D converter is not to be used.
	VREFL0	—	Analog reference ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	VREFH	—	Analog voltage supply pin for the D/A converter. Connect this pin to VCC if the D/A converter is not to be used.
	VREFL	—	Analog ground pin for the D/A converter. Connect this pin to VSS if the D/A converter is not to be used.
I/O ports	P00 to P03, P05, P07	I/O	6-bit input/output pins.
	P12 to P17	I/O	6-bit input/output pins.
	P20 to P27	I/O	8-bit input/output pins.
	P30 to P35	I/O	6-bit input/output pins. (P35 input pin)
	P40 to P47	Input	8-bit input pins.
	P50 to P56	I/O	7-bit input/output pins.
	P60 to P67	I/O	8-bit input/output pins.
	P70 to P77*1	I/O	8-bit input/output pins.
	P80 to P83, P86, P87	I/O	6-bit input/output pins.
	P90 to P93	I/O	4-bit input/output pins.
	PA0 to PA7	I/O	8-bit input/output pins.
	PB0 to PB7	I/O	8-bit input/output pins.
	PC0 to PC7	I/O	8-bit input/output pins.
	PD0 to PD7	I/O	8-bit input/output pins.
	PE0 to PE7	I/O	8-bit input/output pins.
	PF5	I/O	1-bit input/output pin.
	PH0 to PH3	I/O	4-bit input/output pins.
	PJ1 to PJ5	I/O	5-bit input/output pins.
	PK2 to PK5	I/O	4-bit input/output pins.
	PL0, PL1, PL5*2	I/O	3-bit input/output pins.

Note 1. The P73 pin is available only in 5-V packages. It is not available in 3-V packages.
Note 2. The PL5 pin is available only in 3-V packages. It is not available in 5-V packages.

1.5 Pin Assignments

Figure 1.3 shows the pin assignments. Table 1.5 shows the lists of pins and pin functions.



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see the table "List of Pins and Pin Functions (144-Pin LQFP)".
Note 1. The P73 pin is available only in 5-V packages. It is not available in 3-V packages.
Note 2. The PL5 pin is available only in 3-V packages. It is not available in 5-V packages.

Figure 1.3 Pin Assignments of the 144-Pin LQFP

Table 1.5 List of Pins and Pin Functions (144-Pin LQFP) (1 / 4)

Pin No.	Power Supply, Clock, System Control	I/O Port	External Bus	Timers (MTU, TPU, TMR, POE, PPG, CAC)	Communications (SCIe, SCIf, RSPI, RIIC, CEC, RCR)	Interrupt	AD, DA
1	AVSS0						
2		P05					DA1
3	VREFH						
4		P03				IRQ11	DA0
5	VREFL						
6		P02		TMCI1	SCK6	IRQ10	
7		P01		TMCI0	RXD6/SMISO6/SSCL6/PMC1	IRQ9	
8		P00		TMRI0	TXD6/SMOSI6/SSDA6/PMC0	IRQ8	
9		PF5				IRQ4	
10	EMLE						
11		PJ5					
12	VSS						
13		PJ3		MTIOC3C	CTS0#/RTS0#/SS0#/CTS6#/ RTS6#/SS6#		
14	VCL						
15		PJ1		MTIOC3A			
16	MD/FINED						
17		PJ2					
18		PJ4					
19	RES#						
20	XTAL						
21	VSS						
22	EXTAL						
23	VCC						
24		P35				NMI	
25	TRST#	P34		MTIOC0A/TMCI3/POE2#/ PO12	SCK0/SCK6	IRQ4	
26		P33		MTIOC0D/TIOCD0/TMRI3/ POE3#/PO11	RXD0/SMISO0/SSCL0/RXD6/ SMISO6/SSCL6	IRQ3_DS	
27		P32		MTIOC0C/TIOCC0/TMO3/ PO10	TXD0/SMOSI0/SSDA0/TXD6/ SMOSI6/SSDA6	IRQ2_DS	
28	TMS	P31		MTIOC4D/TMCI2/PO9	CTS1#/RTS1#/SS1#/SSLB0	IRQ1_DS	
29	TDI	P30		MTIOC4B/TMRI3/POE8#/ PO8	RXD1/SMISO1/SSCL1/MISOB	IRQ0_DS	
30	FINEC/TCK	P27	CS3#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
31	TDO	P26	CS2#	MTIOC2A/TMO1/PO6	TXD1/SMOSI1/SSDA1/CTS3#/ RTS3#/SS3#/MOSIB		
32		P25	CS1#	MTIOC4C/MTCLKB/TIOCA4/ PO5	RXD3/SMISO3/SSCL3		ADTRG0#
33		P24	CS0#	MTIOC4A/MTCLKA/TIOCB4/ TMRI1/PO4	SCK3		
34		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/SMOSI3/SSDA3/CTS0#/ RTS0#/SS0#		
35		P22		MTIOC3B/MTCLKC/TIOCC3/ TMO0/PO2	SCK0		
36		P21		MTIOC1B/TIOCA3/TMCI0/ PO1	RXD0/SMISO0/SSCL0/SCL1	IRQ9	
37		P20		MTIOC1A/TIOCB3/TMRI0/ PO0	TXD0/SMOSI0/SSDA0/SDA1	IRQ8	
38		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ POE8#/PO15	SCK1/TXD3/SMOSI3/SSDA3/ MISOA/SDA0_DS	IRQ7	

Table 1.5 List of Pins and Pin Functions (144-Pin LQFP) (2 / 4)

Pin No.	Power Supply, Clock, System Control	I/O Port	External Bus	Timers (MTU, TPU, TMR, POE, PPG, CAC)	Communications (SCIe, SCIf, RSPI, RIIC, CEC, RCR)	Interrupt	AD, DA
39		P87		TIOCA2			
40		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/PO14	TXD1/SMOSI1/SSDA1/RXD3/ SMISO3/SSCL3/MOSIA/ SCL0_DS	IRQ6	ADTRG0#
41		P86		TIOCA0			
42		P15		MTIOC0B/MTCLKB/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SMISO1/SSCL1/SCK3	IRQ5	
43		P14		MTIOC3A/MTCLKA/TIOCB5/ TCLKA/TMRI2/PO15	CTS1#/RTS1#/SS1#	IRQ4	
44		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/SDA0	IRQ3	
45		P12		TMCI1	RXD2/SMISO2/SSCL2/SCL0	IRQ2	
46		PH3		TMCI0			
47		PH2		TMRI0		IRQ1	
48		PH1		TMO0		IRQ0	
49		PH0		CACREF			
50		P56		MTIOC3C/TIOCA1			
51	TRDATA3	P55	WAIT#	MTIOC4D/TMO3		IRQ10	
52	TRDATA2	P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#		
53		P53	BCLK				
54		P52	RD#		RXD2/SMISO2/SSCL2/SSLB3		
55		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
56		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/SSLB1		
57	VSS						
58	TRCLK	P83		MTIOC4C	CTS10#/RTS10#/SS10#		
59	VCC						
60		PC7	A23/CS0#	MTIOC3A/MTCLKB/TMO2/ CACREF	TXD8/SMOSI8/SSDA8/MISOA		
61		PC6	A22/CS1#	MTIOC3C/MTCLKA/TMCI2	RXD8/SMISO8/SSCL8/MOSIA		
62		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/TMRI2	SCK8/RSPCKA		
63	TRSYNC#	P82		MTIOC4A	TXD10/SMOSI10/SSDA10		
64	TRDATA1	P81		MTIOC3D	RXD10/SMISO10/SSCL10		
65	TRDATA0	P80		MTIOC3B	SCK10		
66		PC4	A20/CS3#	MTIOC3D/MTCLKC/TMCI1/ POE0#	SCK5/CTS8#/RTS8#/SS8#/ SSLA0		
67		PC3	A19	MTIOC4D/TCLKB	TXD5/SMOSI5/SSDA5		
68		P77			TXD11/SMOSI11/SSDA11		
69		P76			RXD11/SMISO11/SSCL11		
70		PC2	A18	MTIOC4B/TCLKA	RXD5/SMISO5/SSCL5/SSLA3		
71		P75			SCK11		
72		P74			CTS11#/RTS11#/SS11#		
73		PC1	A17	MTIOC3A/TCLKD	SCK5/SSLA2/SDA3	IRQ12	
74		PL1					
75		PC0	A16	MTIOC3C/TCLKC	CTS5#/RTS5#/SS5#/SSLA1/ SCL3		
76		PL0					
77		P73*1				IRQ12	
		PL5*1			CECIO	IRQ12	
78		PB7	A15	MTIOC3B/TIOCB5	TXD9/SMOSI9/SSDA9		

Table 1.5 List of Pins and Pin Functions (144-Pin LQFP) (3 / 4)

Pin No.	Power Supply, Clock, System Control	I/O Port	External Bus	Timers (MTU, TPU, TMR, POE, PPG, CAC)	Communications (SCIe, SCIf, RSPI, RIIC, CEC, RCR)	Interrupt	AD, DA
79		PB6	A14	MTIOC3D/TIOCA5	RXD9/SMISO9/SSCL9		
80		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/POE1#	SCK9		
81		PB4	A12	TIOCA4	CTS9#/RTS9#/SS9#		
82		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ POE3#	SCK4/SCK6		
83		PB2	A10	TIOCC3/TCLKC	CTS4#/RTS4#/SS4#/CTS6#/ RTS6#/SS6#		
84		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0	TXD4/SMOSI4/SSDA4/TXD6/ SMOSI6/SSDA6	IRQ4_DS	
85		P72					
86		P71					
87		PB0	A8	MTIC5W/TIOCA3	RXD4/SMISO4/SSCL4/RXD6/ SMISO6/SSCL6/RSPCKA	IRQ12	
88		PA7	A7	TIOCB2	MISOA		
89		PA6	A6	MTIC5V/MTCLKB/TIOCA2/ TMCI3/POE2#	CTS5#/RTS5#/SS5#/MOSIA		
90		PA5	A5	TIOCB1	RSPCKA		
91	VCC						
92		PA4	A4	MTIC5U/MTCLKA/TIOCA1/ TMRI0	TXD5/SMOSI5/SSDA5/SSLA0	IRQ5_DS	
93	VSS						
94		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB	RXD5/SMISO5/SSCL5	IRQ6_DS	
95		PA2	A2		RXD5/SMISO5/SSCL5/SSLA3		
96		PA1	A1	MTIOC0B/MTCLKC/TIOCB0	SCK5/SSLA2	IRQ11	
97		PA0	A0/BC0#	MTIOC4A/TIOCA0/CACREF	SSLA1		
98		P67					
99		P66					
100		P65					
101		PE7	D15 [A15/D15]		MISOB	IRQ7	AN015
102		PE6	D14 [A14/D14]		CTS4#/RTS4#/SS4#/MOSIB	IRQ6	AN014
103		PK5			TXD4/SMOSI4/SSDA4		
104		P70			SCK4		
105		PK4			RXD4/SMISO4/SSCL4		
106		PE5	D13 [A13/D13]	MTIOC4C/MTIOC2B	RSPCKB	IRQ5	AN013
107		PE4	D12 [A12/D12]	MTIOC4D/MTIOC1A	SSLB0		AN012
108		PE3	D11 [A11/D11]	MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN011
109		PE2	D10 [A10/D10]	MTIOC4A	RXD12/RXDX12/SMISO12/ SSCL12/SSLB3/MOSIB	IRQ7_DS	AN010
110		PE1	D9 [A9/D9]	MTIOC4C	TXD12/TXDX12/SMOSI12/ SSDA12/SSLB2/RSPCKB/ SIOX12		AN009
111		PE0	D8 [A8/D8]		SCK12/SSLB1		AN008
112		P64					
113		P63					
114		P62					
115		P61			CTS9#/RTS9#/SS9#		
116		PK3			RXD9/SMISO9/SSCL9		
117		P60			SCK9		

Table 1.5 List of Pins and Pin Functions (144-Pin LQFP) (4 / 4)

Pin No.	Power Supply, Clock, System Control	I/O Port	External Bus	Timers (MTU, TPU, TMR, POE, PPG, CAC)	Communications (SCIe, SCIf, RSPI, RIIC, CEC, RCR)	Interrupt	AD, DA
118		PK2			TXD9/SMOSI9/SSDA9		
119		PD7	D7 [A7/D7]	MTIC5U/POE0#		IRQ7	
120		PD6	D6 [A6/D6]	MTIC5V/POE1#		IRQ6	
121		PD5	D5 [A5/D5]	MTIC5W/POE2#		IRQ5	
122		PD4	D4 [A4/D4]	POE3#		IRQ4	
123		PD3	D3 [A3/D3]	POE8#		IRQ3	
124		PD2	D2 [A2/D2]	MTIOC4D		IRQ2	
125		PD1	D1 [A1/D1]	MTIOC4B		IRQ1	
126		PD0	D0 [A0/D0]			IRQ0	
127		P93			CTS7#/RTS7#/SS7#		
128		P92			RXD7/SMISO7/SSCL7		
129		P91			SCK7		
130	VSS						
131		P90			TXD7/SMOSI7/SSDA7		
132	VCC						
133		P47					AN007
134		P46					AN006
135		P45					AN005
136		P44					AN004
137		P43					AN003
138		P42					AN002
139		P41					AN001
140	VREFL0						
141		P40					AN000
142	VREFH0						
143	AVCC0						
144		P07					ADTRG0#

Note 1. Pin 77 is available as P73 in 5-V packages, and PL5 in 3-V packages.

2. CPU

This MCU has the high-speed, high-performance RX CPU as its core.
A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.
The CPU has 73 basic instructions and 8 floating-point operation instructions, and nine DSP instructions, for a total of 90 instructions. It has 10 addressing modes and caters to register–register operations, register–memory operations, immediate–register operations, immediate–memory operations, memory–memory transfer, and bitwise operations.
High-speed operation was realized by achieving execution in a single cycle not only for register–register operations, but also for other types of multiple instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.
The RX CPU has a five-stage pipeline for processing instructions. The stages are instruction fetching, instruction decoding, execution, memory access, and write-back. In cases where pipeline processing is drawn-out by memory access, subsequent operations may in fact be executed earlier. By adopting “out-of-order completion” of this kind, the execution of instructions is controlled to optimize numbers of clock cycles.

2.1 Features

- High instruction execution rate: One instruction in one clock cycle
- Address space: 4-Gbyte linear
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Nine 32-bit registers
  Accumulator: One 64-bit register
- Basic instructions: 73 (arithmetic/logic instructions, data-transfer instructions, branch instructions, bit-manipulation instructions, string-manipulation instructions, and system-manipulation instructions)
  Relative branch instructions to suit branch distances
  Variable-length instruction format (lengths from one to eight bytes)
  Short formats for frequently used instructions
- Floating-point operation instructions: 8
- DSP instructions: 9
  Supports 16-bit × 16-bit multiplication and multiply-and-accumulate operations.
  Rounds the data in the accumulator.
- Addressing modes: 10
- Five-stage pipeline
  Adoption of “out-of-order completion”
- Processor modes
  A supervisor mode and a user mode are supported.
- Floating-point operation unit
  Supports single-precision (32-bit) floating point
  Supports data types and exceptions in conformance with the IEEE754 standard
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The RX CPU has sixteen general-purpose registers, nine control registers, and one accumulator used for DSP instructions.

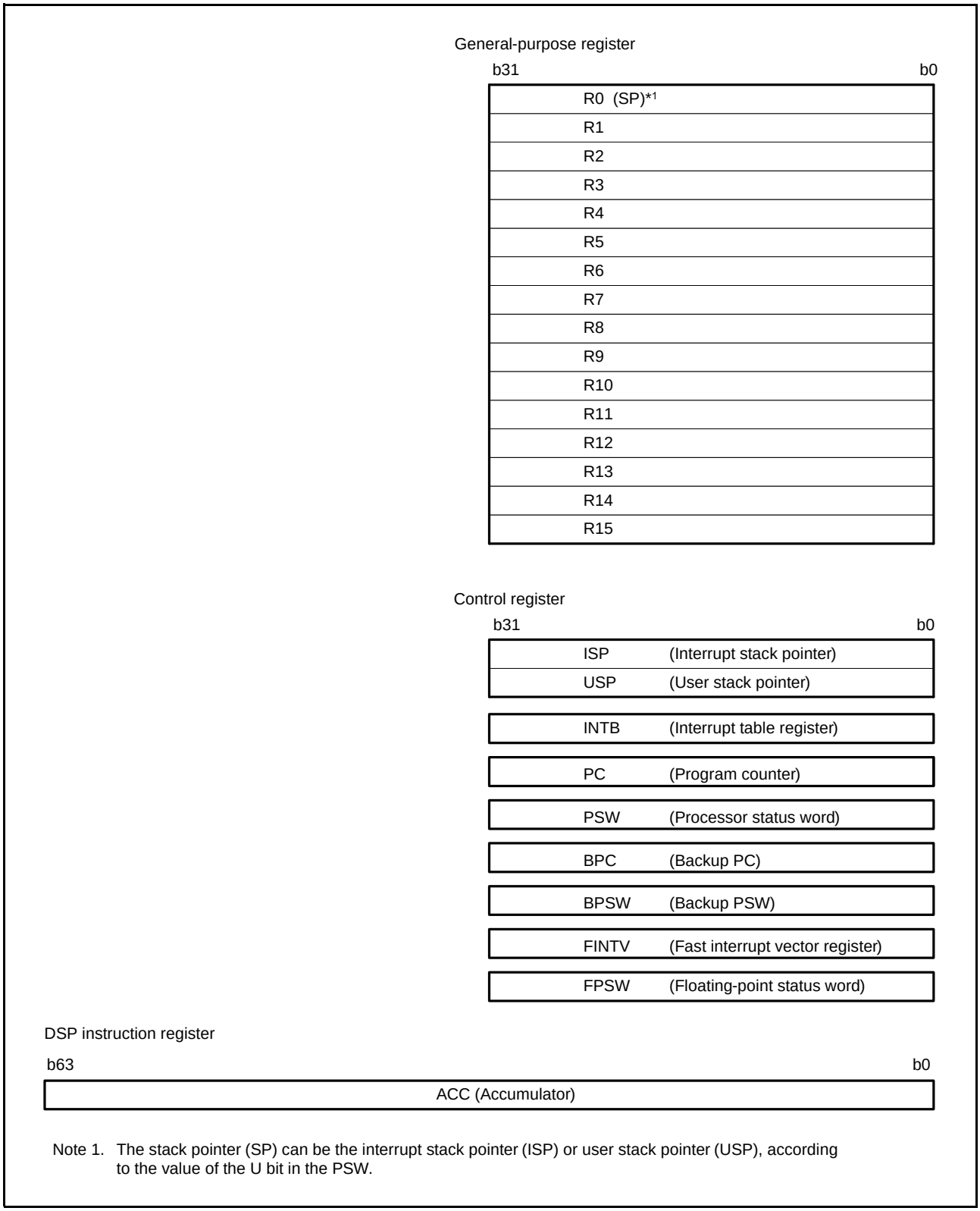


Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

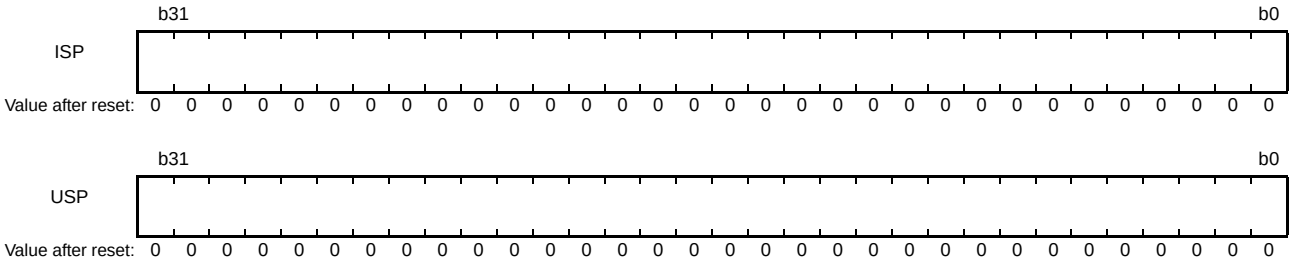
This CPU has sixteen general-purpose registers (R0 to R15). R1 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) 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 nine control registers.

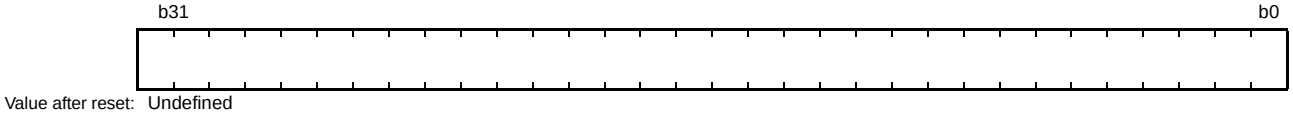
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Floating-point status word (FPSW)

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



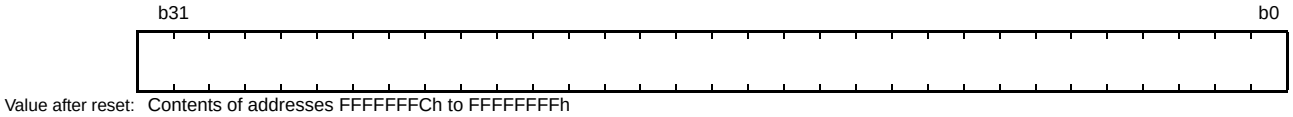
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of four, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.2 Interrupt Table Register (INTB)



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

2.2.2.3 Program Counter (PC)



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

2.2.2.4 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 indicates whether a carry, borrow, or shift-out has occurred as the result of an operation.

Z Flag (Zero Flag)

This flag indicates that the result of an operation was 0.

S Flag (Sign Flag)

This flag indicates that the result of an operation was negative.

O Flag (Overflow Flag)

This flag indicates that an overflow occurred during an operation.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When an exception is accepted, the value of this bit becomes 0.

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.

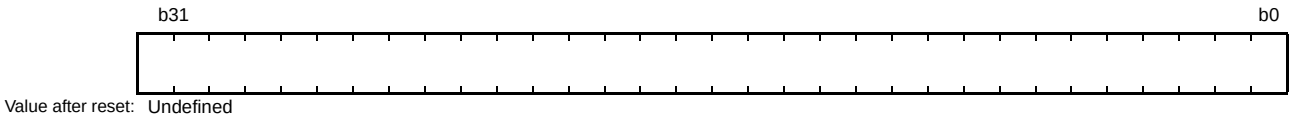
PM Bit (Processor Mode Select)

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)

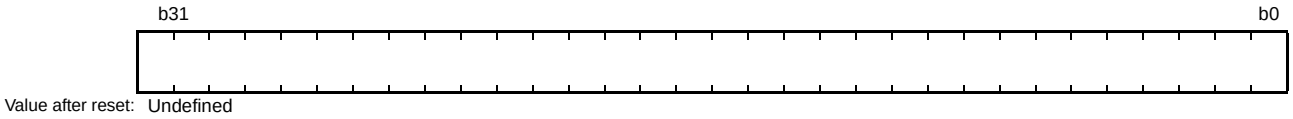
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the 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.

2.2.2.5 Backup PC (BPC)



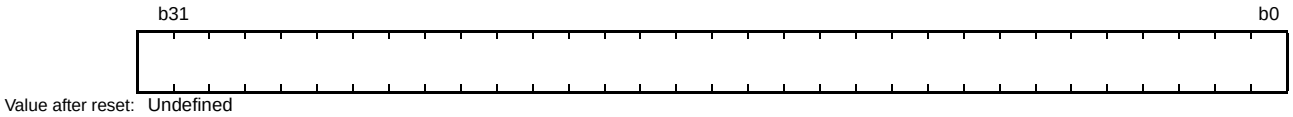
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

2.2.2.6 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.

2.2.2.7 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.

2.2.2.8 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]	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

Bit	Symbol	Bit Name	Description	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	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 cleared to 0 by software.

The floating-point status word (FPSW) indicates the results of floating-point operations.
When an exception handling enable bit (Ej) enables the exception handling (Ej = 1), the exception cause can be identified by checking the corresponding Cj flag in the exception handling routine. If the exception handling is masked (Ej = 0), the occurrence of exception can be checked by reading the Fj flag at the end of a series of processing. Once the Fj flag has been set to 1, this value is retained until it is cleared to 0 by software (j = X, U, Z, O, or V).

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

These bits specify the floating-point rounding-mode.

Explanation of 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 which would be obtained with an infinite number of digits. 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)

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

- The bit that has been set to 1 is cleared to 0 when the FPU instruction is executed.

- 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 Flag (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 floating-point exceptions specified in the IEEE754 standard is generated by the 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 floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

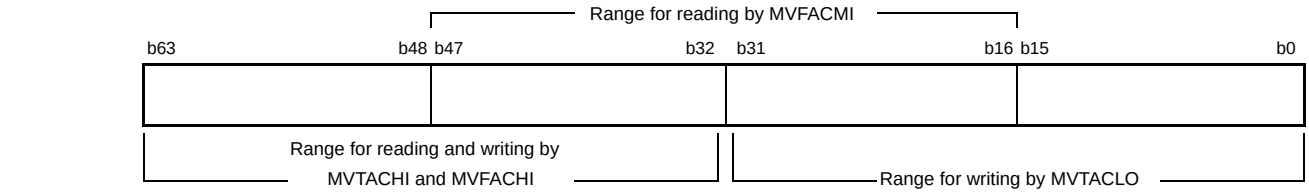
- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is cleared to 0 by software. (accumulation flag)

FS Flag (Floating-Point Error Summary Flag)

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

2.2.3 Register Associated with DSP Instructions

2.2.3.1 Accumulator (ACC)



Value after reset: Undefined

The accumulator (ACC) is a 64-bit register used for DSP instructions. The accumulator is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.
Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively.
Use the MVFACHI and MVFACMI instructions for reading data from the accumulator. The MVFACHI and MVFACMI instructions read data from the higher-order 32 bits (bits 63 to 32) and the middle 32 bits (bits 47 to 16), respectively.

2.3 Processor Mode

The RX CPU supports two processor modes, supervisor and user. These processor modes enable the realization of a hierarchical CPU resource protection.
Each processor mode imposes a level on rights of access to the CPU resources and the instructions that can be executed. Supervisor mode carries greater rights than those of 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 2.2.2.4, 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)
- 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, and WAIT instructions.

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 RX CPU can handle four types of data: integer, floating-point, bit, and string.
For details, refer to RX Family User's Manual: Software.

2.5 Endian

For the RX 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 LH

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 2.2 shows the relation between the sizes of registers and bit numbers.

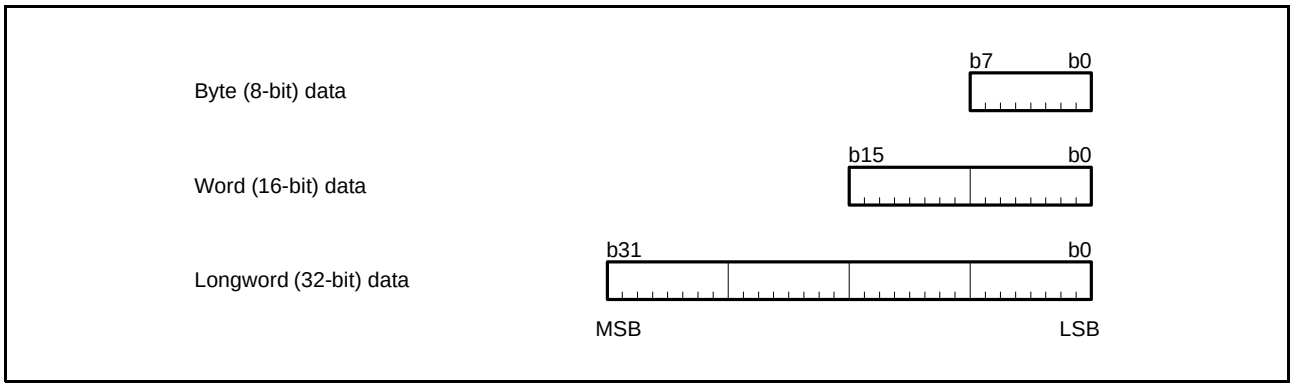


Figure 2.2 Data Arrangement in Registers

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 2.3 shows the arrangement of data in memory.

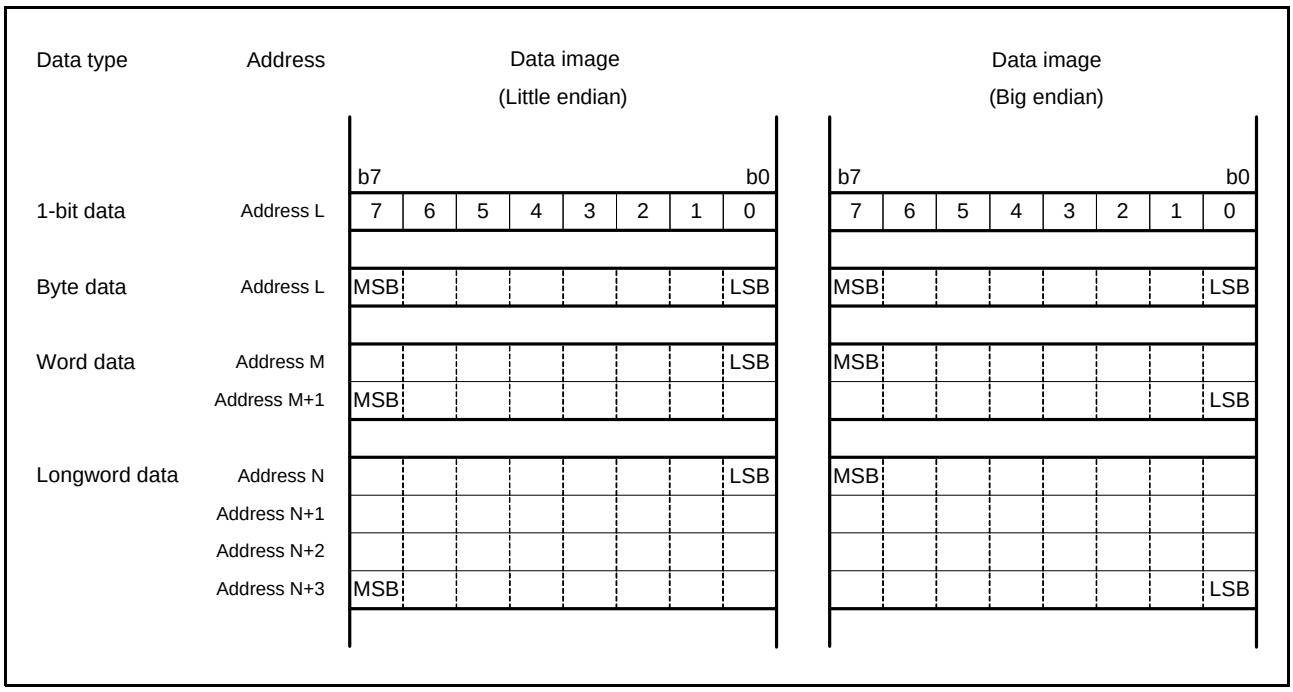


Figure 2.3 Data Arrangement in Memory

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: fixed and relocatable. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, access exception, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFF80h to FFFFFFFFh. Figure 2.4 shows the fixed vector table.

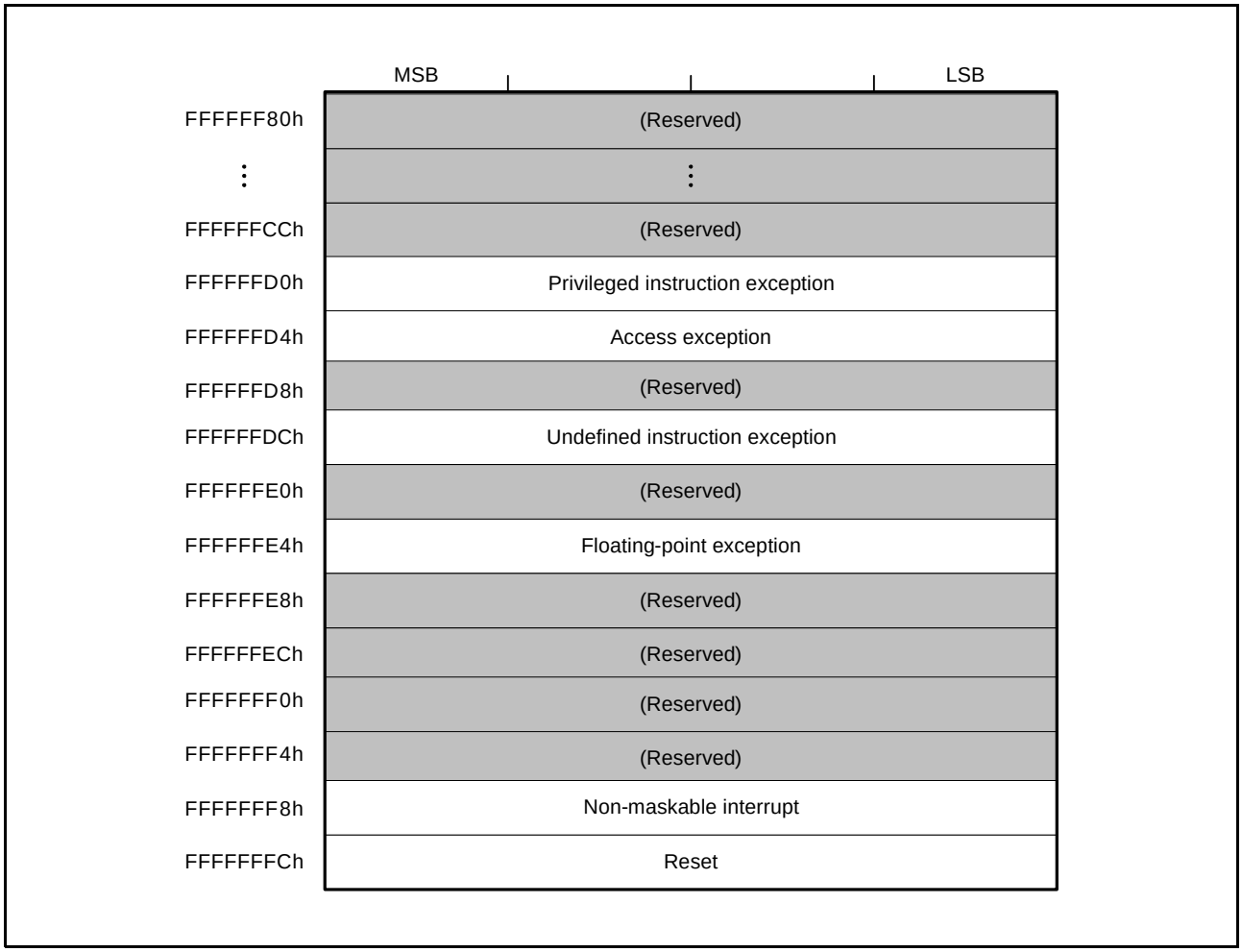


Figure 2.4 Fixed Vector Table

2.6.2 Relocatable Vector Table

The address where the relocatable 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 2.5 shows the relocatable vector table.
Each vector in the relocatable 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.

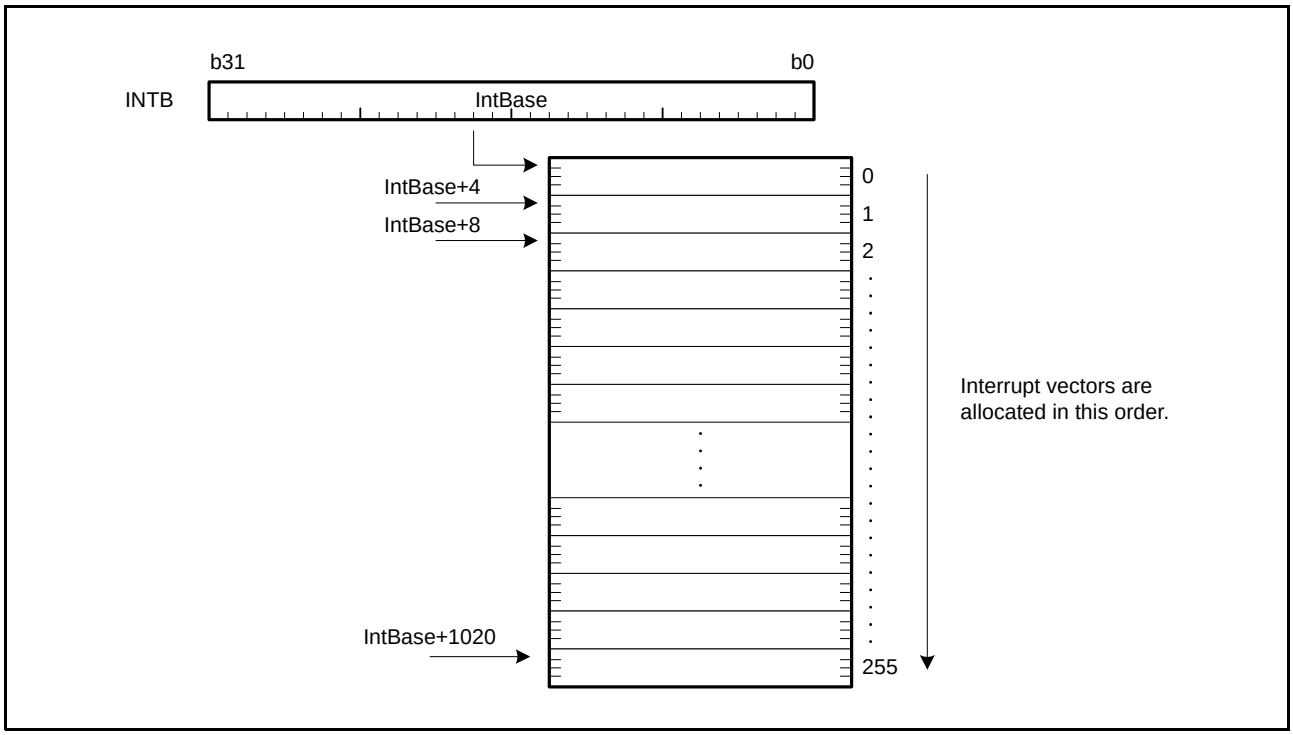


Figure 2.5 Relocatable Vector Table

2.7 Operation of Instructions

2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions

The RMPA instruction and the string-manipulation instructions except the SSTR instruction (that is, SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) may prefetch data from the memory to speed up the read processing. 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 Pipeline

2.8.1 Overview

The RX CPU has 5-stage pipeline structure. The RX CPU instruction is converted into one or more micro-operations, which are then executed in pipeline processing. In the pipeline stage, the IF stage is executed in the unit of instructions, while the D and subsequent stages are executed in the unit of micro-operations.
The operation of pipeline and respective stages is described below.

(1) IF stage (instruction fetch stage)

In the IF stage, the CPU fetches instructions from the memory. As the RX CPU has four 8-byte instruction queues, it fetches instructions until the instruction queue is full, regardless of the completion of decoding in the D (decoding) stage.

(2) D stage (decoding stage)

The CPU decodes instructions in the D stage and converts them into micro-operations. The CPU reads the register information (RF) in this stage and executes a bypass process (BYP) if the result of the preceding instruction will be used in a subsequent instruction. The write of operation result to the register (RW) can be executed with the register reference by using the bypass process.

(3) E stage (execution stage)

Operations and address calculations (OP) are processed in the E stage.

(4) M stage (memory access stage)

Operand memory accesses (OA1, OA2) are processed in the M stage. This stage is used only when the memory is accessed, and is divided into two sub-stages, M1 and M2. The RX CPU enables respective memory accesses for M1 and M2.

- M1 stage (memory-access stage 1)
  Operand memory access (OA1) is processed.
  Store operation: The pipeline processing ends when a write request is received via the bus.
  Load operation: The operation proceeds to the M2 stage when a read request is received via the bus. If a request and load data are received at the same timing (no-wait memory access), the operation proceeds to the WB stage.
- M2 stage (memory-access stage 2)
  Operand memory access (OA2) is processed. The CPU waits for the load data in the M2 stage. When the load data is received, the operation proceeds to the WB stage.

(5) WB stage (write-back stage)

The operation result and the data read from memory are written to the register (RW) in the WB stage. The data read from memory and the other type of data, such as the operation result, can be written to the register in the same clock cycles.

Figure 2.6 shows the pipeline configuration and its operation.

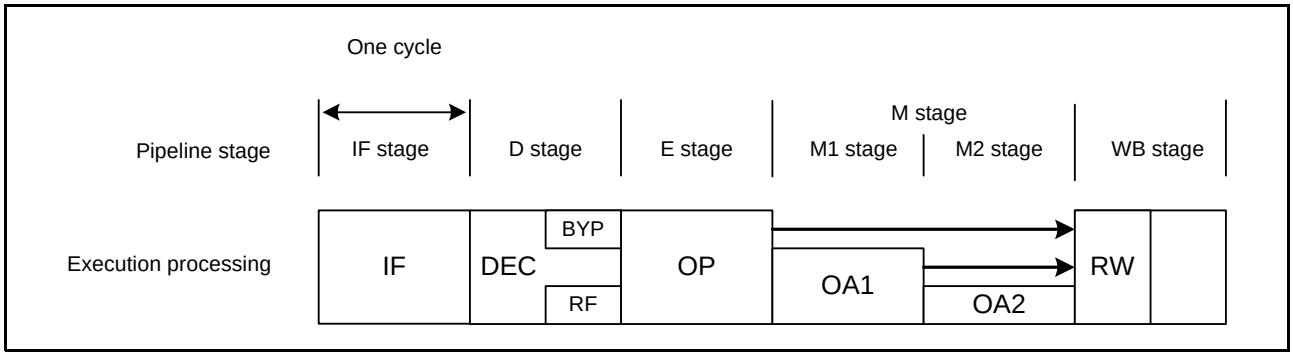


Figure 2.6 Pipeline Configuration and its Operation

2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

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

2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing

The table below lists the instructions that are converted into a single micro-operation. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.13 Instructions that are Converted into a Single Micro-Operation

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register) Except EMUL, EMULU, RMPA, DIV, DIVU and SATR	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR, XOR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB} “#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”	Figure 2.7	1
Arithmetic/logic instructions (division)	• DIV “#IMM, Rd”/“Rs, Rd”	Figure 2.7	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	Figure 2.7	2 to 18*1
Data transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SC*Cnd* “Rd” • {STNZ, STZ} “#IMM, Rd”	Figure 2.7	1
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • POP “Rd”	Figure 2.8	Throughput: 1 Latency: 2*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” • SCCnd “[Rd]”/“dsp[Rd]”	Figure 2.9	1
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	Figure 2.7	1
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Figure 2.18	Branch taken: 3 Branch not taken: 1
Floating-point operation instructions (register-register, immediate-register)	• FCMP “#IMM, Rd”/“Rs, Rs2”	Figure 2.7	1
System manipulation instructions	• {CLRPSW, SETPSW} “flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL“#IMM”	—	1
DSP instructions	• {MACHI, MACLO, MULHI, MULLO} “Rs, Rs2” • {MVFACHI, MVFACMI} “Rd” • {MVTACHI, MVTACLO} “Rs” • RACW“#IMM”	Figure 2.7	1

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. For the number of cycles for throughput and latency, see section 2.8.3, Calculation of the Instruction Processing Time.

Figure 2.7 to Figure 2.9 show the operation of instructions that are converted into a basic single micro-operation.

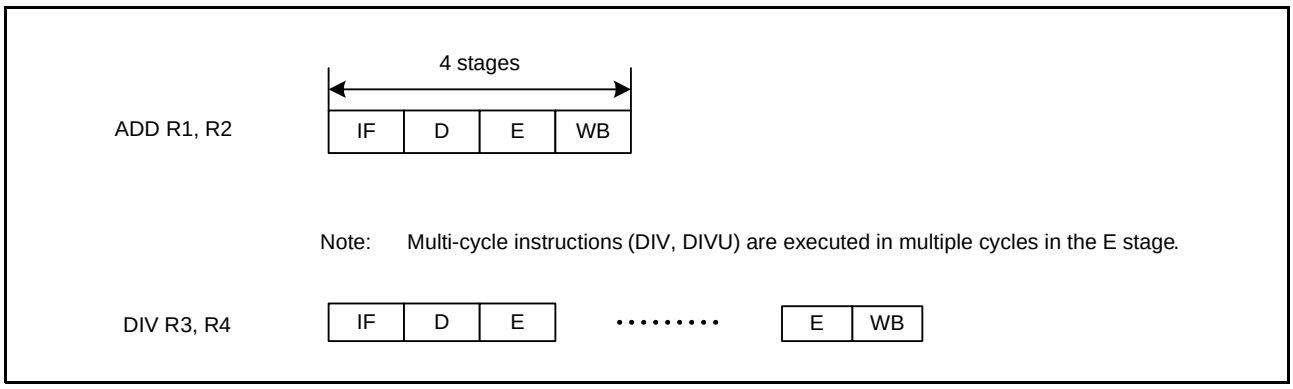


Figure 2.7 Operation for Register-Register, Immediate-Register

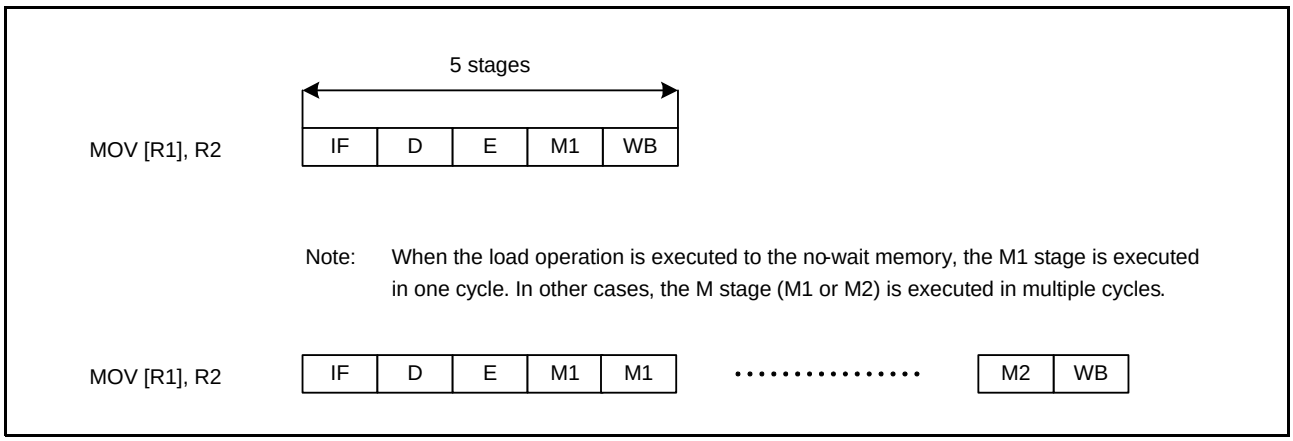


Figure 2.8 Load Operation

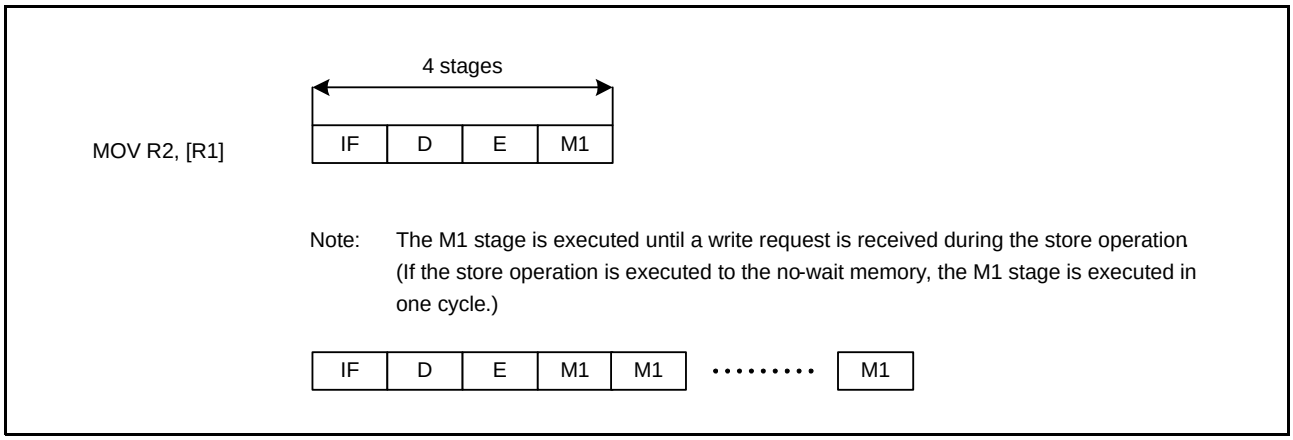


Figure 2.9 Store Operation

2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

The table below lists the instructions that are converted into multiple micro-operations. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (1/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
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”	Figure 2.10	3
Arithmetic/logic instructions (division)	• DIV “[Rs],Rd / dsp[Rs],Rd”	—	5 to 22
	• DIVU“[Rs],Rd / dsp[Rs],Rd”	—	4 to 20
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate)	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	Figure 2.12	2
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (memory source operand)	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	—	4
Arithmetic/logic instructions (multiply-and-accumulate operation)	• RMPA.B	—	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*1
	• RMPA.W	—	6+5×floor(n/2)+4×(n%2) n: Number of processing words*1
	• RMPA.L	—	6+4n n: Number of processing longwords*1
Arithmetic/logic instruction (64-bit signed saturation processing for the RMPA instruction)	• SATR	—	3
Data transfer instructions (memory-memory transfer)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	Figure 2.11	3
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BMCnd “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]	Figure 2.11	3
Transfer instruction (load operation)	• POPC “CR”	—	Throughput: 3 Latency: 4*2
Transfer instruction (save operation of multiple registers)	• PUSHM “Rs-Rs2”	—	n n: Number of registers*3
Transfer instructions (restore operation of multiple registers)	• POPM “Rs-Rs2”	—	Throughput: n Latency: n+1 n: Number of registers*2,*4
Transfer instruction (register-register)	• XCHG “Rs, Rd”	Figure 2.13	2
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	Figure 2.14	2
Branch instructions	• 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*2

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (2/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
String manipulation instructions*5	• SCMPU	—	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*1
	• SMOVB	—	n>3? 6+3×floor(n/4)+3×(n%4): 2+3n n: Number of transfer bytes*1
	• SMOVF, SMOVU	—	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*1
	• SSTR.B	—	2+floor(n/4)+n%4 n: Number of transfer bytes*1
	• SSTR.W	—	2+floor(n/2)+n%2 n: Number of transfer words*1
	• 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*1
	• SUNTIL.W, SWHILE.W	—	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*1
	• SUNTIL.L, SWHILE.L	—	3+3×n n: Number of comparison longwords
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/“Rs, Rd”	Figure 2.15	4
	• FMUL “#IMM, Rd”/“Rs, Rd”	—	3
	• FDIV “#IMM, Rd”/“Rs, Rd”	—	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	—	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	—	6
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	—	3
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	—	5
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	—	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/ “dsp[Rs], Rd”	—	4
System manipulation instructions	• RTE	—	6
	• RTFI	—	3

?: Conditional operator

Note 1. floor (x): Max. integer that is smaller than x
Note 2. For the number of cycles for throughput and latency, see section 2.8.3, Calculation of the Instruction Processing Time.
Note 3. The PUSHM instruction is converted into multiple store operations. The pipeline 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.
Note 4. The POPM instruction is converted into multiple load operations. The pipeline 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 5. 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.

Figure 2.10 to Figure 2.15 show the operation of instructions that are converted into basic multiple micro-operations.
Note: mop: Micro-operation, stall: Pipeline stall

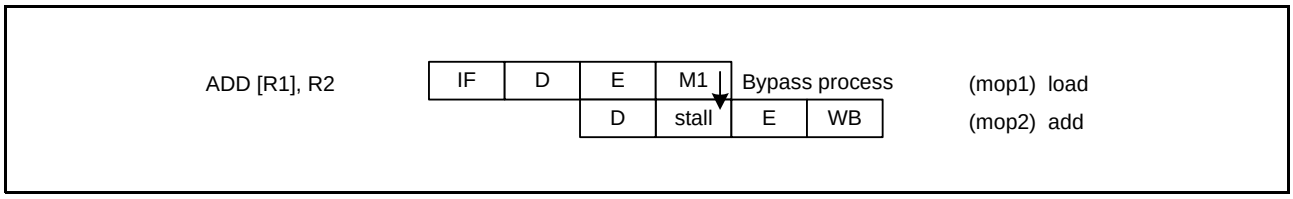


Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)

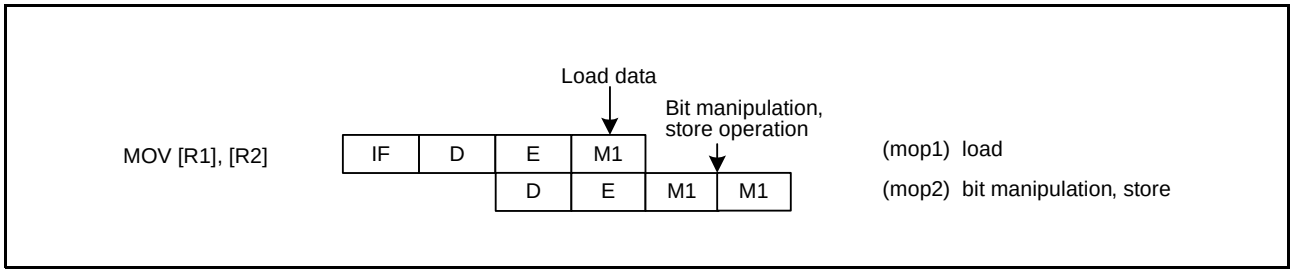


Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

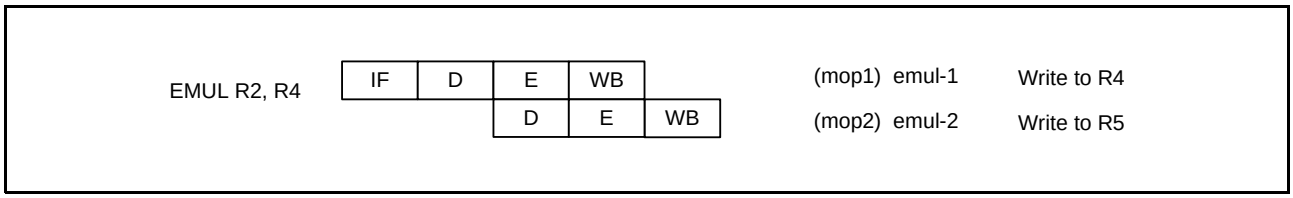


Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

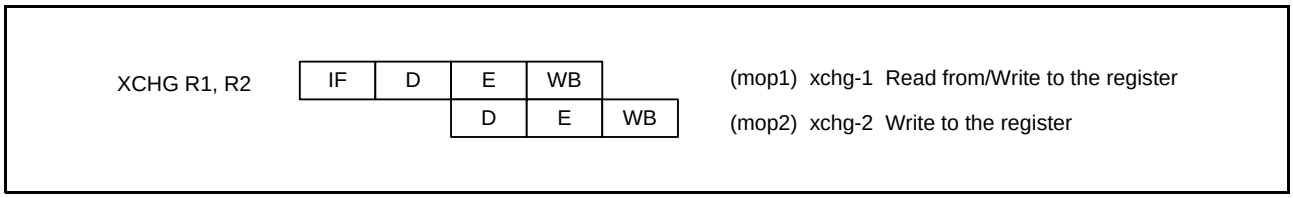


Figure 2.13 XCHG Instruction (Registers)

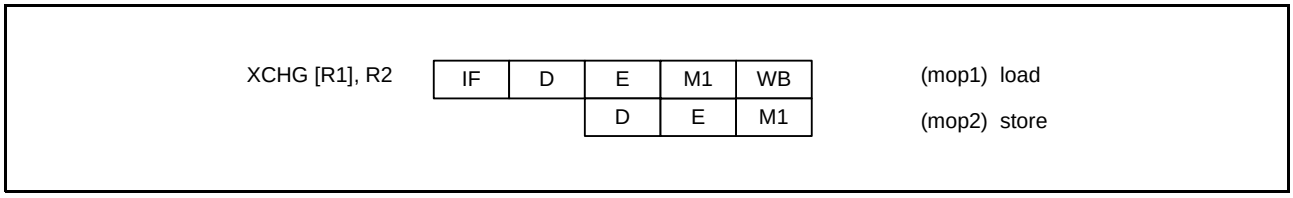


Figure 2.14 XCHG Instruction (Memory Source Operand)

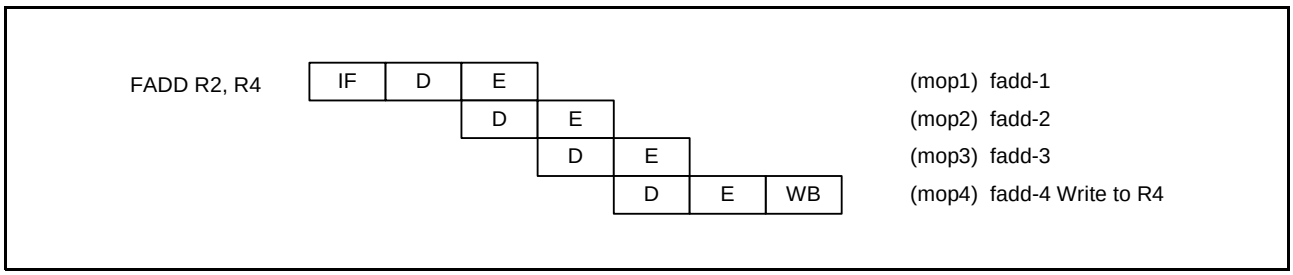


Figure 2.15 Floating-Point Operation Instruction (Register-Register, Immediate-Register)

2.8.2.3 Pipeline Basic Operation

In the ideal pipeline processing, each stage is executed in one cycle, though all instructions may not be pipelined in due to the processing in each stage and the branch execution.
The CPU controls the pipeline stage with the IF stage in the unit of instructions, while the D and subsequent stages in the unit of micro-operations.
The figures below show the pipeline processing of typical cases.
Note: mop: Micro-operation, stall: Pipeline stall

(1) Pipeline Flow with Stalls

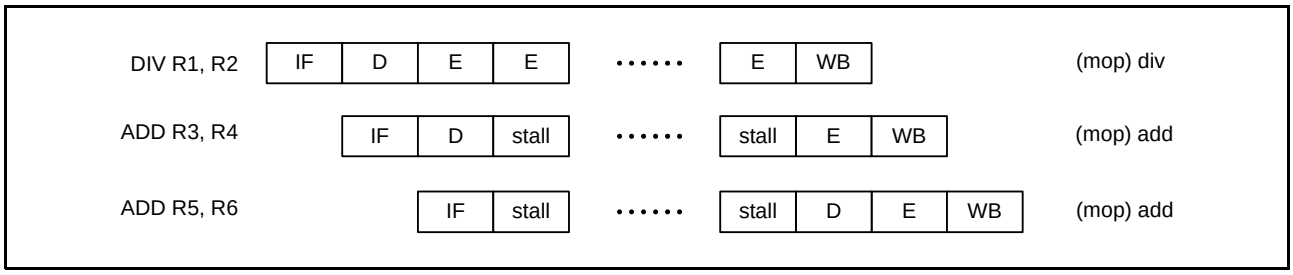


Figure 2.16 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

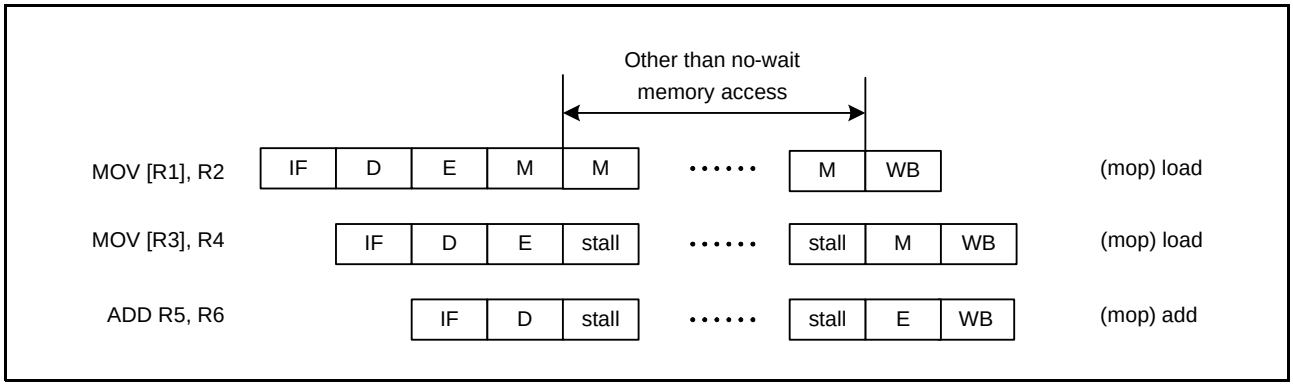


Figure 2.17 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

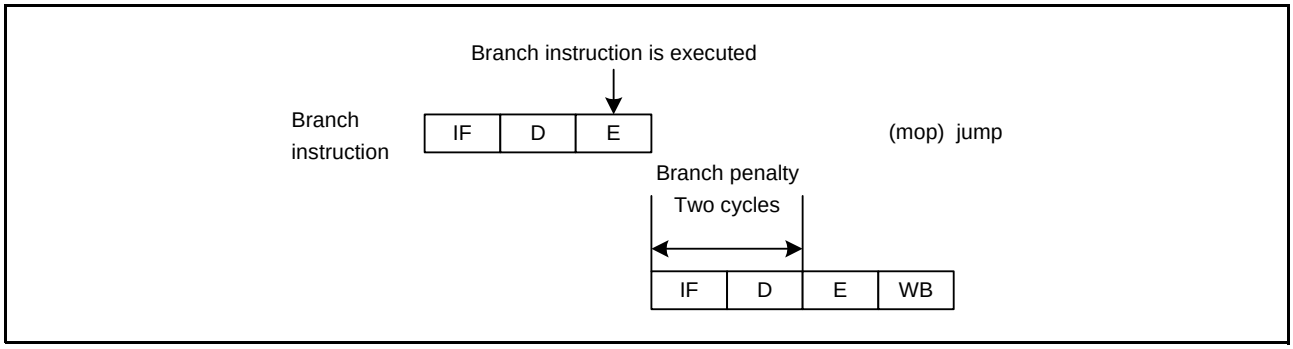


Figure 2.18 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

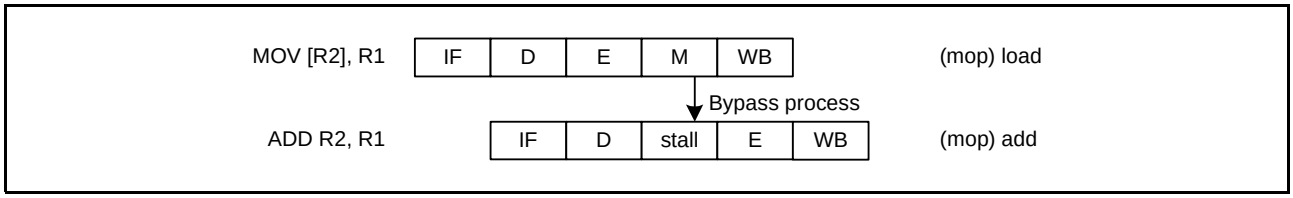


Figure 2.19 When the Subsequent Instruction Uses an Operand Read from the Memory

(2) Pipeline Flow with no Stall

(a) Bypass process

Even when the result of the preceding instruction will be used in a subsequent instruction, the operation processing between registers is pipelined in by the bypass process.

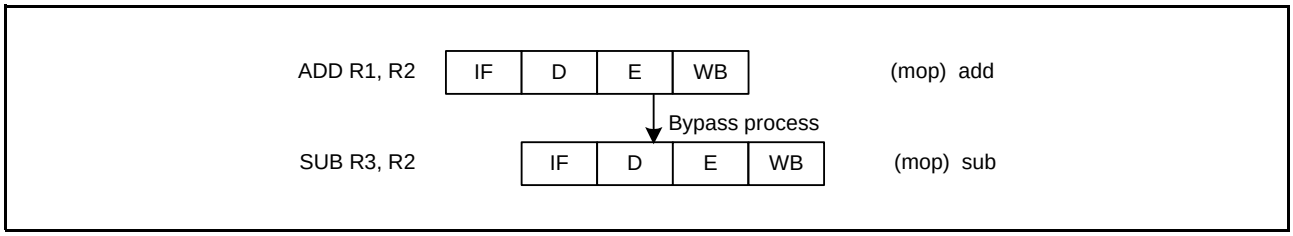


Figure 2.20 Bypass Process

(b) When WB stages for the memory load and for the operation are overlapped

Even when the WB stages for the memory load and for the operation are overlapped, the operation processing is pipelined in, because the load data and the operation result can be written to the register at the same timing.

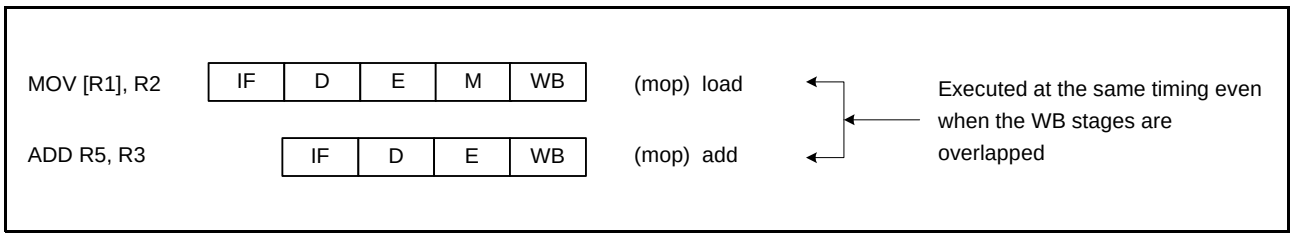


Figure 2.21 When WB Stages for the Memory Load and for the Operation are Overlapped

(c) When subsequent instruction writes to the same register before the end of memory load

Even when the subsequent instruction writes to the same register before the end of memory load, the operation processing is pipelined in, because the WB stage for the memory load is canceled.

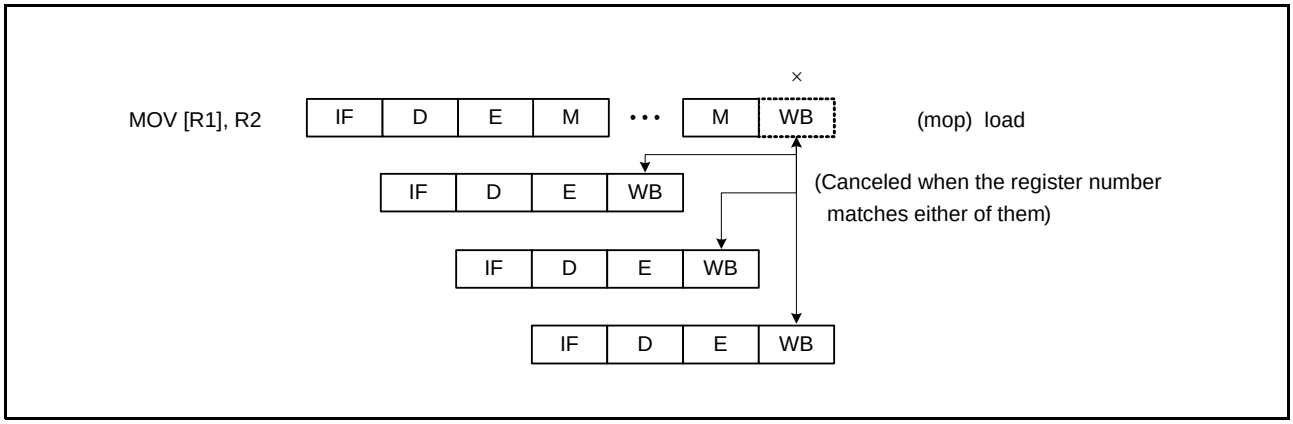


Figure 2.22 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(d) When the load data is not used by the subsequent instruction

When the load data is not used by the subsequent instruction, the subsequent operations are in fact executed earlier and the operation processing ends (out-of-order completion).

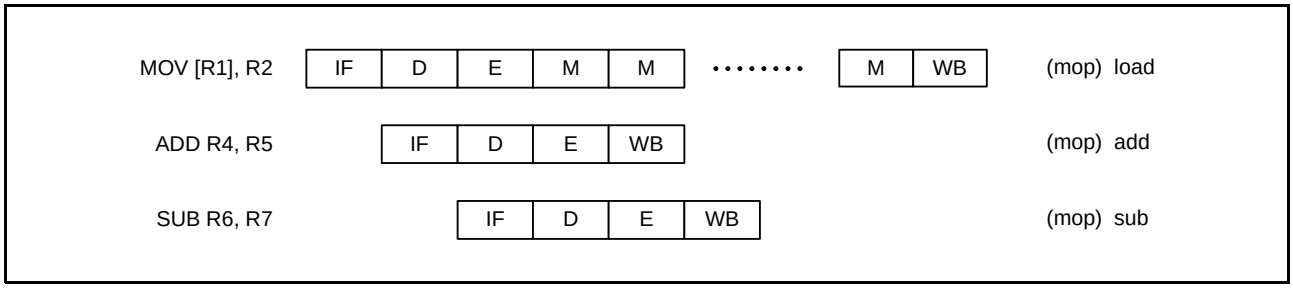


Figure 2.23 When Load Data is not Used by the Subsequent Instruction

2.8.3 Calculation of the Instruction Processing Time

Though the instruction processing time of the CPU varies according to the pipeline processing, the approximate time can be calculated in the following methods.

- Count the number of cycles (see Table 2.13 and Table 2.14)
- When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.
- If the instruction fetch stall is generated, the number of cycles increments.
- Depending on the system configuration, multiple cycles are required for the memory access.

2.8.4 Numbers of Cycles for Response to Interrupts

Table 2.15 lists numbers of cycles taken by processing for response to interrupts.

Table 2.15 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 Number 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 2.15 will be applicable when access to memory from the CPU is processed with no waiting. The RAM and ROM in products of this MCU allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in ROM and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.

For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13, Instructions that are Converted into a Single Micro-Operation, and Table 2.14, Instructions that are Converted into Multiple Micro-Operations.

The timing of interrupt acceptance depends on the state of the pipelines. For more information on this, see section 13.3.1, Acceptance Timing and Saved PC Value.

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 on pins at the time of release from the reset state, and the other is selected by software after release from the reset state.
Table 3.1 shows the relationship between levels on the mode-setting pins (MD, PC7) on release from the reset state 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 (ROM, E2 DataFlash) enabled and the external bus disabled, regardless of the mode in which operation started. Set the SYSCR0.EXBE bit to 1 (external bus enabled) to enable the external bus.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pins

Mode-Setting Pin			SYSCR0 Initial State	
MD*1	PC7*2	Operating Mode	ROME	EXBE
High	—	Single-chip mode	1 (On-chip ROM enabled)	0 (External bus disabled)
Low	Low	Boot mode		
	High	User boot mode		

Note 1. Do not change the level on the MD pin while the MCU is operating.
Note 2. The PC7 pin can also be used as a general port pin.

Table 3.2 gives a list of the operating mode settings that can be made with system control register 0 (SYSCR0). For details on each of the operating modes, see section 3.3, Details of Operating Modes.

Table 3.2 Selection of Operating Modes by Register Setting

SYSCR0		
ROME	EXBE	Operating Mode
0 (On-chip ROM disabled)*1	0 (external bus disabled)	Single-chip mode, user boot mode
1 (On-chip ROM enabled)	0 (external bus disabled)	
0 (On-chip ROM disabled)*1	1 (external bus enabled)	On-chip ROM disabled extended mode
1 (On-chip ROM enabled)	1 (external bus enabled)	On-chip ROM enabled extended mode

Note 1. Once the ROME bit is set to 0, it cannot be reverted to 1.

The endian is selectable in single-chip mode and user boot mode. Endian is set in the given operating mode by using the endian select bits (MDE[2:0]) in the register indicated in Table 3.3. For the correspondence between the setting and endian, see Table 3.4.

Table 3.3 Endian Setting

Operating Mode	Endian Setting
Single-chip mode	Endian is set in the endian selection register (MDES) in the option-setting memory.
User boot mode	Endian is set in the endian selection register (MDEB) in the option-setting memory.

Table 3.4 Selection of Endian

MDE[2:0] Bits	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*1

Note 1. This affects the level on the MD pin at the time of release from the reset state.

Bit	Symbol	Bit Name	Description	R/W
b0	MD	MD Pin Status Flag	0: The MD pin is low. 1: The MD pin is 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

3.2.2 Mode Status Register (MDSR)

Address(es): 0008 0002h

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

Note 1. Depends on the operating mode at startup.

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1.	R
b4 to b1	—	Reserved	These bits are read as 0.	R
b5	UBTS	User Boot Mode Startup Flag	0: Started with a mode except user boot mode. 1: Started with user boot mode.	R
b15 to b6	—	Reserved	These bits are read as 0.	R

3.2.3 System Control Register 0 (SYSCR0)

Address(es): 0008 0006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	EXBE	ROME
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	ROME	On-Chip ROM Enable	0: The on-chip ROM is disabled. 1: The on-chip ROM is enabled.	R/W
b1	EXBE	External Bus Enable	0: The external bus is disabled. 1: The external bus is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	SYSCR0 Key Code	These bits control permission and prohibition of writing to the SYSCR0 register. To modify the SYSCR0 register, write 5Ah 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: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Write data is not retained.

ROME Bit (On-Chip ROM Enable)

The ROME bit enables or disables the on-chip ROM (ROM, E2 DataFlash).
Once cleared to 0, it cannot be reverted to 1.
A 0 should not be written to this bit while a program is being executed from the on-chip ROM. After writing a 0 to this bit, be sure to disable the on-chip ROM by changing the ROME bit to 0 before proceeding with further processing.

EXBE Bit (External Bus Enable)

The EXBE bit enables or disables the external bus.
Do not write 0 to this bit while a program is running from an external address space. Write 0 to this bit after access to the external bus is completed. Furthermore, when an external address space is included in the range of transfer by the DMAC, prohibit DMA transfer before writing 0 to this bit.
After writing to the EXBE bit, confirm that its value has actually changed before proceeding with further processing.
When the EXBE bit is set to 1, the related I/O port settings must also be changed as required. For details, see section 21, Multi-Function Pin Controller (MPC).

3.2.4 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	0	0	0	0	0	0	0	1

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

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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 (RAM disabled) to 1 (RAM enabled), make sure that the RAME bit is 1 before the access.

Even when the RAME bit is cleared to 0, the RAM retains its value. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, see section 41, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, the external bus is disabled (SYSCR0.EXBE bit = 0), all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.
The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state. While the on-chip ROM is enabled (SYSCR0.ROME bit = 1), it can be disabled by clearing the SYSCR0.ROME bit to 0. While the on-chip ROM is disabled (SYSCR0.ROME bit = 0), it cannot be enabled by setting the SYSCR0.ROME bit to 1.
Setting the SYSCR0.EXBE bit to 1 causes a transition to on-chip ROM enabled extended mode or on-chip ROM disabled extended mode where the external bus is available.

3.3.2 On-Chip ROM Enabled Extended Mode

In this mode, the on-chip ROM is enabled (SYSCR0.ROME bit = 1) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 21, Multi-Function Pin Controller (MPC).
After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.
Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM enabled).
Writing 0 to the SYSCR0.ROME bit (on-chip ROM disabled) causes a transition to on-chip ROM disabled extended mode.

3.3.3 On-Chip ROM Disabled Extended Mode

In this mode, the on-chip ROM is disabled (SYSCR0.ROME bit = 0) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 21, Multi-Function Pin Controller (MPC).
After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) and setting the SYSCR0.ROME bit to 0 (on-chip ROM disabled) causes it to make the transition to on-chip ROM disabled extended mode.
In this mode, the on-chip ROM cannot be enabled by setting the SYSCR0.ROME bit to 1.
Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM disabled).

3.3.4 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM, E2 DataFlash) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 40, Flash Memory.
The chip starts up in boot mode if the low level is on both the MD and PC7 pins on release from the reset state.

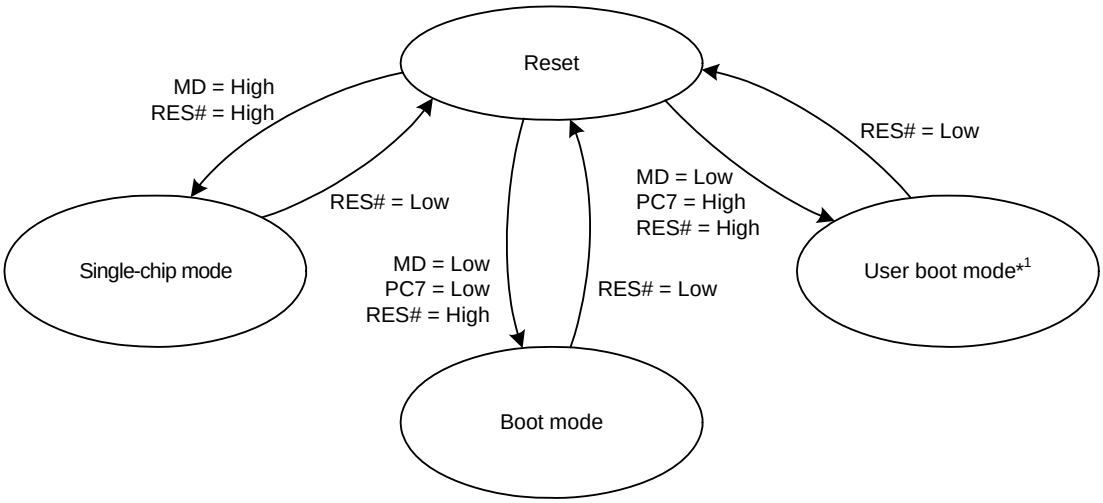
3.3.5 User Boot Mode

In user boot mode, an on-chip flash memory modifying program (user boot program) created by the user operates. After release from the reset state, the chip starts up in a state equivalent to single-chip mode.
After programming the prescribed values for UB code A and the UB code B, the chip starts up in user boot mode if the low level is on the MD pin and the high level is on the PC7 pin on release from the reset state. For UB code A and UB code b, see section 7, Option-Setting Memory.
After the chip has started up in user boot mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.
Note: In user mode, do not make a transition to software standby mode or deep software standby mode.
Note: The setting in the OFS0/OFS1 registers is ineffective in user boot mode, and the value becomes FFFF FFFFh.

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 settings of the MD pin and the PC7 pin.



Note 1. To start the chip in user boot mode, settings for UB code A and UB code B in the option-setting memory are required along with the levels set on the mode-setting pins.

Figure 3.1 Mode-Setting Pin Levels and Operating Modes

3.4.2 Operating Mode Transitions According to Register Setting

Figure 3.2 shows operating mode transitions according to the setting of the ROME and EXBE bits in SYSCR0.

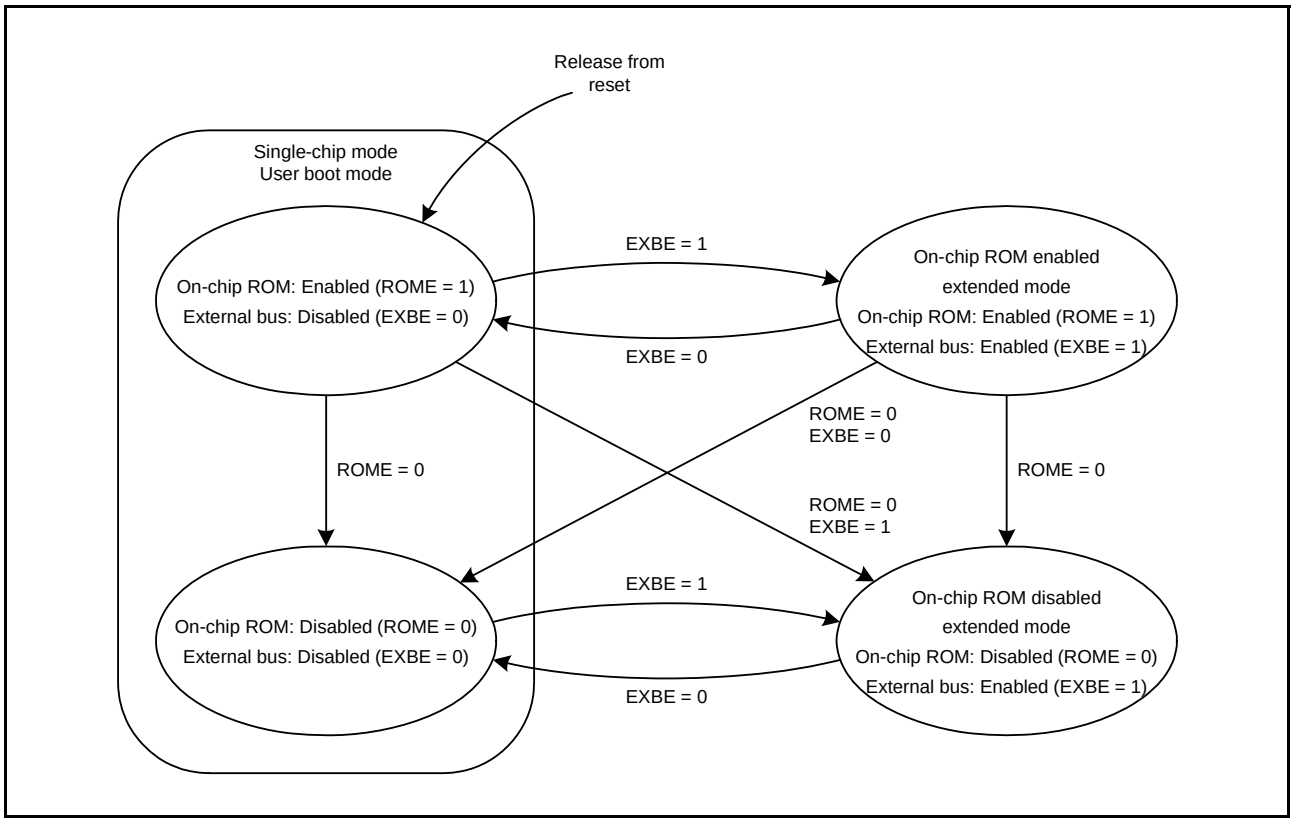
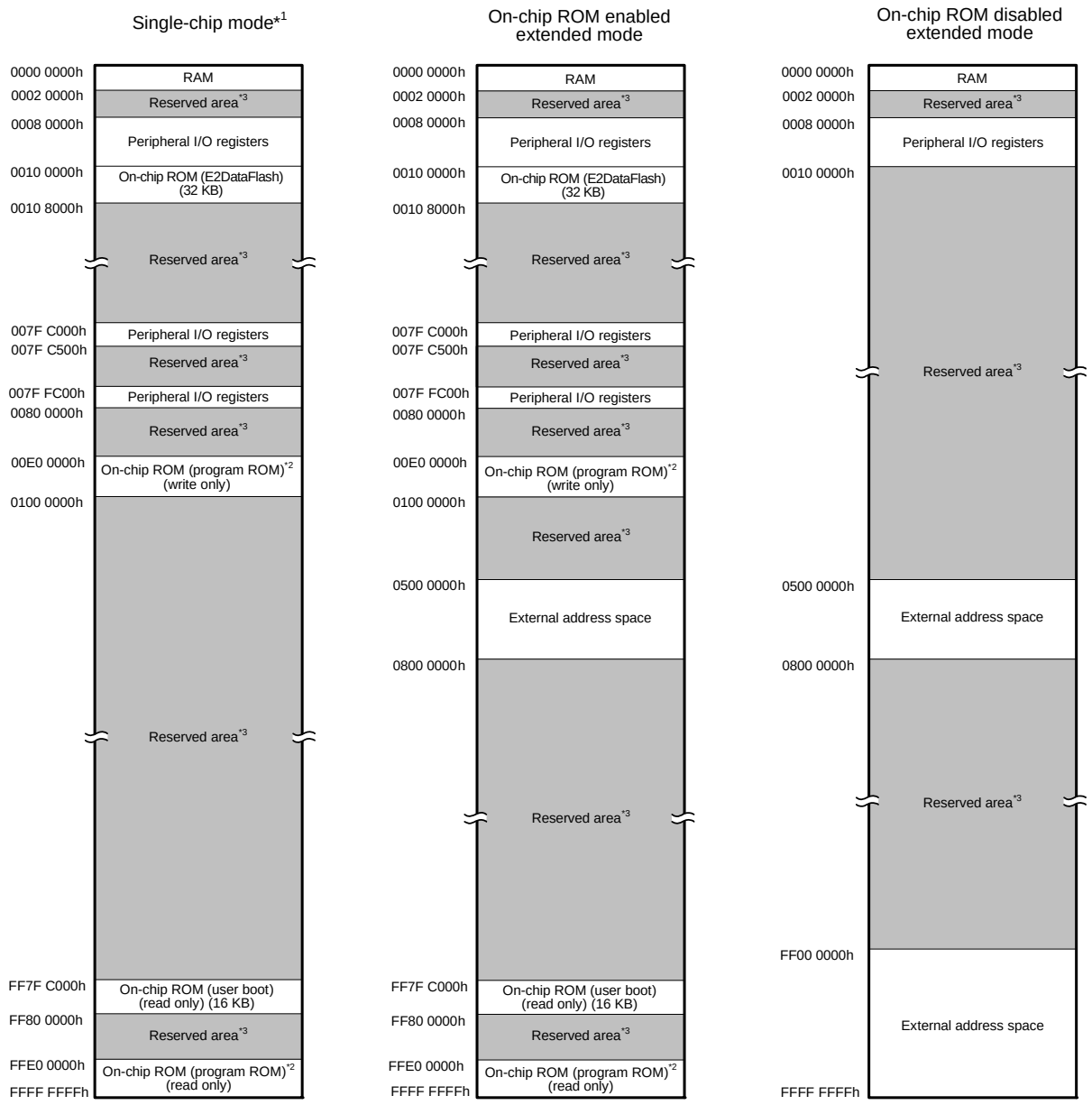


Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

4. Address Space

4.1 Address Space

This LSI has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
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 address space in boot mode and user boot mode is the same as the address space in single-chip mode.
Note 2. The ROM capacity differs depending on the products.

ROM (Byte)			RAM (Byte)	
Capacity	Address (for Reading)	Address (for Writing)	Capacity	Address
2 M	FFE0 0000h to FFFF FFFFh	00E0 0000h to 00FF FFFFh	128 K	0000 0000h to 0001 FFFFh
1.5 M	FFE8 0000h to FFFF FFFFh	00E8 0000h to 00FF FFFFh	128 K	0000 0000h to 0001 FFFFh
1 M	FFF0 0000h to FFFF FFFFh	00F0 0000h to 00FF FFFFh	128 K	0000 0000h to 0001 FFFFh

Note: See Table 1.3, List of Products, for the product type name.

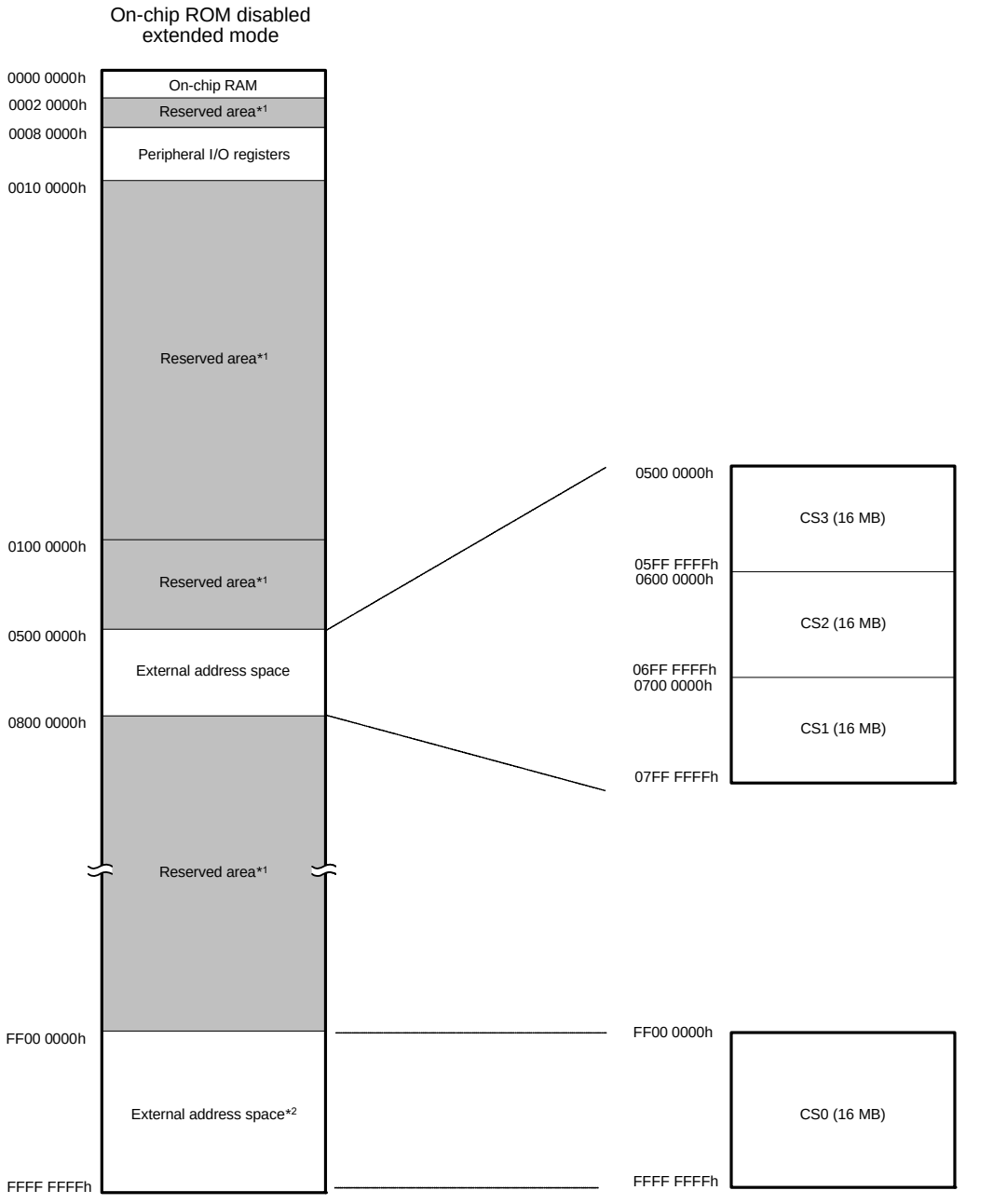
Note 3. Reserved areas should not be accessed.

Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space

The external address space is divided into up to four CS areas (CS0 to CS3), each corresponding to the CSn# signal output from a CSn# (n = 0 to 3) pin.

Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS3) in on-chip ROM disabled extended mode.



Note 1. Reserved areas should not be accessed.

Note 2. The CS0 area is disabled in on-chip ROM enabled extended mode. In this mode, the address space for addresses above 0800 0000h is as shown in figure on this section “Memory Map in Each Operating Mode”.

Figure 4.2 Correspondence between External Address Spaces and CS Areas (In On-Chip ROM Disabled Extended 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) cleared 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 buses 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 or registers for the external bus control unit (except for bus error related registers) 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, BCLK) 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.
In the external bus control unit, the sum of the number of bus cycles for internal main bus 1 and the number of divided clock synchronization cycles will be one cycle of BCLK at a maximum. Therefore, one BCLK is added to the number of access cycles shown in Table 5.1.

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

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK		Operating Modes	85	
0008 0002h	SYSTEM	Mode Status Register	MDSR	16	16	3 ICLK			85	
0008 0006h	SYSTEM	System Control Register 0	SYSCR0	16	16	3 ICLK			86	
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK			87	
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK		Low Power Consumption	204	
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK			205	
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK			207	
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK			209	
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK		Clock Generation Circuit	169	
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK			171	
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK			172	
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK			173	
0008 0030h	SYSTEM	External Bus Clock Control Register	BCKCR	8	8	3 ICLK			174	
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK			175	
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK			176	
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK			177	
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK			178	
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK			179	
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK		Low Power Consumption	210	
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK			212	
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK			213	
0008 00A6h	SYSTEM	PLL Wait Control Register	PLLWTCR	8	8	3 ICLK			214	
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK		Resets	135	
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK			136	
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK		LVDA	153	
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK			153	
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK			154	
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK			154	
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK		Register Write Protection Function	241	
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK		Buses	313	
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK			313	
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK			314	
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK			314	
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK			315	
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACA	367	
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK			367	
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK			368	
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK			370	
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK			371	
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK			372	
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK			374	
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK			377	

Table 5.1 List of I/O Registers (Address Order) (2 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACA	378	
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK			379	
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK			380	
0008 201Fh	DMAC0	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK			382	
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK			367	
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK			367	
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK			368	
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK			370	
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK			371	
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK			372	
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK			374	
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK			378	
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK			379	
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK			380	
0008 205Fh	DMAC1	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK			382	
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK			367	
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK			367	
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK			368	
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK			370	
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK			371	
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK			372	
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK			374	
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK			378	
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK			379	
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK			380	
0008 209Fh	DMAC2	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK			382	
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK			367	
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK			367	
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK			368	
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK			370	
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK			371	
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK			372	
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK			374	
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK			378	
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK			379	
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK			380	
0008 20DFh	DMAC3	DMA Activation Source Flag Control Register	DMCSL	8	8	2 ICLK			382	
0008 2200h	DMAC	DMA Module Activation Register	DMAST	8	8	2 ICLK			383	
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		DTCa	412	
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK			413	
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK			413	
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK			414	
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK			415	
0008 3002h	BSC	CS0 Mode Register	CS0MOD	16	16	1 to 2BCLK		Buses	305	
0008 3004h	BSC	CS0 Wait Control Register 1	CS0WCR1	32	32	1 to 2BCLK			307	
0008 3008h	BSC	CS0 Wait Control Register 2	CS0WCR2	32	32	1 to 2BCLK			310	
0008 3012h	BSC	CS1 Mode Register	CS1MOD	16	16	1 to 2BCLK			305	
0008 3014h	BSC	CS1 Wait Control Register 1	CS1WCR1	32	32	1 to 2BCLK			307	
0008 3018h	BSC	CS1 Wait Control Register 2	CS1WCR2	32	32	1 to 2BCLK			310	

Table 5.1 List of I/O Registers (Address Order) (3 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 3022h	BSC	CS2 Mode Register	CS2MOD	16	16	1 to 2BCLK		Buses	305	
0008 3024h	BSC	CS2 Wait Control Register 1	CS2WCR1	32	32	1 to 2BCLK			307	
0008 3028h	BSC	CS2 Wait Control Register 2	CS2WCR2	32	32	1 to 2BCLK			310	
0008 3032h	BSC	CS3 Mode Register	CS3MOD	16	16	1 to 2BCLK			305	
0008 3034h	BSC	CS3 Wait Control Register 1	CS3WCR1	32	32	1 to 2BCLK			307	
0008 3038h	BSC	CS3 Wait Control Register 2	CS3WCR2	32	32	1 to 2BCLK			310	
0008 3802h	BSC	CS0 Control Register	CS0CR	16	16	1 to 2BCLK			298	
0008 380Ah	BSC	CS0 Recovery Cycle Register	CS0REC	16	16	1 to 2BCLK			300	
0008 3812h	BSC	CS1 Control Register	CS1CR	16	16	1 to 2BCLK			298	
0008 381Ah	BSC	CS1 Recovery Cycle Register	CS1REC	16	16	1 to 2BCLK			300	
0008 3822h	BSC	CS2 Control Register	CS2CR	16	16	1 to 2BCLK			298	
0008 382Ah	BSC	CS2 Recovery Cycle Register	CS2REC	16	16	1 to 2BCLK			300	
0008 3832h	BSC	CS3 Control Register	CS3CR	16	16	1 to 2BCLK			298	
0008 383Ah	BSC	CS3 Recovery Cycle Register	CS3REC	16	16	1 to 2BCLK			300	
0008 3880h	BSC	CS Recovery Cycle Insertion Enable Register	CSRECEN	16	16	1 to 2BCLK			302	
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		MPU	350	
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK			351	
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK			350	
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK			351	
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK			350	
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK			351	
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK			350	
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK			351	
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK			350	
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK			351	
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK			350	
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK			351	
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK			350	
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK			351	
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK			350	
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK			351	
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK			352	
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK			352	
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK			353	
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK			354	
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK			355	
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK			355	
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK			356	
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK			356	
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK			357	
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK			358	
0008 7010h	ICU	Interrupt Request Register 016	IR016	8	8	2 ICLK		ICUb	253	
0008 7015h	ICU	Interrupt Request Register 021	IR021	8	8	2 ICLK			253	
0008 7017h	ICU	Interrupt Request Register 023	IR023	8	8	2 ICLK			253	
0008 701Bh	ICU	Interrupt Request Register 027	IR027	8	8	2 ICLK			253	
0008 701Ch	ICU	Interrupt Request Register 028	IR028	8	8	2 ICLK			253	
0008 701Dh	ICU	Interrupt Request Register 029	IR029	8	8	2 ICLK			253	
0008 701Eh	ICU	Interrupt Request Register 030	IR030	8	8	2 ICLK			253	
0008 701Fh	ICU	Interrupt Request Register 031	IR031	8	8	2 ICLK			253	
0008 7020h	ICU	Interrupt Request Register 032	IR032	8	8	2 ICLK			253	

Table 5.1 List of I/O Registers (Address Order) (4 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 7021h	ICU	Interrupt Request Register 033	IR033	8	8	2 ICLK		ICUb	253	
0008 7022h	ICU	Interrupt Request Register 034	IR034	8	8	2 ICLK			253	
0008 702Ch	ICU	Interrupt Request Register 044	IR044	8	8	2 ICLK			253	
0008 702Dh	ICU	Interrupt Request Register 045	IR045	8	8	2 ICLK			253	
0008 702Eh	ICU	Interrupt Request Register 046	IR046	8	8	2 ICLK			253	
0008 702Fh	ICU	Interrupt Request Register 047	IR047	8	8	2 ICLK			253	
0008 7030h	ICU	Interrupt Request Register 048	IR048	8	8	2 ICLK			253	
0008 7031h	ICU	Interrupt Request Register 049	IR049	8	8	2 ICLK			253	
0008 7032h	ICU	Interrupt Request Register 050	IR050	8	8	2 ICLK			253	
0008 7033h	ICU	Interrupt Request Register 051	IR051	8	8	2 ICLK			253	
0008 7039h	ICU	Interrupt Request Register 057	IR057	8	8	2 ICLK			253	
0008 7040h	ICU	Interrupt Request Register 064	IR064	8	8	2 ICLK			253	
0008 7041h	ICU	Interrupt Request Register 065	IR065	8	8	2 ICLK			253	
0008 7042h	ICU	Interrupt Request Register 066	IR066	8	8	2 ICLK			253	
0008 7043h	ICU	Interrupt Request Register 067	IR067	8	8	2 ICLK			253	
0008 7044h	ICU	Interrupt Request Register 068	IR068	8	8	2 ICLK			253	
0008 7045h	ICU	Interrupt Request Register 069	IR069	8	8	2 ICLK			253	
0008 7046h	ICU	Interrupt Request Register 070	IR070	8	8	2 ICLK			253	
0008 7047h	ICU	Interrupt Request Register 071	IR071	8	8	2 ICLK			253	
0008 7048h	ICU	Interrupt Request Register 072	IR072	8	8	2 ICLK			253	
0008 7049h	ICU	Interrupt Request Register 073	IR073	8	8	2 ICLK			253	
0008 704Ah	ICU	Interrupt Request Register 074	IR074	8	8	2 ICLK			253	
0008 704Bh	ICU	Interrupt Request Register 075	IR075	8	8	2 ICLK			253	
0008 704Ch	ICU	Interrupt Request Register 076	IR076	8	8	2 ICLK			253	
0008 704Dh	ICU	Interrupt Request Register 077	IR077	8	8	2 ICLK			253	Not available in 5-V packages.
0008 704Eh	ICU	Interrupt Request Register 078	IR078	8	8	2 ICLK			253	Not available in 5-V packages.
0008 704Fh	ICU	Interrupt Request Register 079	IR079	8	8	2 ICLK			253	Not available in 5-V packages.
0008 705Eh	ICU	Interrupt Request Register 094	IR094	8	8	2 ICLK			253	Not available in 5-V packages.
0008 705Fh	ICU	Interrupt Request Register 095	IR095	8	8	2 ICLK			253	Not available in 5-V packages.
0008 7062h	ICU	Interrupt Request Register 098	IR098	8	8	2 ICLK			253	
0008 7063h	ICU	Interrupt Request Register 099	IR099	8	8	2 ICLK			253	
0008 7064h	ICU	Interrupt Request Register 100	IR100	8	8	2 ICLK			253	
0008 7065h	ICU	Interrupt Request Register 101	IR101	8	8	2 ICLK			253	
0008 7066h	ICU	Interrupt Request Register 102	IR102	8	8	2 ICLK			253	
0008 7067h	ICU	Interrupt Request Register 103	IR103	8	8	2 ICLK			253	
0008 706Ah	ICU	Interrupt Request Register 106	IR106	8	8	2 ICLK			253	
0008 706Bh	ICU	Interrupt Request Register 107	IR107	8	8	2 ICLK			253	
0008 706Fh	ICU	Interrupt Request Register 111	IR111	8	8	2 ICLK			253	Not available in 5-V packages.
0008 7070h	ICU	Interrupt Request Register 112	IR112	8	8	2 ICLK			253	Not available in 5-V packages.
0008 7071h	ICU	Interrupt Request Register 113	IR113	8	8	2 ICLK			253	Not available in 5-V packages.
0008 7072h	ICU	Interrupt Request Register 114	IR114	8	8	2 ICLK			253	
0008 7073h	ICU	Interrupt Request Register 115	IR115	8	8	2 ICLK			253	

Table 5.1 List of I/O Registers (Address Order) (5 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 7074h	ICU	Interrupt Request Register 116	IR116	8	8	2 ICLK		ICUb	253	
0008 7075h	ICU	Interrupt Request Register 117	IR117	8	8	2 ICLK			253	
0008 7076h	ICU	Interrupt Request Register 118	IR118	8	8	2 ICLK			253	
0008 7077h	ICU	Interrupt Request Register 119	IR119	8	8	2 ICLK			253	
0008 7078h	ICU	Interrupt Request Register 120	IR120	8	8	2 ICLK			253	
0008 7079h	ICU	Interrupt Request Register 121	IR121	8	8	2 ICLK			253	
0008 707Ah	ICU	Interrupt Request Register 122	IR122	8	8	2 ICLK			253	
0008 707Bh	ICU	Interrupt Request Register 123	IR123	8	8	2 ICLK			253	
0008 707Ch	ICU	Interrupt Request Register 124	IR124	8	8	2 ICLK			253	
0008 707Dh	ICU	Interrupt Request Register 125	IR125	8	8	2 ICLK			253	
0008 707Eh	ICU	Interrupt Request Register 126	IR126	8	8	2 ICLK			253	
0008 707Fh	ICU	Interrupt Request Register 127	IR127	8	8	2 ICLK			253	
0008 7080h	ICU	Interrupt Request Register 128	IR128	8	8	2 ICLK			253	
0008 7081h	ICU	Interrupt Request Register 129	IR129	8	8	2 ICLK			253	
0008 7082h	ICU	Interrupt Request Register 130	IR130	8	8	2 ICLK			253	
0008 7083h	ICU	Interrupt Request Register 131	IR131	8	8	2 ICLK			253	
0008 7084h	ICU	Interrupt Request Register 132	IR132	8	8	2 ICLK			253	
0008 7085h	ICU	Interrupt Request Register 133	IR133	8	8	2 ICLK			253	
0008 7086h	ICU	Interrupt Request Register 134	IR134	8	8	2 ICLK			253	
0008 7087h	ICU	Interrupt Request Register 135	IR135	8	8	2 ICLK			253	
0008 7088h	ICU	Interrupt Request Register 136	IR136	8	8	2 ICLK			253	
0008 7089h	ICU	Interrupt Request Register 137	IR137	8	8	2 ICLK			253	
0008 708Ah	ICU	Interrupt Request Register 138	IR138	8	8	2 ICLK			253	
0008 708Bh	ICU	Interrupt Request Register 139	IR139	8	8	2 ICLK			253	
0008 708Ch	ICU	Interrupt Request Register 140	IR140	8	8	2 ICLK			253	
0008 708Dh	ICU	Interrupt Request Register 141	IR141	8	8	2 ICLK			253	
0008 708Eh	ICU	Interrupt Request Register 142	IR142	8	8	2 ICLK			253	
0008 708Fh	ICU	Interrupt Request Register 143	IR143	8	8	2 ICLK			253	
0008 7090h	ICU	Interrupt Request Register 144	IR144	8	8	2 ICLK			253	
0008 7091h	ICU	Interrupt Request Register 145	IR145	8	8	2 ICLK			253	
0008 7092h	ICU	Interrupt Request Register 146	IR146	8	8	2 ICLK			253	
0008 7093h	ICU	Interrupt Request Register 147	IR147	8	8	2 ICLK			253	
0008 7094h	ICU	Interrupt Request Register 148	IR148	8	8	2 ICLK			253	
0008 7095h	ICU	Interrupt Request Register 149	IR149	8	8	2 ICLK			253	
0008 7096h	ICU	Interrupt Request Register 150	IR150	8	8	2 ICLK			253	
0008 7097h	ICU	Interrupt Request Register 151	IR151	8	8	2 ICLK			253	
0008 7098h	ICU	Interrupt Request Register 152	IR152	8	8	2 ICLK			253	
0008 7099h	ICU	Interrupt Request Register 153	IR153	8	8	2 ICLK			253	
0008 709Ah	ICU	Interrupt Request Register 154	IR154	8	8	2 ICLK			253	
0008 709Bh	ICU	Interrupt Request Register 155	IR155	8	8	2 ICLK			253	
0008 709Ch	ICU	Interrupt Request Register 156	IR156	8	8	2 ICLK			253	
0008 709Dh	ICU	Interrupt Request Register 157	IR157	8	8	2 ICLK			253	
0008 709Eh	ICU	Interrupt Request Register 158	IR158	8	8	2 ICLK			253	
0008 709Fh	ICU	Interrupt Request Register 159	IR159	8	8	2 ICLK			253	
0008 70A0h	ICU	Interrupt Request Register 160	IR160	8	8	2 ICLK			253	
0008 70A1h	ICU	Interrupt Request Register 161	IR161	8	8	2 ICLK			253	
0008 70A2h	ICU	Interrupt Request Register 162	IR162	8	8	2 ICLK			253	
0008 70A3h	ICU	Interrupt Request Register 163	IR163	8	8	2 ICLK			253	
0008 70A4h	ICU	Interrupt Request Register 164	IR164	8	8	2 ICLK			253	
0008 70A5h	ICU	Interrupt Request Register 165	IR165	8	8	2 ICLK			253	
0008 70A6h	ICU	Interrupt Request Register 166	IR166	8	8	2 ICLK			253	

Table 5.1 List of I/O Registers (Address Order) (6 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 70A7h	ICU	Interrupt Request Register 167	IR167	8	8	2 ICLK		ICUb	253	
0008 70AAh	ICU	Interrupt Request Register 170	IR170	8	8	2 ICLK			253	
0008 70ABh	ICU	Interrupt Request Register 171	IR171	8	8	2 ICLK			253	
0008 70AEh	ICU	Interrupt Request Register 174	IR174	8	8	2 ICLK			253	
0008 70AFh	ICU	Interrupt Request Register 175	IR175	8	8	2 ICLK			253	
0008 70B0h	ICU	Interrupt Request Register 176	IR176	8	8	2 ICLK			253	
0008 70B1h	ICU	Interrupt Request Register 177	IR177	8	8	2 ICLK			253	
0008 70B2h	ICU	Interrupt Request Register 178	IR178	8	8	2 ICLK			253	
0008 70B3h	ICU	Interrupt Request Register 179	IR179	8	8	2 ICLK			253	
0008 70B4h	ICU	Interrupt Request Register 180	IR180	8	8	2 ICLK			253	
0008 70B5h	ICU	Interrupt Request Register 181	IR181	8	8	2 ICLK			253	
0008 70B6h	ICU	Interrupt Request Register 182	IR182	8	8	2 ICLK			253	
0008 70B7h	ICU	Interrupt Request Register 183	IR183	8	8	2 ICLK			253	
0008 70B8h	ICU	Interrupt Request Register 184	IR184	8	8	2 ICLK			253	
0008 70B9h	ICU	Interrupt Request Register 185	IR185	8	8	2 ICLK			253	
0008 70BAh	ICU	Interrupt Request Register 186	IR186	8	8	2 ICLK			253	
0008 70BBh	ICU	Interrupt Request Register 187	IR187	8	8	2 ICLK			253	
0008 70BCh	ICU	Interrupt Request Register 188	IR188	8	8	2 ICLK			253	
0008 70BDh	ICU	Interrupt Request Register 189	IR189	8	8	2 ICLK			253	
0008 70BEh	ICU	Interrupt Request Register 190	IR190	8	8	2 ICLK			253	
0008 70BFh	ICU	Interrupt Request Register 191	IR191	8	8	2 ICLK			253	
0008 70C0h	ICU	Interrupt Request Register 192	IR192	8	8	2 ICLK			253	
0008 70C1h	ICU	Interrupt Request Register 193	IR193	8	8	2 ICLK			253	
0008 70C2h	ICU	Interrupt Request Register 194	IR194	8	8	2 ICLK			253	
0008 70C3h	ICU	Interrupt Request Register 195	IR195	8	8	2 ICLK			253	
0008 70C4h	ICU	Interrupt Request Register 196	IR196	8	8	2 ICLK			253	
0008 70C5h	ICU	Interrupt Request Register 197	IR197	8	8	2 ICLK			253	
0008 70C6h	ICU	Interrupt Request Register 198	IR198	8	8	2 ICLK			253	
0008 70C7h	ICU	Interrupt Request Register 199	IR199	8	8	2 ICLK			253	
0008 70C8h	ICU	Interrupt Request Register 200	IR200	8	8	2 ICLK			253	
0008 70C9h	ICU	Interrupt Request Register 201	IR201	8	8	2 ICLK			253	
0008 70CAh	ICU	Interrupt Request Register 202	IR202	8	8	2 ICLK			253	
0008 70CBh	ICU	Interrupt Request Register 203	IR203	8	8	2 ICLK			253	
0008 70CCh	ICU	Interrupt Request Register 204	IR204	8	8	2 ICLK			253	
0008 70CDh	ICU	Interrupt Request Register 205	IR205	8	8	2 ICLK			253	
0008 70CEh	ICU	Interrupt Request Register 206	IR206	8	8	2 ICLK			253	
0008 70CFh	ICU	Interrupt Request Register 207	IR207	8	8	2 ICLK			253	
0008 70D0h	ICU	Interrupt Request Register 208	IR208	8	8	2 ICLK			253	
0008 70D1h	ICU	Interrupt Request Register 209	IR209	8	8	2 ICLK			253	
0008 70D2h	ICU	Interrupt Request Register 210	IR210	8	8	2 ICLK			253	
0008 70D3h	ICU	Interrupt Request Register 211	IR211	8	8	2 ICLK			253	
0008 70D4h	ICU	Interrupt Request Register 212	IR212	8	8	2 ICLK			253	
0008 70D5h	ICU	Interrupt Request Register 213	IR213	8	8	2 ICLK			253	
0008 70D6h	ICU	Interrupt Request Register 214	IR214	8	8	2 ICLK			253	
0008 70D7h	ICU	Interrupt Request Register 215	IR215	8	8	2 ICLK			253	
0008 70D8h	ICU	Interrupt Request Register 216	IR216	8	8	2 ICLK			253	
0008 70D9h	ICU	Interrupt Request Register 217	IR217	8	8	2 ICLK			253	
0008 70DAh	ICU	Interrupt Request Register 218	IR218	8	8	2 ICLK			253	
0008 70DBh	ICU	Interrupt Request Register 219	IR219	8	8	2 ICLK			253	
0008 70DCh	ICU	Interrupt Request Register 220	IR220	8	8	2 ICLK			253	
0008 70DDh	ICU	Interrupt Request Register 221	IR221	8	8	2 ICLK			253	

Table 5.1 List of I/O Registers (Address Order) (7 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 70DEh	ICU	Interrupt Request Register 222	IR222	8	8	2 ICLK		ICUb	253	
0008 70DFh	ICU	Interrupt Request Register 223	IR223	8	8	2 ICLK			253	
0008 70E0h	ICU	Interrupt Request Register 224	IR224	8	8	2 ICLK			253	
0008 70E1h	ICU	Interrupt Request Register 225	IR225	8	8	2 ICLK			253	
0008 70E2h	ICU	Interrupt Request Register 226	IR226	8	8	2 ICLK			253	
0008 70E3h	ICU	Interrupt Request Register 227	IR227	8	8	2 ICLK			253	
0008 70E4h	ICU	Interrupt Request Register 228	IR228	8	8	2 ICLK			253	
0008 70E5h	ICU	Interrupt Request Register 229	IR229	8	8	2 ICLK			253	
0008 70E6h	ICU	Interrupt Request Register 230	IR230	8	8	2 ICLK			253	
0008 70E7h	ICU	Interrupt Request Register 231	IR231	8	8	2 ICLK			253	
0008 70E8h	ICU	Interrupt Request Register 232	IR232	8	8	2 ICLK			253	
0008 70E9h	ICU	Interrupt Request Register 233	IR233	8	8	2 ICLK			253	
0008 70EAh	ICU	Interrupt Request Register 234	IR234	8	8	2 ICLK			253	
0008 70EBh	ICU	Interrupt Request Register 235	IR235	8	8	2 ICLK			253	
0008 70ECh	ICU	Interrupt Request Register 236	IR236	8	8	2 ICLK			253	
0008 70EDh	ICU	Interrupt Request Register 237	IR237	8	8	2 ICLK			253	
0008 70EEh	ICU	Interrupt Request Register 238	IR238	8	8	2 ICLK			253	
0008 70EFh	ICU	Interrupt Request Register 239	IR239	8	8	2 ICLK			253	
0008 70F0h	ICU	Interrupt Request Register 240	IR240	8	8	2 ICLK			253	
0008 70F1h	ICU	Interrupt Request Register 241	IR241	8	8	2 ICLK			253	
0008 70F2h	ICU	Interrupt Request Register 242	IR242	8	8	2 ICLK			253	
0008 70F3h	ICU	Interrupt Request Register 243	IR243	8	8	2 ICLK			253	
0008 70F4h	ICU	Interrupt Request Register 244	IR244	8	8	2 ICLK			253	
0008 70F5h	ICU	Interrupt Request Register 245	IR245	8	8	2 ICLK			253	
0008 70F6h	ICU	Interrupt Request Register 246	IR246	8	8	2 ICLK			253	
0008 70F7h	ICU	Interrupt Request Register 247	IR247	8	8	2 ICLK			253	
0008 70F8h	ICU	Interrupt Request Register 248	IR248	8	8	2 ICLK			253	
0008 70F9h	ICU	Interrupt Request Register 249	IR249	8	8	2 ICLK			253	
0008 70FAh	ICU	Interrupt Request Register 250	IR250	8	8	2 ICLK			253	
0008 70FBh	ICU	Interrupt Request Register 251	IR251	8	8	2 ICLK			253	
0008 70FCh	ICU	Interrupt Request Register 252	IR252	8	8	2 ICLK			253	
0008 70FDh	ICU	Interrupt Request Register 253	IR253	8	8	2 ICLK			253	
0008 711Bh	ICU	DTC Activation Enable Register 027	DTCER027	8	8	2 ICLK			258	
0008 711Ch	ICU	DTC Activation Enable Register 028	DTCER028	8	8	2 ICLK			258	
0008 711Dh	ICU	DTC Activation Enable Register 029	DTCER029	8	8	2 ICLK			258	
0008 711Eh	ICU	DTC Activation Enable Register 030	DTCER030	8	8	2 ICLK			258	
0008 711Fh	ICU	DTC Activation Enable Register 031	DTCER031	8	8	2 ICLK			258	
0008 712Dh	ICU	DTC Activation Enable Register 045	DTCER045	8	8	2 ICLK			258	
0008 712Eh	ICU	DTC Activation Enable Register 046	DTCER046	8	8	2 ICLK			258	
0008 7131h	ICU	DTC Activation Enable Register 049	DTCER049	8	8	2 ICLK			258	
0008 7132h	ICU	DTC Activation Enable Register 050	DTCER050	8	8	2 ICLK			258	
0008 7140h	ICU	DTC Activation Enable Register 064	DTCER064	8	8	2 ICLK			258	
0008 7141h	ICU	DTC Activation Enable Register 065	DTCER065	8	8	2 ICLK			258	
0008 7142h	ICU	DTC Activation Enable Register 066	DTCER066	8	8	2 ICLK			258	
0008 7143h	ICU	DTC Activation Enable Register 067	DTCER067	8	8	2 ICLK			258	
0008 7144h	ICU	DTC Activation Enable Register 068	DTCER068	8	8	2 ICLK			258	
0008 7145h	ICU	DTC Activation Enable Register 069	DTCER069	8	8	2 ICLK			258	
0008 7146h	ICU	DTC Activation Enable Register 070	DTCER070	8	8	2 ICLK			258	
0008 7147h	ICU	DTC Activation Enable Register 071	DTCER071	8	8	2 ICLK			258	
0008 7148h	ICU	DTC Activation Enable Register 072	DTCER072	8	8	2 ICLK			258	
0008 7149h	ICU	DTC Activation Enable Register 073	DTCER073	8	8	2 ICLK			258	

Table 5.1 List of I/O Registers (Address Order) (8 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 714Ah	ICU	DTC Activation Enable Register 074	DTCER074	8	8	2 ICLK		ICUb	258	
0008 714Bh	ICU	DTC Activation Enable Register 075	DTCER075	8	8	2 ICLK			258	
0008 714Ch	ICU	DTC Activation Enable Register 076	DTCER076	8	8	2 ICLK			258	
0008 714Dh	ICU	DTC Activation Enable Register 077	DTCER077	8	8	2 ICLK			258	Not available in 5-V packages.
0008 714Eh	ICU	DTC Activation Enable Register 078	DTCER078	8	8	2 ICLK			258	Not available in 5-V packages.
0008 714Fh	ICU	DTC Activation Enable Register 079	DTCER079	8	8	2 ICLK			258	Not available in 5-V packages.
0008 7163h	ICU	DTC Activation Enable Register 099	DTCER099	8	8	2 ICLK			258	
0008 7164h	ICU	DTC Activation Enable Register 100	DTCER100	8	8	2 ICLK			258	
0008 7166h	ICU	DTC Activation Enable Register 102	DTCER102	8	8	2 ICLK			258	
0008 7167h	ICU	DTC Activation Enable Register 103	DTCER103	8	8	2 ICLK			258	
0008 716Ah	ICU	DTC Activation Enable Register 106	DTCER106	8	8	2 ICLK			258	
0008 716Bh	ICU	DTC Activation Enable Register 107	DTCER107	8	8	2 ICLK			258	
0008 716Fh	ICU	DTC Activation Enable Register 111	DTCER111	8	8	2 ICLK			258	Not available in 5-V packages.
0008 7170h	ICU	DTC Activation Enable Register 112	DTCER112	8	8	2 ICLK			258	Not available in 5-V packages.
0008 7172h	ICU	DTC Activation Enable Register 114	DTCER114	8	8	2 ICLK			258	
0008 7173h	ICU	DTC Activation Enable Register 115	DTCER115	8	8	2 ICLK			258	
0008 7174h	ICU	DTC Activation Enable Register 116	DTCER116	8	8	2 ICLK			258	
0008 7175h	ICU	DTC Activation Enable Register 117	DTCER117	8	8	2 ICLK			258	
0008 7179h	ICU	DTC Activation Enable Register 121	DTCER121	8	8	2 ICLK			258	
0008 717Ah	ICU	DTC Activation Enable Register 122	DTCER122	8	8	2 ICLK			258	
0008 717Dh	ICU	DTC Activation Enable Register 125	DTCER125	8	8	2 ICLK			258	
0008 717Eh	ICU	DTC Activation Enable Register 126	DTCER126	8	8	2 ICLK			258	
0008 7181h	ICU	DTC Activation Enable Register 129	DTCER129	8	8	2 ICLK			258	
0008 7182h	ICU	DTC Activation Enable Register 130	DTCER130	8	8	2 ICLK			258	
0008 7183h	ICU	DTC Activation Enable Register 131	DTCER131	8	8	2 ICLK			258	
0008 7184h	ICU	DTC Activation Enable Register 132	DTCER132	8	8	2 ICLK			258	
0008 7186h	ICU	DTC Activation Enable Register 134	DTCER134	8	8	2 ICLK			258	
0008 7187h	ICU	DTC Activation Enable Register 135	DTCER135	8	8	2 ICLK			258	
0008 7188h	ICU	DTC Activation Enable Register 136	DTCER136	8	8	2 ICLK			258	
0008 7189h	ICU	DTC Activation Enable Register 137	DTCER137	8	8	2 ICLK			258	
0008 718Ah	ICU	DTC Activation Enable Register 138	DTCER138	8	8	2 ICLK			258	
0008 718Bh	ICU	DTC Activation Enable Register 139	DTCER139	8	8	2 ICLK			258	
0008 718Ch	ICU	DTC Activation Enable Register 140	DTCER140	8	8	2 ICLK			258	
0008 718Dh	ICU	DTC Activation Enable Register 141	DTCER141	8	8	2 ICLK			258	
0008 718Eh	ICU	DTC Activation Enable Register 142	DTCER142	8	8	2 ICLK			258	
0008 718Fh	ICU	DTC Activation Enable Register 143	DTCER143	8	8	2 ICLK			258	
0008 7190h	ICU	DTC Activation Enable Register 144	DTCER144	8	8	2 ICLK			258	
0008 7191h	ICU	DTC Activation Enable Register 145	DTCER145	8	8	2 ICLK			258	
0008 7193h	ICU	DTC Activation Enable Register 147	DTCER147	8	8	2 ICLK			258	
0008 7194h	ICU	DTC Activation Enable Register 148	DTCER148	8	8	2 ICLK			258	
0008 7197h	ICU	DTC Activation Enable Register 151	DTCER151	8	8	2 ICLK			258	
0008 7198h	ICU	DTC Activation Enable Register 152	DTCER152	8	8	2 ICLK			258	
0008 719Bh	ICU	DTC Activation Enable Register 155	DTCER155	8	8	2 ICLK			258	
0008 719Ch	ICU	DTC Activation Enable Register 156	DTCER156	8	8	2 ICLK			258	
0008 719Dh	ICU	DTC Activation Enable Register 157	DTCER157	8	8	2 ICLK			258	
0008 719Eh	ICU	DTC Activation Enable Register 158	DTCER158	8	8	2 ICLK			258	

Table 5.1 List of I/O Registers (Address Order) (9 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 71A0h	ICU	DTC Activation Enable Register 160	DTCER160	8	8	2 ICLK		ICUb	258	
0008 71A1h	ICU	DTC Activation Enable Register 161	DTCER161	8	8	2 ICLK			258	
0008 71A4h	ICU	DTC Activation Enable Register 164	DTCER164	8	8	2 ICLK			258	
0008 71A5h	ICU	DTC Activation Enable Register 165	DTCER165	8	8	2 ICLK			258	
0008 71AEh	ICU	DTC Activation Enable Register 174	DTCER174	8	8	2 ICLK			258	
0008 71AFh	ICU	DTC Activation Enable Register 175	DTCER175	8	8	2 ICLK			258	
0008 71B1h	ICU	DTC Activation Enable Register 177	DTCER177	8	8	2 ICLK			258	
0008 71B2h	ICU	DTC Activation Enable Register 178	DTCER178	8	8	2 ICLK			258	
0008 71B4h	ICU	DTC Activation Enable Register 180	DTCER180	8	8	2 ICLK			258	
0008 71B5h	ICU	DTC Activation Enable Register 181	DTCER181	8	8	2 ICLK			258	
0008 71B7h	ICU	DTC Activation Enable Register 183	DTCER183	8	8	2 ICLK			258	
0008 71B8h	ICU	DTC Activation Enable Register 184	DTCER184	8	8	2 ICLK			258	
0008 71BBh	ICU	DTC Activation Enable Register 187	DTCER187	8	8	2 ICLK			258	
0008 71BCh	ICU	DTC Activation Enable Register 188	DTCER188	8	8	2 ICLK			258	
0008 71BFh	ICU	DTC Activation Enable Register 191	DTCER191	8	8	2 ICLK			258	
0008 71C0h	ICU	DTC Activation Enable Register 192	DTCER192	8	8	2 ICLK			258	
0008 71C3h	ICU	DTC Activation Enable Register 195	DTCER195	8	8	2 ICLK			258	
0008 71C4h	ICU	DTC Activation Enable Register 196	DTCER196	8	8	2 ICLK			258	
0008 71C6h	ICU	DTC Activation Enable Register 198	DTCER198	8	8	2 ICLK			258	
0008 71C7h	ICU	DTC Activation Enable Register 199	DTCER199	8	8	2 ICLK			258	
0008 71C8h	ICU	DTC Activation Enable Register 200	DTCER200	8	8	2 ICLK			258	
0008 71C9h	ICU	DTC Activation Enable Register 201	DTCER201	8	8	2 ICLK			258	
0008 71CBh	ICU	DTC Activation Enable Register 203	DTCER203	8	8	2 ICLK			258	
0008 71CCh	ICU	DTC Activation Enable Register 204	DTCER204	8	8	2 ICLK			258	
0008 71CFh	ICU	DTC Activation Enable Register 207	DTCER207	8	8	2 ICLK			258	
0008 71D0h	ICU	DTC Activation Enable Register 208	DTCER208	8	8	2 ICLK			258	
0008 71D3h	ICU	DTC Activation Enable Register 211	DTCER211	8	8	2 ICLK			258	
0008 71D4h	ICU	DTC Activation Enable Register 212	DTCER212	8	8	2 ICLK			258	
0008 71D7h	ICU	DTC Activation Enable Register 215	DTCER215	8	8	2 ICLK			258	
0008 71D8h	ICU	DTC Activation Enable Register 216	DTCER216	8	8	2 ICLK			258	
0008 71DBh	ICU	DTC Activation Enable Register 219	DTCER219	8	8	2 ICLK			258	
0008 71DCh	ICU	DTC Activation Enable Register 220	DTCER220	8	8	2 ICLK			258	
0008 71DFh	ICU	DTC Activation Enable Register 223	DTCER223	8	8	2 ICLK			258	
0008 71E0h	ICU	DTC Activation Enable Register 224	DTCER224	8	8	2 ICLK			258	
0008 71E3h	ICU	DTC Activation Enable Register 227	DTCER227	8	8	2 ICLK			258	
0008 71E4h	ICU	DTC Activation Enable Register 228	DTCER228	8	8	2 ICLK			258	
0008 71E7h	ICU	DTC Activation Enable Register 231	DTCER231	8	8	2 ICLK			258	
0008 71E8h	ICU	DTC Activation Enable Register 232	DTCER232	8	8	2 ICLK			258	
0008 71EBh	ICU	DTC Activation Enable Register 235	DTCER235	8	8	2 ICLK			258	
0008 71ECh	ICU	DTC Activation Enable Register 236	DTCER236	8	8	2 ICLK			258	
0008 71EFh	ICU	DTC Activation Enable Register 239	DTCER239	8	8	2 ICLK			258	
0008 71F0h	ICU	DTC Activation Enable Register 240	DTCER240	8	8	2 ICLK			258	
0008 71F7h	ICU	DTC Activation Enable Register 247	DTCER247	8	8	2 ICLK			258	
0008 71F8h	ICU	DTC Activation Enable Register 248	DTCER248	8	8	2 ICLK			258	
0008 71FBh	ICU	DTC Activation Enable Register 251	DTCER251	8	8	2 ICLK			258	
0008 71FCh	ICU	DTC Activation Enable Register 252	DTCER252	8	8	2 ICLK			258	
0008 7202h	ICU	Interrupt Request Enable Register 02	IER02	8	8	2 ICLK			254	
0008 7203h	ICU	Interrupt Request Enable Register 03	IER03	8	8	2 ICLK			254	
0008 7204h	ICU	Interrupt Request Enable Register 04	IER04	8	8	2 ICLK			254	
0008 7205h	ICU	Interrupt Request Enable Register 05	IER05	8	8	2 ICLK			254	
0008 7206h	ICU	Interrupt Request Enable Register 06	IER06	8	8	2 ICLK			254	

Table 5.1 List of I/O Registers (Address Order) (10 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 7207h	ICU	Interrupt Request Enable Register 07	IER07	8	8	2 ICLK		ICUb	254	
0008 7208h	ICU	Interrupt Request Enable Register 08	IER08	8	8	2 ICLK			254	
0008 7209h	ICU	Interrupt Request Enable Register 09	IER09	8	8	2 ICLK			254	
0008 720Bh	ICU	Interrupt Request Enable Register 0B	IER0B	8	8	2 ICLK			254	
0008 720Ch	ICU	Interrupt Request Enable Register 0C	IER0C	8	8	2 ICLK			254	
0008 720Dh	ICU	Interrupt Request Enable Register 0D	IER0D	8	8	2 ICLK			254	
0008 720Eh	ICU	Interrupt Request Enable Register 0E	IER0E	8	8	2 ICLK			254	
0008 720Fh	ICU	Interrupt Request Enable Register 0F	IER0F	8	8	2 ICLK			254	
0008 7210h	ICU	Interrupt Request Enable Register 10	IER10	8	8	2 ICLK			254	
0008 7211h	ICU	Interrupt Request Enable Register 11	IER11	8	8	2 ICLK			254	
0008 7212h	ICU	Interrupt Request Enable Register 12	IER12	8	8	2 ICLK			254	
0008 7213h	ICU	Interrupt Request Enable Register 13	IER13	8	8	2 ICLK			254	
0008 7214h	ICU	Interrupt Request Enable Register 14	IER14	8	8	2 ICLK			254	
0008 7215h	ICU	Interrupt Request Enable Register 15	IER15	8	8	2 ICLK			254	
0008 7216h	ICU	Interrupt Request Enable Register 16	IER16	8	8	2 ICLK			254	
0008 7217h	ICU	Interrupt Request Enable Register 17	IER17	8	8	2 ICLK			254	
0008 7218h	ICU	Interrupt Request Enable Register 18	IER18	8	8	2 ICLK			254	
0008 7219h	ICU	Interrupt Request Enable Register 19	IER19	8	8	2 ICLK			254	
0008 721Ah	ICU	Interrupt Request Enable Register 1A	IER1A	8	8	2 ICLK			254	
0008 721Bh	ICU	Interrupt Request Enable Register 1B	IER1B	8	8	2 ICLK			254	
0008 721Ch	ICU	Interrupt Request Enable Register 1C	IER1C	8	8	2 ICLK			254	
0008 721Dh	ICU	Interrupt Request Enable Register 1D	IER1D	8	8	2 ICLK			254	
0008 721Eh	ICU	Interrupt Request Enable Register 1E	IER1E	8	8	2 ICLK			254	
0008 721Fh	ICU	Interrupt Request Enable Register 1F	IER1F	8	8	2 ICLK			254	
0008 72E0h	ICU	Software Interrupt Activation Register	SWINTR	8	8	2 ICLK			257	
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK			256	
0008 7300h	ICU	Interrupt Source Priority Register 000	IPR000	8	8	2 ICLK			255	
0008 7301h	ICU	Interrupt Source Priority Register 001	IPR001	8	8	2 ICLK			255	
0008 7302h	ICU	Interrupt Source Priority Register 002	IPR002	8	8	2 ICLK			255	
0008 7303h	ICU	Interrupt Source Priority Register 003	IPR003	8	8	2 ICLK			255	
0008 7304h	ICU	Interrupt Source Priority Register 004	IPR004	8	8	2 ICLK			255	
0008 7305h	ICU	Interrupt Source Priority Register 005	IPR005	8	8	2 ICLK			255	
0008 7306h	ICU	Interrupt Source Priority Register 006	IPR006	8	8	2 ICLK			255	
0008 7307h	ICU	Interrupt Source Priority Register 007	IPR007	8	8	2 ICLK			255	
0008 7320h	ICU	Interrupt Source Priority Register 032	IPR032	8	8	2 ICLK			255	
0008 7321h	ICU	Interrupt Source Priority Register 033	IPR033	8	8	2 ICLK			255	
0008 7322h	ICU	Interrupt Source Priority Register 034	IPR034	8	8	2 ICLK			255	
0008 732Ch	ICU	Interrupt Source Priority Register 044	IPR044	8	8	2 ICLK			255	
0008 7330h	ICU	Interrupt Source Priority Register 048	IPR048	8	8	2 ICLK			255	
0008 7339h	ICU	Interrupt Source Priority Register 057	IPR057	8	8	2 ICLK			255	
0008 7340h	ICU	Interrupt Source Priority Register 064	IPR064	8	8	2 ICLK			255	
0008 7341h	ICU	Interrupt Source Priority Register 065	IPR065	8	8	2 ICLK			255	
0008 7342h	ICU	Interrupt Source Priority Register 066	IPR066	8	8	2 ICLK			255	
0008 7343h	ICU	Interrupt Source Priority Register 067	IPR067	8	8	2 ICLK			255	
0008 7344h	ICU	Interrupt Source Priority Register 068	IPR068	8	8	2 ICLK			255	
0008 7345h	ICU	Interrupt Source Priority Register 069	IPR069	8	8	2 ICLK			255	
0008 7346h	ICU	Interrupt Source Priority Register 070	IPR070	8	8	2 ICLK			255	
0008 7347h	ICU	Interrupt Source Priority Register 071	IPR071	8	8	2 ICLK			255	
0008 7348h	ICU	Interrupt Source Priority Register 072	IPR072	8	8	2 ICLK			255	
0008 7349h	ICU	Interrupt Source Priority Register 073	IPR073	8	8	2 ICLK			255	
0008 734Ah	ICU	Interrupt Source Priority Register 074	IPR074	8	8	2 ICLK			255	

Table 5.1 List of I/O Registers (Address Order) (11 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 734Bh	ICU	Interrupt Source Priority Register 075	IPR075	8	8	2 ICLK		ICUb	255	
0008 734Ch	ICU	Interrupt Source Priority Register 076	IPR076	8	8	2 ICLK			255	
0008 734Dh	ICU	Interrupt Source Priority Register 077	IPR077	8	8	2 ICLK			255	Not available in 5-V packages.
0008 734Eh	ICU	Interrupt Source Priority Register 078	IPR078	8	8	2 ICLK			255	Not available in 5-V packages.
0008 734Fh	ICU	Interrupt Source Priority Register 079	IPR079	8	8	2 ICLK			255	Not available in 5-V packages.
0008 735Eh	ICU	Interrupt Source Priority Register 094	IPR094	8	8	2 ICLK			255	Not available in 5-V packages.
0008 735Fh	ICU	Interrupt Source Priority Register 095	IPR095	8	8	2 ICLK			255	Not available in 5-V packages.
0008 7362h	ICU	Interrupt Source Priority Register 098	IPR098	8	8	2 ICLK			255	
0008 7363h	ICU	Interrupt Source Priority Register 099	IPR099	8	8	2 ICLK			255	
0008 7364h	ICU	Interrupt Source Priority Register 100	IPR100	8	8	2 ICLK			255	
0008 7365h	ICU	Interrupt Source Priority Register 101	IPR101	8	8	2 ICLK			255	
0008 7366h	ICU	Interrupt Source Priority Register 102	IPR102	8	8	2 ICLK			255	
0008 7367h	ICU	Interrupt Source Priority Register 103	IPR103	8	8	2 ICLK			255	
0008 736Ah	ICU	Interrupt Source Priority Register 106	IPR106	8	8	2 ICLK			255	
0008 736Bh	ICU	Interrupt Source Priority Register 107	IPR107	8	8	2 ICLK			255	
0008 736Fh	ICU	Interrupt Source Priority Register 111	IPR111	8	8	2 ICLK			255	Not available in 5-V packages.
0008 7372h	ICU	Interrupt Source Priority Register 114	IPR114	8	8	2 ICLK			255	
0008 7376h	ICU	Interrupt Source Priority Register 118	IPR118	8	8	2 ICLK			255	
0008 7379h	ICU	Interrupt Source Priority Register 121	IPR121	8	8	2 ICLK			255	
0008 737Bh	ICU	Interrupt Source Priority Register 123	IPR123	8	8	2 ICLK			255	
0008 737Dh	ICU	Interrupt Source Priority Register 125	IPR125	8	8	2 ICLK			255	
0008 737Fh	ICU	Interrupt Source Priority Register 127	IPR127	8	8	2 ICLK			255	
0008 7381h	ICU	Interrupt Source Priority Register 129	IPR129	8	8	2 ICLK			255	
0008 7385h	ICU	Interrupt Source Priority Register 133	IPR133	8	8	2 ICLK			255	
0008 7386h	ICU	Interrupt Source Priority Register 134	IPR134	8	8	2 ICLK			255	
0008 738Ah	ICU	Interrupt Source Priority Register 138	IPR138	8	8	2 ICLK			255	
0008 738Bh	ICU	Interrupt Source Priority Register 139	IPR139	8	8	2 ICLK			255	
0008 738Eh	ICU	Interrupt Source Priority Register 142	IPR142	8	8	2 ICLK			255	
0008 7392h	ICU	Interrupt Source Priority Register 146	IPR146	8	8	2 ICLK			255	
0008 7393h	ICU	Interrupt Source Priority Register 147	IPR147	8	8	2 ICLK			255	
0008 7395h	ICU	Interrupt Source Priority Register 149	IPR149	8	8	2 ICLK			255	
0008 7397h	ICU	Interrupt Source Priority Register 151	IPR151	8	8	2 ICLK			255	
0008 7399h	ICU	Interrupt Source Priority Register 153	IPR153	8	8	2 ICLK			255	
0008 739Bh	ICU	Interrupt Source Priority Register 155	IPR155	8	8	2 ICLK			255	
0008 739Fh	ICU	Interrupt Source Priority Register 159	IPR159	8	8	2 ICLK			255	
0008 73A0h	ICU	Interrupt Source Priority Register 160	IPR160	8	8	2 ICLK			255	
0008 73A2h	ICU	Interrupt Source Priority Register 162	IPR162	8	8	2 ICLK			255	
0008 73A4h	ICU	Interrupt Source Priority Register 164	IPR164	8	8	2 ICLK			255	
0008 73A6h	ICU	Interrupt Source Priority Register 166	IPR166	8	8	2 ICLK			255	
0008 73AAh	ICU	Interrupt Source Priority Register 170	IPR170	8	8	2 ICLK			255	
0008 73ABh	ICU	Interrupt Source Priority Register 171	IPR171	8	8	2 ICLK			255	
0008 73AEh	ICU	Interrupt Source Priority Register 174	IPR174	8	8	2 ICLK			255	
0008 73B1h	ICU	Interrupt Source Priority Register 177	IPR177	8	8	2 ICLK			255	
0008 73B4h	ICU	Interrupt Source Priority Register 180	IPR180	8	8	2 ICLK			255	
0008 73B7h	ICU	Interrupt Source Priority Register 183	IPR183	8	8	2 ICLK			255	

Table 5.1 List of I/O Registers (Address Order) (12 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 73BAh	ICU	Interrupt Source Priority Register 186	IPR186	8	8	2 ICLK		ICUb	255	
0008 73BEh	ICU	Interrupt Source Priority Register 190	IPR190	8	8	2 ICLK			255	
0008 73C2h	ICU	Interrupt Source Priority Register 194	IPR194	8	8	2 ICLK			255	
0008 73C6h	ICU	Interrupt Source Priority Register 198	IPR198	8	8	2 ICLK			255	
0008 73C7h	ICU	Interrupt Source Priority Register 199	IPR199	8	8	2 ICLK			255	
0008 73C8h	ICU	Interrupt Source Priority Register 200	IPR200	8	8	2 ICLK			255	
0008 73C9h	ICU	Interrupt Source Priority Register 201	IPR201	8	8	2 ICLK			255	
0008 73CAh	ICU	Interrupt Source Priority Register 202	IPR202	8	8	2 ICLK			255	
0008 73CBh	ICU	Interrupt Source Priority Register 203	IPR203	8	8	2 ICLK			255	
0008 73CCh	ICU	Interrupt Source Priority Register 204	IPR204	8	8	2 ICLK			255	
0008 73CDh	ICU	Interrupt Source Priority Register 205	IPR205	8	8	2 ICLK			255	
0008 73CEh	ICU	Interrupt Source Priority Register 206	IPR206	8	8	2 ICLK			255	
0008 73D2h	ICU	Interrupt Source Priority Register 210	IPR210	8	8	2 ICLK			255	
0008 73D6h	ICU	Interrupt Source Priority Register 214	IPR214	8	8	2 ICLK			255	
0008 73DAh	ICU	Interrupt Source Priority Register 218	IPR218	8	8	2 ICLK			255	
0008 73DEh	ICU	Interrupt Source Priority Register 222	IPR222	8	8	2 ICLK			255	
0008 73E2h	ICU	Interrupt Source Priority Register 226	IPR226	8	8	2 ICLK			255	
0008 73E6h	ICU	Interrupt Source Priority Register 230	IPR230	8	8	2 ICLK			255	
0008 73EAh	ICU	Interrupt Source Priority Register 234	IPR234	8	8	2 ICLK			255	
0008 73EEh	ICU	Interrupt Source Priority Register 238	IPR238	8	8	2 ICLK			255	
0008 73F2h	ICU	Interrupt Source Priority Register 242	IPR242	8	8	2 ICLK			255	
0008 73F3h	ICU	Interrupt Source Priority Register 243	IPR243	8	8	2 ICLK			255	
0008 73F4h	ICU	Interrupt Source Priority Register 244	IPR244	8	8	2 ICLK			255	
0008 73F5h	ICU	Interrupt Source Priority Register 245	IPR245	8	8	2 ICLK			255	
0008 73F6h	ICU	Interrupt Source Priority Register 246	IPR246	8	8	2 ICLK			255	
0008 73F7h	ICU	Interrupt Source Priority Register 247	IPR247	8	8	2 ICLK			255	
0008 73F8h	ICU	Interrupt Source Priority Register 248	IPR248	8	8	2 ICLK			255	
0008 73F9h	ICU	Interrupt Source Priority Register 249	IPR249	8	8	2 ICLK			255	
0008 73FAh	ICU	Interrupt Source Priority Register 250	IPR250	8	8	2 ICLK			255	
0008 7400h	ICU	DMAC Activation Request Select Register 0	DMRSR0	8	8	2 ICLK			259	
0008 7404h	ICU	DMAC Activation Request Select Register 1	DMRSR1	8	8	2 ICLK			259	
0008 7408h	ICU	DMAC Activation Request Select Register 2	DMRSR2	8	8	2 ICLK			259	
0008 740Ch	ICU	DMAC Activation Request Select Register 3	DMRSR3	8	8	2 ICLK			259	
0008 7500h	ICU	IRQ Control Register 0	IRQCR0	8	8	2 ICLK			260	
0008 7501h	ICU	IRQ Control Register 1	IRQCR1	8	8	2 ICLK			260	
0008 7502h	ICU	IRQ Control Register 2	IRQCR2	8	8	2 ICLK			260	
0008 7503h	ICU	IRQ Control Register 3	IRQCR3	8	8	2 ICLK			260	
0008 7504h	ICU	IRQ Control Register 4	IRQCR4	8	8	2 ICLK			260	
0008 7505h	ICU	IRQ Control Register 5	IRQCR5	8	8	2 ICLK			260	
0008 7506h	ICU	IRQ Control Register 6	IRQCR6	8	8	2 ICLK			260	
0008 7507h	ICU	IRQ Control Register 7	IRQCR7	8	8	2 ICLK			260	
0008 7508h	ICU	IRQ Control Register 8	IRQCR8	8	8	2 ICLK			260	
0008 7509h	ICU	IRQ Control Register 9	IRQCR9	8	8	2 ICLK			260	
0008 750Ah	ICU	IRQ Control Register 10	IRQCR10	8	8	2 ICLK			260	
0008 750Bh	ICU	IRQ Control Register 11	IRQCR11	8	8	2 ICLK			260	
0008 750Ch	ICU	IRQ Control Register 12	IRQCR12	8	8	2 ICLK			260	
0008 750Dh	CEC	CEC Interrupt Control Register 1	CECINTCR1	8	8	2 ICLK		CEC	1085	Not available in 5-V packages.

Table 5.1 List of I/O Registers (Address Order) (13 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 750Eh	CEC	CEC Interrupt Control Register 2	CECINTCR2	8	8	2 ICLK		CEC	1085	Not available in 5-V packages.
0008 750Fh	CEC	CEC Interrupt Control Register 3	CECINTCR3	8	8	2 ICLK			1085	Not available in 5-V packages.
0008 7510h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		ICUb	261	
0008 7511h	ICU	IRQ Pin Digital Filter Enable Register 1	IRQFLTE1	8	8	2 ICLK			262	
0008 7514h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK			263	
0008 7516h	ICU	IRQ Pin Digital Filter Setting Register 1	IRQFLTC1	16	16	2 ICLK			264	
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK			265	
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK			267	
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK			269	
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK			270	
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK			270	
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK			271	
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 to 3PCLKB	2 ICLK	CMT	844	
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 to 3PCLKB	2 ICLK		845	
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 to 3PCLKB	2 ICLK		845	
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2 to 3PCLKB	2 ICLK		846	
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2 to 3PCLKB	2 ICLK		844	
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2 to 3PCLKB	2 ICLK		845	
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2 to 3PCLKB	2 ICLK		845	
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2 to 3PCLKB	2 ICLK		846	
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2 to 3PCLKB	2 ICLK		846	
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2 to 3PCLKB	2 ICLK	WDTA	852	
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2 to 3PCLKB	2 ICLK		853	
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2 to 3PCLKB	2 ICLK		856	
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2 to 3PCLKB	2 ICLK		857	
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 to 3PCLKB	2 ICLK	IWDTa	867	
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 to 3PCLKB	2 ICLK		868	
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 to 3PCLKB	2 ICLK		871	
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 to 3PCLKB	2 ICLK		872	
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 to 3PCLKB	2 ICLK		873	
0008 80C0h	DA	D/A Data Register 0	DADR0	16	16	2 to 3PCLKB	2 ICLK	DAa	1337	
0008 80C2h	DA	D/A Data Register 1	DADR1	16	16	2 to 3PCLKB	2 ICLK		1337	
0008 80C4h	DA	D/A Control Register	DACR	8	8	2 to 3PCLKB	2 ICLK		1338	
0008 80C5h	DA	DADRm Format Select Register	DADPR	8	8	2 to 3PCLKB	2 ICLK		1339	
0008 8100h	TPU	Timer Start Register	TSTR	8	8	2 to 3PCLKB	2 ICLK	TPUa	749	
0008 8101h	TPU	Timer Synchronous Register	TSYR	8	8	2 to 3PCLKB	2 ICLK		750	
0008 8108h	TPU0	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 8109h	TPU1	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 810Ah	TPU2	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 810Bh	TPU3	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 810Ch	TPU4	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 810Dh	TPU5	Noise Filter Control Register	NFCR	8	8	2 to 3PCLKB	2 ICLK		751	
0008 8110h	TPU0	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	

Table 5.1 List of I/O Registers (Address Order) (14 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8111h	TPU0	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK	TPUa	734	
0008 8112h	TPU0	Timer I/O Control Register H	TIORH	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8113h	TPU0	Timer I/O Control Register L	TIORL	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8114h	TPU0	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8115h	TPU0	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8116h	TPU0	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8118h	TPU0	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		748	
0008 811Ah	TPU0	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 811Ch	TPU0	Timer General Register C	TGRC	16	16	2 to 3PCLKB	2 ICLK		748	
0008 811Eh	TPU0	Timer General Register D	TGRD	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8120h	TPU1	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	
0008 8121h	TPU1	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		734	
0008 8122h	TPU1	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8124h	TPU1	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8125h	TPU1	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8126h	TPU1	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8128h	TPU1	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		748	
0008 812Ah	TPU1	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8130h	TPU2	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	
0008 8131h	TPU2	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		734	
0008 8132h	TPU2	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8134h	TPU2	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8135h	TPU2	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8136h	TPU2	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8138h	TPU2	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		748	
0008 813Ah	TPU2	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8140h	TPU3	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	
0008 8141h	TPU3	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		734	
0008 8142h	TPU3	Timer I/O Control Register H	TIORH	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8143h	TPU3	Timer I/O Control Register L	TIORL	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8144h	TPU3	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8145h	TPU3	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8146h	TPU3	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8148h	TPU3	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		748	
0008 814Ah	TPU3	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 814Ch	TPU3	Timer General Register C	TGRC	16	16	2 to 3PCLKB	2 ICLK		748	
0008 814Eh	TPU3	Timer General Register D	TGRD	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8150h	TPU4	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	
0008 8151h	TPU4	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		734	
0008 8152h	TPU4	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8154h	TPU4	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8155h	TPU4	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8156h	TPU4	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8158h	TPU4	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		748	
0008 815Ah	TPU4	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 8160h	TPU5	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		730	
0008 8161h	TPU5	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		734	
0008 8162h	TPU5	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		735	
0008 8164h	TPU5	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		744	
0008 8165h	TPU5	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		745	
0008 8166h	TPU5	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		748	

Table 5.1 List of I/O Registers (Address Order) (15 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8168h	TPU5	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK	TPUa	748	
0008 816Ah	TPU5	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		748	
0008 81E6h	PPG0	PPG Output Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK	PPG	803	
0008 81E7h	PPG0	PPG Output Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		804	
0008 81E8h	PPG0	Next Data Enable Registers H	NDERH	8	8	2 to 3PCLKB	2 ICLK		797	
0008 81E9h	PPG0	Next Data Enable Registers L	NDERL	8	8	2 to 3PCLKB	2 ICLK		797	
0008 81EAh	PPG0	Output Data Registers H	PODRH	8	8	2 to 3PCLKB	2 ICLK		799	
0008 81EBh	PPG0	Output Data Registers L	PODRL	8	8	2 to 3PCLKB	2 ICLK		799	
0008 81ECh	PPG0	Next Data Registers H	NDRH	8	8	2 to 3PCLKB	2 ICLK		801	
0008 81EDh	PPG0	Next Data Registers L	NDRL	8	8	2 to 3PCLKB	2 ICLK		801	
0008 81EEh	PPG0	Next Data Registers H	NDRH2	8	8	2 to 3PCLKB	2 ICLK		801	
0008 81EFh	PPG0	Next Data Registers L	NDRL2	8	8	2 to 3PCLKB	2 ICLK		801	
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK	TMR	822	
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		822	
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2 to 3PCLKB	2 ICLK		825	
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2 to 3PCLKB	2 ICLK		825	
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2 to 3PCLKB	2 ICLK		821	
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8*1	2 to 3PCLKB	2 ICLK		821	
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2 to 3PCLKB	2 ICLK		821	
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8*1	2 to 3PCLKB	2 ICLK		821	
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2 to 3PCLKB	2 ICLK		820	
0008 8209h	TMR1	Timer Counter	TCNT	8	8*1	2 to 3PCLKB	2 ICLK		820	
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2 to 3PCLKB	2 ICLK		823	
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8*1	2 to 3PCLKB	2 ICLK		823	
0008 820Ch	TMR0	Time Count Start Register	TCSTR	8	8	2 to 3PCLKB	2 ICLK		827	
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		822	
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		822	
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2 to 3PCLKB	2 ICLK		825	
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2 to 3PCLKB	2 ICLK		825	
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2 to 3PCLKB	2 ICLK		821	
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8*1	2 to 3PCLKB	2 ICLK		821	
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2 to 3PCLKB	2 ICLK		821	
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8*1	2 to 3PCLKB	2 ICLK		821	
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2 to 3PCLKB	2 ICLK		820	
0008 8219h	TMR3	Timer Counter	TCNT	8	8*1	2 to 3PCLKB	2 ICLK		820	
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2 to 3PCLKB	2 ICLK		823	
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8*1	2 to 3PCLKB	2 ICLK		823	
0008 821Ch	TMR2	Time Count Start Register	TCSTR	8	8	2 to 3PCLKB	2 ICLK		827	
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 to 3PCLKB	2 ICLK	CRC	1284	
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 to 3PCLKB	2 ICLK		1284	
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 to 3PCLKB	2 ICLK		1285	
0008 8300h	RIIC0	I^2C Bus Control Register 1	ICCR1	8	8	2 to 3PCLKB	2 ICLK	RIIC	1130	
0008 8301h	RIIC0	I^2C Bus Control Register 2	ICCR2	8	8	2 to 3PCLKB	2 ICLK		1132	
0008 8302h	RIIC0	I^2C Bus Mode Register 1	ICMR1	8	8	2 to 3PCLKB	2 ICLK		1136	
0008 8303h	RIIC0	I^2C Bus Mode Register 2	ICMR2	8	8	2 to 3PCLKB	2 ICLK		1137	
0008 8304h	RIIC0	I^2C Bus Mode Register 3	ICMR3	8	8	2 to 3PCLKB	2 ICLK		1139	
0008 8305h	RIIC0	I^2C Bus Function Enable Register	ICFER	8	8	2 to 3PCLKB	2 ICLK		1141	
0008 8306h	RIIC0	I^2C Bus Status Enable Register	ICSER	8	8	2 to 3PCLKB	2 ICLK		1143	
0008 8307h	RIIC0	I^2C Bus Interrupt Enable Register	ICIER	8	8	2 to 3PCLKB	2 ICLK		1145	
0008 8308h	RIIC0	I^2C Bus Status Register 1	ICSR1	8	8	2 to 3PCLKB	2 ICLK		1147	
0008 8309h	RIIC0	I^2C Bus Status Register 2	ICSR2	8	8	2 to 3PCLKB	2 ICLK		1150	

Table 5.1 List of I/O Registers (Address Order) (16 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 to 3PCLKB	2 ICLK	RIIC	1153	
0008 830Ah	RIIC0	Timeout Internal Counter L	TMOCNTL	8	8	2 to 3PCLKB	2 ICLK		1159	
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 830Bh	RIIC0	Timeout Internal Counter U	TMOCNTU	8	8*2	2 to 3PCLKB	2 ICLK		1159	
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 8310h	RIIC0	I2C Bus Bit Rate Low-Level Register	ICBRL	8	8	2 to 3PCLKB	2 ICLK		1155	
0008 8311h	RIIC0	I2C Bus Bit Rate High-Level Register	ICBRH	8	8	2 to 3PCLKB	2 ICLK		1156	
0008 8312h	RIIC0	I2C Bus Transmit Data Register	ICDRT	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8313h	RIIC0	I2C Bus Receive Data Register	ICDRR	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8320h	RIIC1	I2C Bus Control Register 1	ICCR1	8	8	2 to 3PCLKB	2 ICLK		1130	
0008 8321h	RIIC1	I2C Bus Control Register 2	ICCR2	8	8	2 to 3PCLKB	2 ICLK		1132	
0008 8322h	RIIC1	I2C Bus Mode Register 1	ICMR1	8	8	2 to 3PCLKB	2 ICLK		1136	
0008 8323h	RIIC1	I2C Bus Mode Register 2	ICMR2	8	8	2 to 3PCLKB	2 ICLK		1137	
0008 8324h	RIIC1	I2C Bus Mode Register 3	ICMR3	8	8	2 to 3PCLKB	2 ICLK		1139	
0008 8325h	RIIC1	I2C Bus Function Enable Register	ICFER	8	8	2 to 3PCLKB	2 ICLK		1141	
0008 8326h	RIIC1	I2C Bus Status Enable Register	ICSER	8	8	2 to 3PCLKB	2 ICLK		1143	
0008 8327h	RIIC1	I2C Bus Interrupt Enable Register	ICIER	8	8	2 to 3PCLKB	2 ICLK		1145	
0008 8328h	RIIC1	I2C Bus Status Register 1	ICSR1	8	8	2 to 3PCLKB	2 ICLK		1147	
0008 8329h	RIIC1	I2C Bus Status Register 2	ICSR2	8	8	2 to 3PCLKB	2 ICLK		1150	
0008 832Ah	RIIC1	Slave Address Register L0	SARL0	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 832Ah	RIIC1	Timeout Internal Counter L	TMOCNTL	8	8	2 to 3PCLKB	2 ICLK		1159	
0008 832Bh	RIIC1	Slave Address Register U0	SARU0	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 832Bh	RIIC1	Timeout Internal Counter U	TMOCNTU	8	8*2	2 to 3PCLKB	2 ICLK		1159	
0008 832Ch	RIIC1	Slave Address Register L1	SARL1	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 832Dh	RIIC1	Slave Address Register U1	SARU1	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 832Eh	RIIC1	Slave Address Register L2	SARL2	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 832Fh	RIIC1	Slave Address Register U2	SARU2	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 8330h	RIIC1	I2C Bus Bit Rate Low-Level Register	ICBRL	8	8	2 to 3PCLKB	2 ICLK		1155	
0008 8331h	RIIC1	I2C Bus Bit Rate High-Level Register	ICBRH	8	8	2 to 3PCLKB	2 ICLK		1156	
0008 8332h	RIIC1	I2C Bus Transmit Data Register	ICDRT	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8333h	RIIC1	I2C Bus Receive Data Register	ICDRR	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8360h	RIIC3	I2C Bus Control Register 1	ICCR1	8	8	2 to 3PCLKB	2 ICLK		1130	
0008 8361h	RIIC3	I2C Bus Control Register 2	ICCR2	8	8	2 to 3PCLKB	2 ICLK		1132	
0008 8362h	RIIC3	I2C Bus Mode Register 1	ICMR1	8	8	2 to 3PCLKB	2 ICLK		1136	
0008 8363h	RIIC3	I2C Bus Mode Register 2	ICMR2	8	8	2 to 3PCLKB	2 ICLK		1137	
0008 8364h	RIIC3	I2C Bus Mode Register 3	ICMR3	8	8	2 to 3PCLKB	2 ICLK		1139	
0008 8365h	RIIC3	I2C Bus Function Enable Register	ICFER	8	8	2 to 3PCLKB	2 ICLK		1141	
0008 8366h	RIIC3	I2C Bus Status Enable Register	ICSER	8	8	2 to 3PCLKB	2 ICLK		1143	
0008 8367h	RIIC3	I2C Bus Interrupt Enable Register	ICIER	8	8	2 to 3PCLKB	2 ICLK		1145	
0008 8368h	RIIC3	I2C Bus Status Register 1	ICSR1	8	8	2 to 3PCLKB	2 ICLK		1147	
0008 8369h	RIIC3	I2C Bus Status Register 2	ICSR2	8	8	2 to 3PCLKB	2 ICLK		1150	
0008 836Ah	RIIC3	Slave Address Register L0	SARL0	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 836Ah	RIIC3	Timeout Internal Counter L	TMOCNTL	8	8	2 to 3PCLKB	2 ICLK		1159	
0008 836Bh	RIIC3	Slave Address Register U0	SARU0	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 836Bh	RIIC3	Timeout Internal Counter U	TMOCNTU	8	8*2	2 to 3PCLKB	2 ICLK		1159	
0008 836Ch	RIIC3	Slave Address Register L1	SARL1	8	8	2 to 3PCLKB	2 ICLK		1153	
0008 836Dh	RIIC3	Slave Address Register U1	SARU1	8	8	2 to 3PCLKB	2 ICLK		1154	
0008 836Eh	RIIC3	Slave Address Register L2	SARL2	8	8	2 to 3PCLKB	2 ICLK		1153	

Table 5.1 List of I/O Registers (Address Order) (17 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 836Fh	RIIC3	Slave Address Register U2	SARU2	8	8	2 to 3PCLKB	2 ICLK	RIIC	1154	
0008 8370h	RIIC3	I²C Bus Bit Rate Low-Level Register	ICBRL	8	8	2 to 3PCLKB	2 ICLK		1155	
0008 8371h	RIIC3	I²C Bus Bit Rate High-Level Register	ICBRH	8	8	2 to 3PCLKB	2 ICLK		1156	
0008 8372h	RIIC3	I²C Bus Transmit Data Register	ICDRT	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8373h	RIIC3	I²C Bus Receive Data Register	ICDRR	8	8	2 to 3PCLKB	2 ICLK		1158	
0008 8380h	RSPI0	RSPI Control Register	SPCR	8	8	2 to 3PCLKB	2 ICLK	RSPI	1208	
0008 8381h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	2 to 3PCLKB	2 ICLK		1210	
0008 8382h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	2 to 3PCLKB	2 ICLK		1211	
0008 8383h	RSPI0	RSPI Status Register	SPSR	8	8	2 to 3PCLKB	2 ICLK		1212	
0008 8384h	RSPI0	RSPI Data Register	SPDR	32	16, 32	2 to 3PCLKB	2 ICLK		1214	
0008 8388h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	2 to 3PCLKB	2 ICLK		1217	
0008 8389h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	2 to 3PCLKB	2 ICLK		1218	
0008 838Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	2 to 3PCLKB	2 ICLK		1219	
0008 838Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	2 to 3PCLKB	2 ICLK		1220	
0008 838Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	2 to 3PCLKB	2 ICLK		1222	
0008 838Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 to 3PCLKB	2 ICLK		1223	
0008 838Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	2 to 3PCLKB	2 ICLK		1224	
0008 838Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	2 to 3PCLKB	2 ICLK		1225	
0008 8390h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 8392h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 8394h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 8396h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 8398h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 839Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 839Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 839Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83A0h	RSPI1	RSPI Control Register	SPCR	8	8	2 to 3PCLKB	2 ICLK		1208	
0008 83A1h	RSPI1	RSPI Slave Select Polarity Register	SSLP	8	8	2 to 3PCLKB	2 ICLK		1210	
0008 83A2h	RSPI1	RSPI Pin Control Register	SPPCR	8	8	2 to 3PCLKB	2 ICLK		1211	
0008 83A3h	RSPI1	RSPI Status Register	SPSR	8	8	2 to 3PCLKB	2 ICLK		1212	
0008 83A4h	RSPI1	RSPI Data Register	SPDR	32	16, 32	2 to 3PCLKB	2 ICLK		1214	
0008 83A8h	RSPI1	RSPI Sequence Control Register	SPSCR	8	8	2 to 3PCLKB	2 ICLK		1217	
0008 83A9h	RSPI1	RSPI Sequence Status Register	SPSSR	8	8	2 to 3PCLKB	2 ICLK		1218	
0008 83AAh	RSPI1	RSPI Bit Rate Register	SPBR	8	8	2 to 3PCLKB	2 ICLK		1219	
0008 83ABh	RSPI1	RSPI Data Control Register	SPDCR	8	8	2 to 3PCLKB	2 ICLK		1220	
0008 83ACh	RSPI1	RSPI Clock Delay Register	SPCKD	8	8	2 to 3PCLKB	2 ICLK		1222	
0008 83ADh	RSPI1	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 to 3PCLKB	2 ICLK		1223	
0008 83AEh	RSPI1	RSPI Next-Access Delay Register	SPND	8	8	2 to 3PCLKB	2 ICLK		1224	
0008 83AFh	RSPI1	RSPI Control Register 2	SPCR2	8	8	2 to 3PCLKB	2 ICLK		1225	
0008 83B0h	RSPI1	RSPI Command Register 0	SPCMD0	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83B2h	RSPI1	RSPI Command Register 1	SPCMD1	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83B4h	RSPI1	RSPI Command Register 2	SPCMD2	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83B6h	RSPI1	RSPI Command Register 3	SPCMD3	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83B8h	RSPI1	RSPI Command Register 4	SPCMD4	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83BAh	RSPI1	RSPI Command Register 5	SPCMD5	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83BCh	RSPI1	RSPI Command Register 6	SPCMD6	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 83BEh	RSPI1	RSPI Command Register 7	SPCMD7	16	16	2 to 3PCLKB	2 ICLK		1226	
0008 8600h	MTU3	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK	MTU2a	523	
0008 8601h	MTU4	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8602h	MTU3	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		526	

Table 5.1 List of I/O Registers (Address Order) (18 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8603h	MTU4	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK	MTU2a	526	
0008 8604h	MTU3	Timer I/O Control Register H	TIORH	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8605h	MTU3	Timer I/O Control Register L	TIORL	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8606h	MTU4	Timer I/O Control Register H	TIORH	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8607h	MTU4	Timer I/O Control Register L	TIORL	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8608h	MTU3	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 8609h	MTU4	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 860Ah	MTU	Timer Output Master Enable Register	TOER	8	8	2 to 3PCLKB	2 ICLK		554	
0008 860Dh	MTU	Timer Gate Control Register	TGCR	8	8	2 to 3PCLKB	2 ICLK		561	
0008 860Eh	MTU	Timer Output Control Register 1	TOCR1	8	8	2 to 3PCLKB	2 ICLK		555	
0008 860Fh	MTU	Timer Output Control Register 2	TOCR2	8	8	2 to 3PCLKB	2 ICLK		557	
0008 8610h	MTU3	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8612h	MTU4	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8614h	MTU	Timer Cycle Data Register	TCDR	16	16	2 to 3PCLKB	2 ICLK		563	
0008 8616h	MTU	Timer Dead Time Data Register	TDDR	16	16	2 to 3PCLKB	2 ICLK		562	
0008 8618h	MTU3	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		549	
0008 861Ah	MTU3	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		549	
0008 861Ch	MTU4	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		549	
0008 861Eh	MTU4	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8620h	MTU	Timer Subcounter	TCNTS	16	16	2 to 3PCLKB	2 ICLK		562	
0008 8622h	MTU	Timer Cycle Buffer Register	TCBR	16	16	2 to 3PCLKB	2 ICLK		563	
0008 8624h	MTU3	Timer General Register C	TGRC	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8626h	MTU3	Timer General Register D	TGRD	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8628h	MTU4	Timer General Register C	TGRC	16	16	2 to 3PCLKB	2 ICLK		549	
0008 862Ah	MTU4	Timer General Register D	TGRD	16	16	2 to 3PCLKB	2 ICLK		549	
0008 862Ch	MTU3	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		543	
0008 862Dh	MTU4	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		543	
0008 8630h	MTU	Timer Interrupt Skipping Set Register	TITCR	8	8	2 to 3PCLKB	2 ICLK		564	
0008 8631h	MTU	Timer Interrupt Skipping Counter	TITCNT	8	8	2 to 3PCLKB	2 ICLK		566	
0008 8632h	MTU	Timer Buffer Transfer Set Register	TBTER	8	8	2 to 3PCLKB	2 ICLK		567	
0008 8634h	MTU	Timer Dead Time Enable Register	TDER	8	8	2 to 3PCLKB	2 ICLK		568	
0008 8636h	MTU	Timer Output Level Buffer Register	TOLBR	8	8	2 to 3PCLKB	2 ICLK		560	
0008 8638h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 to 3PCLKB	2 ICLK		544	
0008 8639h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 to 3PCLKB	2 ICLK		544	
0008 8640h	MTU4	Timer A/D Converter Start Request Control Register	TADCR	16	16	2 to 3PCLKB	2 ICLK		546	
0008 8644h	MTU4	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16	2 to 3PCLKB	2 ICLK		547	
0008 8646h	MTU4	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	2 to 3PCLKB	2 ICLK		547	
0008 8648h	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	2 to 3PCLKB	2 ICLK		548	
0008 864Ah	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	2 to 3PCLKB	2 ICLK		548	
0008 8660h	MTU	Timer Waveform Control Register	TWCR	8	8, 16	2 to 3PCLKB	2 ICLK		569	
0008 8680h	MTU	Timer Start Register	TSTR	8	8, 16	2 to 3PCLKB	2 ICLK		550	
0008 8681h	MTU	Timer Synchronous Register	TSYR	8	8, 16	2 to 3PCLKB	2 ICLK		552	
0008 8684h	MTU	Timer Read/Write Enable Register	TRWER	8	8, 16	2 to 3PCLKB	2 ICLK		553	
0008 8690h	MTU0	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK		570	
0008 8691h	MTU1	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK		570	
0008 8692h	MTU2	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK		570	
0008 8693h	MTU3	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK		570	

Table 5.1 List of I/O Registers (Address Order) (19 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8694h	MTU4	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK	MTU2a	570	
0008 8695h	MTU5	Noise Filter Control Register	NFCR	8	8, 16	2 to 3PCLKB	2 ICLK		570	
0008 8700h	MTU0	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8701h	MTU0	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		526	
0008 8702h	MTU0	Timer I/O Control Register H	TIORH	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8703h	MTU0	Timer I/O Control Register L	TIORL	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8704h	MTU0	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 8705h	MTU0	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		543	
0008 8706h	MTU0	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8708h	MTU0	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		549	
0008 870Ah	MTU0	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		549	
0008 870Ch	MTU0	Timer General Register C	TGRC	16	16	2 to 3PCLKB	2 ICLK		549	
0008 870Eh	MTU0	Timer General Register D	TGRD	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8720h	MTU0	Timer General Register E	TGRE	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8722h	MTU0	Timer General Register F	TGRF	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8724h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	2 to 3PCLKB	2 ICLK		540	
0008 8726h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 to 3PCLKB	2 ICLK		544	
0008 8780h	MTU1	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8781h	MTU1	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		526	
0008 8782h	MTU1	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8784h	MTU1	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 8785h	MTU1	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		543	
0008 8786h	MTU1	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8788h	MTU1	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		549	
0008 878Ah	MTU1	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8790h	MTU1	Timer Input Capture Control Register	TICCR	8	8	2 to 3PCLKB	2 ICLK		545	
0008 8800h	MTU2	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8801h	MTU2	Timer Mode Register	TMDR	8	8	2 to 3PCLKB	2 ICLK		526	
0008 8802h	MTU2	Timer I/O Control Register	TIOR	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8804h	MTU2	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 8805h	MTU2	Timer Status Register	TSR	8	8	2 to 3PCLKB	2 ICLK		543	
0008 8806h	MTU2	Timer Counter	TCNT	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8808h	MTU2	Timer General Register A	TGRA	16	16	2 to 3PCLKB	2 ICLK		549	
0008 880Ah	MTU2	Timer General Register B	TGRB	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8880h	MTU5	Timer Counter U	TCNTU	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8882h	MTU5	Timer General Register U	TGRU	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8884h	MTU5	Timer Control Register U	TCRU	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8886h	MTU5	Timer I/O Control Register U	TIORU	8	8	2 to 3PCLKB	2 ICLK		528	
0008 8890h	MTU5	Timer Counter V	TCNTV	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8892h	MTU5	Timer General Register V	TGRV	16	16	2 to 3PCLKB	2 ICLK		549	
0008 8894h	MTU5	Timer Control Register V	TCRV	8	8	2 to 3PCLKB	2 ICLK		523	
0008 8896h	MTU5	Timer I/O Control Register V	TIORV	8	8	2 to 3PCLKB	2 ICLK		528	
0008 88A0h	MTU5	Timer Counter W	TCNTW	16	16	2 to 3PCLKB	2 ICLK		549	
0008 88A2h	MTU5	Timer General Register W	TGRW	16	16	2 to 3PCLKB	2 ICLK		549	
0008 88A4h	MTU5	Timer Control Register W	TCRW	8	8	2 to 3PCLKB	2 ICLK		523	
0008 88A6h	MTU5	Timer I/O Control Register W	TIORW	8	8	2 to 3PCLKB	2 ICLK		528	
0008 88B2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	2 to 3PCLKB	2 ICLK		540	
0008 88B4h	MTU5	Timer Start Register	TSTR	8	8	2 to 3PCLKB	2 ICLK		550	
0008 88B6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	2 to 3PCLKB	2 ICLK		539	
0008 8900h	POE	Input Level Control/Status Register 1	ICSR1	16	16	2 to 3PCLKB	2 ICLK	POE2a	712	
0008 8902h	POE	Output Level Control/Status Register 1	OCSR1	16	16	2 to 3PCLKB	2 ICLK		714	

Table 5.1 List of I/O Registers (Address Order) (20 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8908h	POE	Input Level Control/Status Register 2	ICSR2	16	16	2 to 3PCLKB	2 ICLK	POE2a	715	
0008 890Ah	POE	Software Port Output Enable Register	SPOER	8	8	2 to 3PCLKB	2 ICLK		716	
0008 890Bh	POE	Port Output Enable Control Register 1	POECR1	8	8	2 to 3PCLKB	2 ICLK		717	
0008 890Ch	POE	Port Output Enable Control Register 2	POECR2	8	8	2 to 3PCLKB	2 ICLK		718	
0008 890Eh	POE	Input Level Control/Status Register 3	ICSR3	16	16	2 to 3PCLKB	2 ICLK		719	
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 to 3PCLKB	2 ICLK	S12ADB	1299	
0008 9004h	S12AD	A/D Channel Select Register A	ADANSA	16	16	2 to 3PCLKB	2 ICLK		1302	
0008 9008h	S12AD	A/D-Converted Value Addition Mode Select Register	ADADS	16	16	2 to 3PCLKB	2 ICLK		1303	
0008 900Ch	S12AD	A/D-Converted Value Addition Count Select Register	ADADC	8	8	2 to 3PCLKB	2 ICLK		1304	
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 to 3PCLKB	2 ICLK		1305	
0008 9010h	S12AD	A/D Start Trigger Select Register	ADSTRGR	16	16	2 to 3PCLKB	2 ICLK		1307	
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2 to 3PCLKB	2 ICLK		1309	
0008 9014h	S12AD	A/D Channel Select Register B	ADANSB	16	16	2 to 3PCLKB	2 ICLK		1302	
0008 9018h	S12AD	A/D Data-Doubling Register	ADDBLDR	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 to 3PCLKB	2 ICLK		1297	
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2 to 3PCLKB	2 ICLK		1298	
0008 9020h	S12AD	A/D Data Registers 0	ADDR0	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9022h	S12AD	A/D Data Registers 1	ADDR1	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9024h	S12AD	A/D Data Registers 2	ADDR2	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9026h	S12AD	A/D Data Registers 3	ADDR3	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9028h	S12AD	A/D Data Registers 4	ADDR4	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 902Ah	S12AD	A/D Data Registers 5	ADDR5	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 902Ch	S12AD	A/D Data Registers 6	ADDR6	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 902Eh	S12AD	A/D Data Registers 7	ADDR7	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9030h	S12AD	A/D Data Registers 8	ADDR8	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9032h	S12AD	A/D Data Registers 9	ADDR9	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9034h	S12AD	A/D Data Registers 10	ADDR10	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9036h	S12AD	A/D Data Registers 11	ADDR11	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9038h	S12AD	A/D Data Registers 12	ADDR12	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 903Ah	S12AD	A/D Data Registers 13	ADDR13	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 903Ch	S12AD	A/D Data Registers 14	ADDR14	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 903Eh	S12AD	A/D Data Registers 15	ADDR15	16	16	2 to 3PCLKB	2 ICLK		1294	
0008 9060h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9061h	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9066h	S12AD	A/D Sample and Hold Circuit Control Register	ADSHCR	16	16	2 to 3PCLKB	2 ICLK		1311	
0008 9071h	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9073h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9074h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9075h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9076h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9077h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9078h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 9079h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 to 3PCLKB	2 ICLK		1310	
0008 907Ah	S12AD	A/D Disconnecting Detection Control Register	ADDISCR	8	8	2 to 3PCLKB	2 ICLK		1312	
0008 A000h	SCI0	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	896	
0008 A001h	SCI0	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A002h	SCI0	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A003h	SCI0	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	

Table 5.1 List of I/O Registers (Address Order) (21 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 A004h	SCI0	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	905	
0008 A005h	SCI0	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A006h	SCI0	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A007h	SCI0	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A008h	SCI0	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A009h	SCI0	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A00Ah	SCI0	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A00Bh	SCI0	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A00Ch	SCI0	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A00Dh	SCI0	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A040h	SCI2	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A041h	SCI2	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A042h	SCI2	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A043h	SCI2	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A044h	SCI2	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A045h	SCI2	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A046h	SCI2	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A047h	SCI2	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A048h	SCI2	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A049h	SCI2	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A04Ah	SCI2	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A04Bh	SCI2	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A04Ch	SCI2	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A04Dh	SCI2	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A060h	SCI3	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A061h	SCI3	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A062h	SCI3	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A063h	SCI3	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A064h	SCI3	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A065h	SCI3	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A066h	SCI3	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A067h	SCI3	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A068h	SCI3	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A069h	SCI3	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A06Ah	SCI3	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A06Bh	SCI3	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A06Ch	SCI3	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	

Table 5.1 List of I/O Registers (Address Order) (22 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 A06Dh	SCI3	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	926	
0008 A080h	SCI4	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A081h	SCI4	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A082h	SCI4	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A083h	SCI4	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A084h	SCI4	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A085h	SCI4	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A086h	SCI4	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A087h	SCI4	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A088h	SCI4	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A089h	SCI4	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A08Ah	SCI4	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A08Bh	SCI4	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A08Ch	SCI4	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A08Dh	SCI4	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A0C9h	SCI6	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A0CAh	SCI6	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A0CBh	SCI6	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A0CCh	SCI6	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A0E0h	SCI7	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A0E1h	SCI7	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A0E2h	SCI7	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A0E3h	SCI7	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0E4h	SCI7	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A0E5h	SCI7	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A0E6h	SCI7	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A0E7h	SCI7	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	

Table 5.1 List of I/O Registers (Address Order) (23 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 A0E8h	SCI7	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	920	
0008 A0E9h	SCI7	I²C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A0EAh	SCI7	I²C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A0EBh	SCI7	I²C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A0ECh	SCI7	I²C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A0EDh	SCI7	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A100h	SCI8	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A101h	SCI8	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A102h	SCI8	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A103h	SCI8	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A104h	SCI8	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A105h	SCI8	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A106h	SCI8	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A107h	SCI8	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A108h	SCI8	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A109h	SCI8	I²C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A10Ah	SCI8	I²C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A10Bh	SCI8	I²C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A10Ch	SCI8	I²C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A10Dh	SCI8	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A120h	SCI9	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A121h	SCI9	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A122h	SCI9	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A123h	SCI9	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A124h	SCI9	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A125h	SCI9	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A126h	SCI9	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A127h	SCI9	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A128h	SCI9	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A129h	SCI9	I²C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A12Ah	SCI9	I²C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A12Bh	SCI9	I²C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A12Ch	SCI9	I²C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A12Dh	SCI9	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A140h	SCI10	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A141h	SCI10	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A142h	SCI10	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 A143h	SCI10	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A144h	SCI10	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A145h	SCI10	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A146h	SCI10	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A147h	SCI10	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A148h	SCI10	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A149h	SCI10	I²C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A14Ah	SCI10	I²C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A14Bh	SCI10	I²C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A14Ch	SCI10	I²C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A14Dh	SCI10	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 A160h	SCI11	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK		896	
0008 A161h	SCI11	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 A162h	SCI11	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	

Table 5.1 List of I/O Registers (Address Order) (24 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 A163h	SCI11	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	894	
0008 A164h	SCI11	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 A165h	SCI11	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 A166h	SCI11	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 A167h	SCI11	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 A168h	SCI11	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 A169h	SCI11	I²C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 A16Ah	SCI11	I²C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 A16Bh	SCI11	I²C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 A16Ch	SCI11	I²C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 A16Dh	SCI11	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 to 3PCLKB	2 ICLK	CAC	191	
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 to 3PCLKB	2 ICLK		192	
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 to 3PCLKB	2 ICLK		193	
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 to 3PCLKB	2 ICLK		194	
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 to 3PCLKB	2 ICLK		195	
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 to 3PCLKB	2 ICLK		196	
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 to 3PCLKB	2 ICLK		196	
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 to 3PCLKB	2 ICLK		196	
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 to 3PCLKB	2 ICLK	DOC	1344	
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 to 3PCLKB	2 ICLK		1345	
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 to 3PCLKB	2 ICLK		1345	
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2 to 3PCLKB	2 ICLK	ELC	439	
0008 B102h	ELC	Event Link Setting Register 1	ELSR1	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B103h	ELC	Event Link Setting Register 2	ELSR2	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Eh	ELC	Event Link Setting Register 29	ELSR29	8	8	2 to 3PCLKB	2 ICLK		440	
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2 to 3PCLKB	2 ICLK		443	
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2 to 3PCLKB	2 ICLK		444	
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2 to 3PCLKB	2 ICLK		444	
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2 to 3PCLKB	2 ICLK		445	
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2 to 3PCLKB	2 ICLK		446	
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2 to 3PCLKB	2 ICLK		446	
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2 to 3PCLKB	2 ICLK		447	

Table 5.1 List of I/O Registers (Address Order) (25 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2 to 3PCLKB	2 ICLK	ELC	447	
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2 to 3PCLKB	2 ICLK		448	
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2 to 3PCLKB	2 ICLK		448	
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2 to 3PCLKB	2 ICLK		449	
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2 to 3PCLKB	2 ICLK		449	
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2 to 3PCLKB	2 ICLK		449	
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2 to 3PCLKB	2 ICLK		449	
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2 to 3PCLKB	2 ICLK		450	
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2 to 3PCLKB	2 ICLK	SCIe, SCIf	896	
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2 to 3PCLKB	2 ICLK		911	
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2 to 3PCLKB	2 ICLK		900	
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2 to 3PCLKB	2 ICLK		905	
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2 to 3PCLKB	2 ICLK		894	
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2 to 3PCLKB	2 ICLK		909	
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2 to 3PCLKB	2 ICLK		918	
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2 to 3PCLKB	2 ICLK		920	
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2 to 3PCLKB	2 ICLK		921	
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2 to 3PCLKB	2 ICLK		922	
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2 to 3PCLKB	2 ICLK		923	
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2 to 3PCLKB	2 ICLK		925	
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2 to 3PCLKB	2 ICLK		926	
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2 to 3PCLKB	2 ICLK		927	
0008 B321h	SCI12	Control Register 0	CR0	8	8	2 to 3PCLKB	2 ICLK		928	
0008 B322h	SCI12	Control Register 1	CR1	8	8	2 to 3PCLKB	2 ICLK		928	
0008 B323h	SCI12	Control Register 2	CR2	8	8	2 to 3PCLKB	2 ICLK		929	
0008 B324h	SCI12	Control Register 3	CR3	8	8	2 to 3PCLKB	2 ICLK		930	
0008 B325h	SCI12	Port Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		930	
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2 to 3PCLKB	2 ICLK		931	
0008 B327h	SCI12	Status Register	STR	8	8	2 to 3PCLKB	2 ICLK		932	
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2 to 3PCLKB	2 ICLK		933	
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2 to 3PCLKB	2 ICLK		933	
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2 to 3PCLKB	2 ICLK		934	
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2 to 3PCLKB	2 ICLK		934	
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2 to 3PCLKB	2 ICLK		934	
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2 to 3PCLKB	2 ICLK		935	
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2 to 3PCLKB	2 ICLK		935	
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2 to 3PCLKB	2 ICLK		935	
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2 to 3PCLKB	2 ICLK		936	
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2 to 3PCLKB	2 ICLK		936	
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2 to 3PCLKB	2 ICLK		937	
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2 to 3PCLKB	2 ICLK		937	
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK	I/O Ports	467	
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C006h	PORT6	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C008h	PORT8	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	

Table 5.1 List of I/O Registers (Address Order) (26 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK	I/O Ports	467	
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C00Fh	PORTF	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C011h	PORTH	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C013h	PORTK	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C014h	PORTL	Port Direction Register	PDR	8	8	2 to 3PCLKB	2 ICLK		467	
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C026h	PORT6	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C028h	PORT8	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C02Fh	PORTF	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C031h	PORTH	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C033h	PORTK	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C034h	PORTL	Port Output Data Register	PODR	8	8	2 to 3PCLKB	2 ICLK		468	
0008 C040h	PORT0	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C045h	PORT5	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C046h	PORT6	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C047h	PORT7	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C048h	PORT8	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C049h	PORT9	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Ah	PORTA	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Ch	PORTC	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C04Fh	PORTF	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C051h	PORTH	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C052h	PORTJ	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C053h	PORTK	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C054h	PORTL	Port Input Data Register	PIDR	8	8	2 to 3PCLKB	2 ICLK		469	
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	

Table 5.1 List of I/O Registers (Address Order) (27 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK	I/O Ports	470	
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C066h	PORT6	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C068h	PORT8	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C06Fh	PORTF	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C071h	PORTH	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C073h	PORTK	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C074h	PORTL	Port Mode Register	PMR	8	8	2 to 3PCLKB	2 ICLK		470	
0008 C080h	PORT0	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C082h	PORT1	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C083h	PORT1	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C086h	PORT3	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C087h	PORT3	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C08Ah	PORT5	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C08Bh	PORT5	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C08Ch	PORT6	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C08Eh	PORT7	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C08Fh	PORT7	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C090h	PORT8	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C092h	PORT9	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C094h	PORTA	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C095h	PORTA	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C098h	PORTC	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C099h	PORTC	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C09Ch	PORTE	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C09Dh	PORTE	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C0A6h	PORTK	Open Drain Control Register 0	ODR0	8	8, 16	2 to 3PCLKB	2 ICLK		471	
0008 C0A7h	PORTK	Open Drain Control Register 1	ODR1	8	8, 16	2 to 3PCLKB	2 ICLK		472	
0008 C0C0h	PORT0	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C5h	PORT5	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C6h	PORT6	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C7h	PORT7	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C8h	PORT8	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0C9h	PORT9	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0CAh	PORTA	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	

Table 5.1 List of I/O Registers (Address Order) (28 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK	I/O Ports	473	
0008 C0CCh	PORTC	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0CEh	PORTE	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0CFh	PORTF	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0D1h	PORTH	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0D2h	PORTJ	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0D3h	PORTK	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0D4h	PORTL	Pull-Up Control Register	PCR	8	8	2 to 3PCLKB	2 ICLK		473	
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E3h	PORT3	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E5h	PORT5	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E6h	PORT6	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E8h	PORT8	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0ECh	PORTC	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0F1h	PORTH	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0F2h	PORTJ	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C0F3h	PORTK	Drive Capacity Control Register	DSCR	8	8	2 to 3PCLKB	2 ICLK		474	
0008 C100h	MPC	CS Output Enable Register	PFCSE	8	8	2 to 3PCLKB	2 ICLK	MPC	508	
0008 C104h	MPC	Address Output Enable Register 0	PFAOE0	8	8, 16	2 to 3PCLKB	2 ICLK		509	
0008 C105h	MPC	Address Output Enable Register 1	PFAOE1	8	8, 16	2 to 3PCLKB	2 ICLK		510	
0008 C106h	MPC	External Bus Control Register 0	PFBCR0	8	8, 16	2 to 3PCLKB	2 ICLK		511	
0008 C107h	MPC	External Bus Control Register 1	PFBCR1	8	8, 16	2 to 3PCLKB	2 ICLK		512	
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 to 3PCLKB	2 ICLK		488	
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2 to 3PCLKB	2 ICLK		489	
0008 C14Ah	MPC	P12 Pin Function Control Registers	P12PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C14Bh	MPC	P13 Pin Function Control Registers	P13PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C14Ch	MPC	P14 Pin Function Control Registers	P14PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C14Dh	MPC	P15 Pin Function Control Registers	P15PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C14Eh	MPC	P16 Pin Function Control Registers	P16PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C14Fh	MPC	P17 Pin Function Control Registers	P17PFS	8	8	2 to 3PCLKB	2 ICLK		490	
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2 to 3PCLKB	2 ICLK		491	
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2 to 3PCLKB	2 ICLK		491	

Table 5.1 List of I/O Registers (Address Order) (29 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2 to 3PCLKB	2 ICLK	MPC	491	
0008 C158h	MPC	P30 Pin Function Control Registers	P30PFS	8	8	2 to 3PCLKB	2 ICLK		492	
0008 C159h	MPC	P31 Pin Function Control Registers	P31PFS	8	8	2 to 3PCLKB	2 ICLK		492	
0008 C15Ah	MPC	P32 Pin Function Control Registers	P32PFS	8	8	2 to 3PCLKB	2 ICLK		492	
0008 C15Bh	MPC	P33 Pin Function Control Registers	P33PFS	8	8	2 to 3PCLKB	2 ICLK		492	
0008 C15Ch	MPC	P34 Pin Function Control Registers	P34PFS	8	8	2 to 3PCLKB	2 ICLK		492	
0008 C160h	MPC	P40 Pin Function Control Registers	P40PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C161h	MPC	P41 Pin Function Control Registers	P41PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C162h	MPC	P42 Pin Function Control Registers	P42PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C163h	MPC	P43 Pin Function Control Registers	P43PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C164h	MPC	P44 Pin Function Control Registers	P44PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C165h	MPC	P45 Pin Function Control Registers	P45PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C166h	MPC	P46 Pin Function Control Registers	P46PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C167h	MPC	P47 Pin Function Control Registers	P47PFS	8	8	2 to 3PCLKB	2 ICLK		493	
0008 C168h	MPC	P50 Pin Function Control Registers	P50PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C169h	MPC	P51 Pin Function Control Registers	P51PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C16Ah	MPC	P52 Pin Function Control Registers	P52PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C16Ch	MPC	P54 Pin Function Control Registers	P54PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C16Dh	MPC	P55 Pin Function Control Registers	P55PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C16Eh	MPC	P56 Pin Function Control Registers	P56PFS	8	8	2 to 3PCLKB	2 ICLK		494	
0008 C170h	MPC	P60 Pin Function Control Registers	P60PFS	8	8	2 to 3PCLKB	2 ICLK		495	
0008 C171h	MPC	P61 Pin Function Control Registers	P61PFS	8	8	2 to 3PCLKB	2 ICLK		495	
0008 C178h	MPC	P70 Pin Function Control Registers	P70PFS	8	8	2 to 3PCLKB	2 ICLK		496	
0008 C17Bh	MPC	P73 Pin Function Control Registers	P73PFS	8	8	2 to 3PCLKB	2 ICLK		496	Not available in 3-V packages.
0008 C17Ch	MPC	P74 Pin Function Control Registers	P74PFS	8	8	2 to 3PCLKB	2 ICLK		496	
0008 C17Dh	MPC	P75 Pin Function Control Registers	P75PFS	8	8	2 to 3PCLKB	2 ICLK		496	
0008 C17Eh	MPC	P76 Pin Function Control Registers	P76PFS	8	8	2 to 3PCLKB	2 ICLK		496	
0008 C17Fh	MPC	P77 Pin Function Control Registers	P77PFS	8	8	2 to 3PCLKB	2 ICLK		496	
0008 C180h	MPC	P80 Pin Function Control Registers	P80PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C181h	MPC	P81 Pin Function Control Registers	P81PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C182h	MPC	P82 Pin Function Control Registers	P82PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C183h	MPC	P83 Pin Function Control Registers	P83PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C186h	MPC	P86 Pin Function Control Registers	P86PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C187h	MPC	P87 Pin Function Control Registers	P87PFS	8	8	2 to 3PCLKB	2 ICLK		497	
0008 C188h	MPC	P90 Pin Function Control Registers	P90PFS	8	8	2 to 3PCLKB	2 ICLK		498	
0008 C189h	MPC	P91 Pin Function Control Registers	P91PFS	8	8	2 to 3PCLKB	2 ICLK		498	
0008 C18Ah	MPC	P92 Pin Function Control Registers	P92PFS	8	8	2 to 3PCLKB	2 ICLK		498	
0008 C18Bh	MPC	P93 Pin Function Control Registers	P93PFS	8	8	2 to 3PCLKB	2 ICLK		498	
0008 C190h	MPC	PA0 Pin Function Control Registers	PA0PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C191h	MPC	PA1 Pin Function Control Registers	PA1PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C192h	MPC	PA2 Pin Function Control Registers	PA2PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C193h	MPC	PA3 Pin Function Control Registers	PA3PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C194h	MPC	PA4 Pin Function Control Registers	PA4PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C195h	MPC	PA5 Pin Function Control Registers	PA5PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C196h	MPC	PA6 Pin Function Control Registers	PA6PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C197h	MPC	PA7 Pin Function Control Registers	PA7PFS	8	8	2 to 3PCLKB	2 ICLK		499	
0008 C198h	MPC	PB0 Pin Function Control Registers	PB0PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C199h	MPC	PB1 Pin Function Control Registers	PB1PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C19Ah	MPC	PB2 Pin Function Control Registers	PB2PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C19Bh	MPC	PB3 Pin Function Control Registers	PB3PFS	8	8	2 to 3PCLKB	2 ICLK		500	

Table 5.1 List of I/O Registers (Address Order) (30 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C19Ch	MPC	PB4 Pin Function Control Registers	PB4PFS	8	8	2 to 3PCLKB	2 ICLK	MPC	500	
0008 C19Dh	MPC	PB5 Pin Function Control Registers	PB5PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C19Eh	MPC	PB6 Pin Function Control Registers	PB6PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C19Fh	MPC	PB7 Pin Function Control Registers	PB7PFS	8	8	2 to 3PCLKB	2 ICLK		500	
0008 C1A0h	MPC	PC0 Pin Function Control Registers	PC0PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A1h	MPC	PC1 Pin Function Control Registers	PC1PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A2h	MPC	PC2 Pin Function Control Registers	PC2PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A3h	MPC	PC3 Pin Function Control Registers	PC3PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A4h	MPC	PC4 Pin Function Control Registers	PC4PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A5h	MPC	PC5 Pin Function Control Registers	PC5PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A6h	MPC	PC6 Pin Function Control Registers	PC6PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A7h	MPC	PC7 Pin Function Control Registers	PC7PFS	8	8	2 to 3PCLKB	2 ICLK		501	
0008 C1A8h	MPC	PD0 Pin Function Control Registers	PD0PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1A9h	MPC	PD1 Pin Function Control Registers	PD1PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1AAh	MPC	PD2 Pin Function Control Registers	PD2PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1ABh	MPC	PD3 Pin Function Control Registers	PD3PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1ACh	MPC	PD4 Pin Function Control Registers	PD4PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1ADh	MPC	PD5 Pin Function Control Registers	PD5PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1AEh	MPC	PD6 Pin Function Control Registers	PD6PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1AFh	MPC	PD7 Pin Function Control Registers	PD7PFS	8	8	2 to 3PCLKB	2 ICLK		502	
0008 C1B0h	MPC	PE0 Pin Function Control Registers	PE0PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B1h	MPC	PE1 Pin Function Control Registers	PE1PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B2h	MPC	PE2 Pin Function Control Registers	PE2PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B3h	MPC	PE3 Pin Function Control Registers	PE3PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B4h	MPC	PE4 Pin Function Control Registers	PE4PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B5h	MPC	PE5 Pin Function Control Registers	PE5PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B6h	MPC	PE6 Pin Function Control Registers	PE6PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1B7h	MPC	PE7 Pin Function Control Registers	PE7PFS	8	8	2 to 3PCLKB	2 ICLK		503	
0008 C1BDh	MPC	PF5 Pin Function Control Registers	PF5PFS	8	8	2 to 3PCLKB	2 ICLK		504	
0008 C1C8h	MPC	PH0 Pin Function Control Registers	PH0PFS	8	8	2 to 3PCLKB	2 ICLK		504	
0008 C1C9h	MPC	PH1 Pin Function Control Registers	PH1PFS	8	8	2 to 3PCLKB	2 ICLK		504	
0008 C1CAh	MPC	PH2 Pin Function Control Registers	PH2PFS	8	8	2 to 3PCLKB	2 ICLK		504	
0008 C1CBh	MPC	PH3 Pin Function Control Registers	PH3PFS	8	8	2 to 3PCLKB	2 ICLK		504	
0008 C1D1h	MPC	PJ1 Pin Function Control Registers	PJ1PFS	8	8	2 to 3PCLKB	2 ICLK		505	
0008 C1D3h	MPC	PJ3 Pin Function Control Registers	PJ3PFS	8	8	2 to 3PCLKB	2 ICLK		505	
0008 C1DAh	MPC	PK2 Pin Function Control Registers	PK2PFS	8	8	2 to 3PCLKB	2 ICLK		506	
0008 C1DBh	MPC	PK3 Pin Function Control Registers	PK3PFS	8	8	2 to 3PCLKB	2 ICLK		506	
0008 C1DCh	MPC	PK4 Pin Function Control Registers	PK4PFS	8	8	2 to 3PCLKB	2 ICLK		506	
0008 C1DDh	MPC	PK5 Pin Function Control Registers	PK5PFS	8	8	2 to 3PCLKB	2 ICLK		506	
0008 C1E5h	MPC	PL5 Pin Function Control Register	PL5PFS	8	8	2 to 3PCLKB	2 ICLK		507	Not available in 5-V packages.
0008 C280h	SYSTEM	Deep Standby Control Register	DPSBYCR	8	8	4 to 5PCLKB	2 to 3 ICLK	Low Power Consumption	216	
0008 C282h	SYSTEM	Deep Standby Interrupt Enable Register 0	DPSIER0	8	8	4 to 5PCLKB	2 to 3 ICLK		218	
0008 C284h	SYSTEM	Deep Standby Interrupt Enable Register 2	DPSIER2	8	8	4 to 5PCLKB	2 to 3 ICLK		219	
0008 C286h	SYSTEM	Deep Standby Interrupt Flag Register 0	DPSIFR0	8	8	4 to 5PCLKB	2 to 3 ICLK		220	
0008 C288h	SYSTEM	Deep Standby Interrupt Flag Register 2	DPSIFR2	8	8	4 to 5PCLKB	2 to 3 ICLK		221	
0008 C28Ah	SYSTEM	Deep Standby Interrupt Edge Register 0	DPSIEGR0	8	8	4 to 5PCLKB	2 to 3 ICLK		223	
0008 C28Ch	SYSTEM	Deep Standby Interrupt Edge Register 2	DPSIEGR2	8	8	4 to 5PCLKB	2 to 3 ICLK		224	
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 to 5PCLKB	2 to 3 ICLK	Resets	132	
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 to 5PCLKB	2 to 3 ICLK		134	

Table 5.1 List of I/O Registers (Address Order) (31 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 to 5PCLKB	2 to 3 ICLK	Clock Generation Circuit	180	
0008 C296h	FLASH	Flash Write Erase Protection Register	FWEPROR	8	8	4 to 5PCLKB	2 to 3 ICLK	Flash Memory	1356	
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 to 5PCLKB	2 to 3 ICLK	LVDA	155	
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4 to 5PCLKB	2 to 3 ICLK		156	
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 to 5PCLKB	2 to 3 ICLK		157	
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4 to 5PCLKB	2 to 3 ICLK		158	
0008 C2A0h to 0008 C2BFh	SYSTEM	Deep Standby Backup Register 0 to 31	DPSBKR0 to 31	8	8	4 to 5PCLKB	2 to 3 ICLK	Low Power Consumption	224	
000A 0A00h	CEC	CEC Local Address Setting Register	CADR	16	16	1 to 2PCLK	1 ICLK	CEC	1065	Not available in 5-V packages.
000A 0A02h	CEC	CEC Control Register 1	CECCTL1	8	8	1 to 2PCLK	1 ICLK		1070	Not available in 5-V packages.
000A 0A04h	CEC	CEC Transmission Start Bit Width Setting Register	STATB	16	16	1 to 2PCLK	1 ICLK		1076	Not available in 5-V packages.
000A 0A06h	CEC	CEC Transmission Start Bit Low Width Setting Register	STATL	16	16	1 to 2PCLK	1 ICLK		1075	Not available in 5-V packages.
000A 0A08h	CEC	CEC Transmission Logical 0 Low Width Setting Register	LGC0L	16	16	1 to 2PCLK	1 ICLK		1076	Not available in 5-V packages.
000A 0A0Ah	CEC	CEC Transmission Logical 1 Low Width Setting Register	LGC1L	16	16	1 to 2PCLK	1 ICLK		1076	Not available in 5-V packages.
000A 0A0Ch	CEC	CEC Transmission Data Bit Width Setting Register	DATB	16	16	1 to 2PCLK	1 ICLK		1077	Not available in 5-V packages.
000A 0A0Eh	CEC	CEC Reception Data Sampling Time Setting Register	NOMT	16	16	1 to 2PCLK	1 ICLK		1081	Not available in 5-V packages.
000A 0A10h	CEC	CEC Reception Start Bit Minimum Low Width Setting Register	STATLL	16	16	1 to 2PCLK	1 ICLK		1077	Not available in 5-V packages.
000A 0A12h	CEC	CEC Reception Start Bit Maximum Low Width Setting Register	STATLH	16	16	1 to 2PCLK	1 ICLK		1078	Not available in 5-V packages.
000A 0A14h	CEC	CEC Reception Start Bit Minimum Bit Width Setting Register	STATBL	16	16	1 to 2PCLK	1 ICLK		1078	Not available in 5-V packages.
000A 0A16h	CEC	CEC Reception Start Bit Maximum Bit Width Setting Register	STATBH	16	16	1 to 2PCLK	1 ICLK		1079	Not available in 5-V packages.
000A 0A18h	CEC	CEC Reception Logical 0 Minimum Low Width Setting Register	LGC0LL	16	16	1 to 2PCLK	1 ICLK		1079	Not available in 5-V packages.
000A 0A1Ah	CEC	CEC Reception Logical 0 Maximum Low Width Setting Register	LGC0LH	16	16	1 to 2PCLK	1 ICLK		1080	Not available in 5-V packages.
000A 0A1Ch	CEC	CEC Reception Logical 1 Minimum Low Width Setting Register	LGC1LL	16	16	1 to 2PCLK	1 ICLK		1080	Not available in 5-V packages.
000A 0A1Eh	CEC	CEC Reception Logical 1 Maximum Low Width Setting Register	LGC1LH	16	16	1 to 2PCLK	1 ICLK		1080	Not available in 5-V packages.
000A 0A20h	CEC	CEC Reception Data Bit Minimum Bit Width Setting Register	DATBL	16	16	1 to 2PCLK	1 ICLK		1081	Not available in 5-V packages.
000A 0A22h	CEC	CEC Reception Data Bit Maximum Bit Width Setting Register	DATBH	16	16	1 to 2PCLK	1 ICLK		1081	Not available in 5-V packages.
000A 0A24h	CEC	CEC Data Bit Reference Width Setting Register	NOMP	16	16	1 to 2PCLK	1 ICLK		1082	Not available in 5-V packages.

Table 5.1 List of I/O Registers (Address Order) (32 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
000A 0A28h	CEC	CEC Extension Mode Register	CECEXMD	8	8	1 to 2PCLK	1 ICLK	CEC	1083	Not available in 5-V packages.
000A 0A2Ah	CEC	CEC Extension Monitor Register	CECEXMON	8	8	1 to 2PCLK	1 ICLK		1084	Not available in 5-V packages.
000A 0A30h	CEC	CEC Transmission Buffer Register	CTXD	8	8	1 to 2PCLK	1 ICLK		1066	Not available in 5-V packages.
000A 0A31h	CEC	CEC Reception Buffer Register	CRXD	8	8	1 to 2PCLK	1 ICLK		1066	Not available in 5-V packages.
000A 0A32h	CEC	CEC Communication Error Status Register	CECES	8	8	1 to 2PCLK	1 ICLK		1073	Not available in 5-V packages.
000A 0A33h	CEC	CEC Communication Status Register	CECS	8	8	1 to 2PCLK	1 ICLK		1071	Not available in 5-V packages.
000A 0A34h	CEC	CEC Communication Error Flag Clear Trigger Register	CECFC	8	8	1 to 2PCLK	1 ICLK		1075	Not available in 5-V packages.
000A 0A35h	CEC	CEC Control Register 0	CECCTL0	8	8	1 to 2PCLK	1 ICLK		1067	Not available in 5-V packages.
000A 0B00h	RCR0	Function Select Register 0	CON0	8	8	1 to 2PCLK	1 ICLK	RCR	1024	Not available in 5-V packages.
000A 0B01h	RCR0	Function Select Register 1	CON1	8	8	1 to 2PCLK	1 ICLK		1025	Not available in 5-V packages.
000A 0B02h	RCR0	Status Register	STS	8	8	1 to 2PCLK	1 ICLK		1026	Not available in 5-V packages.
000A 0B03h	RCR0	Interrupt Control Register	INT	8	8	1 to 2PCLK	1 ICLK		1029	Not available in 5-V packages.
000A 0B04h	RCR0	Compare Control Register	CPC	8	8	1 to 2PCLK	1 ICLK		1029	Not available in 5-V packages.
000A 0B05h	RCR0	Compare Value Setting Register	CPD	8	8	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B06h	RCR0	Header Pattern Setting Register (Min)	HDPMIN	16	16	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B08h	RCR0	Header Pattern Setting Register (Max)	HDPMAX	16	16	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B0Ah	RCR0	Data 0 Pattern Setting Register (Min)	D0PMIN	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B0Bh	RCR0	Data 0 Pattern Setting Register (Max)	D0PMAX	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B0Ch	RCR0	Data 1 Pattern Setting Register (Min)	D1PMIN	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B0Dh	RCR0	Data 1 Pattern Setting Register (Max)	D1PMAX	8	8	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B0Eh	RCR0	Special Data Pattern Setting Register (Min)	SDPMIN	16	16	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B10h	RCR0	Special Data Pattern Setting Register (Max)	SDPMAX	16	16	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B12h	RCR0	Pattern End Setting Register	PE	16	16	1 to 2PCLK	1 ICLK		1033	Not available in 5-V packages.
000A 0B15h	RCR0	Receive Bit Count Register	RBIT	8	8	1 to 2PCLK	1 ICLK		1033	Not available in 5-V packages.

Table 5.1 List of I/O Registers (Address Order) (33 / 34)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
000A 0B16h	RCR0	Receive Data 0 Register	DAT0	8	8	1 to 2PCLK	1 ICLK	RCR	1034	Not available in 5-V packages.
000A 0B17h	RCR0	Receive Data 1 Register	DAT1	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B18h	RCR0	Receive Data 2 Register	DAT2	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B19h	RCR0	Receive Data 3 Register	DAT3	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B1Ah	RCR0	Receive Data 4 Register	DAT4	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B1Bh	RCR0	Receive Data 5 Register	DAT5	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B1Ch	RCR0	Receive Data 6 Register	DAT6	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B1Dh	RCR0	Receive Data 7 Register	DAT7	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B1Eh	RCR0	Measurement Result Register	TIM	16	16	1 to 2PCLK	1 ICLK		1035	Not available in 5-V packages.
000A 0B80h	RCR1	Function Select Register 0	CON0	8	8	1 to 2PCLK	1 ICLK		1024	Not available in 5-V packages.
000A 0B81h	RCR1	Function Select Register 1	CON1	8	8	1 to 2PCLK	1 ICLK		1025	Not available in 5-V packages.
000A 0B82h	RCR1	Status Register	STS	8	8	1 to 2PCLK	1 ICLK		1026	Not available in 5-V packages.
000A 0B83h	RCR1	Interrupt Control Register	INT	8	8	1 to 2PCLK	1 ICLK		1029	Not available in 5-V packages.
000A 0B84h	RCR1	Compare Control Register	CPC	8	8	1 to 2PCLK	1 ICLK		1029	Not available in 5-V packages.
000A 0B85h	RCR1	Compare Value Setting Register	CPD	8	8	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B86h	RCR1	Header Pattern Setting Register (Min)	HDPMIN	16	16	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B88h	RCR1	Header Pattern Setting Register (Max)	HDPMAX	16	16	1 to 2PCLK	1 ICLK		1030	Not available in 5-V packages.
000A 0B8Ah	RCR1	Data 0 Pattern Setting Register (Min)	D0PMIN	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B8Bh	RCR1	Data 0 Pattern Setting Register (Max)	D0PMAX	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B8Ch	RCR1	Data 1 Pattern Setting Register (Min)	D1PMIN	8	8	1 to 2PCLK	1 ICLK		1031	Not available in 5-V packages.
000A 0B8Dh	RCR1	Data 1 Pattern Setting Register (Max)	D1PMAX	8	8	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B8Eh	RCR1	Special Data Pattern Setting Register (Min)	SDPMIN	16	16	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B90h	RCR1	Special Data Pattern Setting Register (Max)	SDPMAX	16	16	1 to 2PCLK	1 ICLK		1032	Not available in 5-V packages.
000A 0B92h	RCR1	Pattern End Setting Register	PE	16	16	1 to 2PCLK	1 ICLK		1033	Not available in 5-V packages.

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page	Remarks
						ICLK ≥ PCLK	ICLK < PCLK			
000A 0B95h	RCR1	Receive Bit Count Register	RBIT	8	8	1 to 2PCLK	1 ICLK	RCR	1033	Not available in 5-V packages.
000A 0B96h	RCR1	Receive Data 0 Register	DAT0	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B97h	RCR1	Receive Data 1 Register	DAT1	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B98h	RCR1	Receive Data 2 Register	DAT2	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B99h	RCR1	Receive Data 3 Register	DAT3	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B9Ah	RCR1	Receive Data 4 Register	DAT4	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B9Bh	RCR1	Receive Data 5 Register	DAT5	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B9Ch	RCR1	Receive Data 6 Register	DAT6	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B9Dh	RCR1	Receive Data 7 Register	DAT7	8	8	1 to 2PCLK	1 ICLK		1034	Not available in 5-V packages.
000A 0B9Eh	RCR1	Measurement Result Register	TIM	16	16	1 to 2PCLK	1 ICLK		1035	Not available in 5-V packages.
000A 0C00h	SYSTEM	Main Clock Supply Control Register	MOSCR	8	8	1 to 2PCLK	1 ICLK	Clock Generation Circuit	180	Not available in 5-V packages.
000A 0C02h	SYSTEM	Main Clock Noise Filter Control Register	MONFCR	8	8	1 to 2PCLK	1 ICLK		181	Not available in 5-V packages.
007F C402h	FLASH	Flash Mode Register	FMODR	8	8	2 to 3 FCLK	2 to 3 ICLK	Flash Memory	1357	
007F C410h	FLASH	Flash Access Status Register	FASTAT	8	8	2 to 3 FCLK	2 to 3 ICLK		1358	
007F C411h	FLASH	Flash Access Error Interrupt Enable Register	FAEINT	8	8	2 to 3 FCLK	2 to 3 ICLK		1361	
007F C412h	FLASH	Flash Ready Interrupt Enable Register	FRDYIE	8	8	2 to 3 FCLK	2 to 3 ICLK		1362	
007F C440h	FLASH	E2 DataFlash Read Enable Register 0	DFLRE0	16	16	2 to 3 FCLK	2 to 3 ICLK		1363	
007F C442h	FLASH	E2 DataFlash Read Enable Register 1	DFLRE1	16	16	2 to 3 FCLK	2 to 3 ICLK		1364	
007F C450h	FLASH	E2 DataFlash P/E Enable Register 0	DFLWE0	16	16	2 to 3 FCLK	2 to 3 ICLK		1365	
007F C452h	FLASH	E2 DataFlash P/E Enable Register 1	DFLWE1	16	16	2 to 3 FCLK	2 to 3 ICLK		1366	
007F FFB0h	FLASH	Flash Status Register 0	FSTATR0	8	8	2 to 3 FCLK	2 to 3 ICLK		1367	
007F FFB1h	FLASH	Flash Status Register 1	FSTATR1	8	8	2 to 3 FCLK	2 to 3 ICLK		1369	
007F FFB2h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 to 3 FCLK	2 to 3 ICLK		1370	
007F FFB4h	FLASH	Flash Protection Register	FPROTR	16	16	2 to 3 FCLK	2 to 3 ICLK		1372	
007F FFB6h	FLASH	Flash Reset Register	FRESETR	16	16	2 to 3 FCLK	2 to 3 ICLK		1373	
007F FFBAh	FLASH	FCU Command Register	FCMDR	16	16	2 to 3 FCLK	2 to 3 ICLK		1374	
007F FFC8h	FLASH	FCU Processing Switching Register	FCPSR	16	16	2 to 3 FCLK	2 to 3 ICLK		1375	
007F FFCAh	FLASH	E2 DataFlash Blank Check Control Register	DFLBCCNT	16	16	2 to 3 FCLK	2 to 3 ICLK		1375	
007F FFCCh	FLASH	Flash P/E Status Register	FPESTAT	16	16	2 to 3 FCLK	2 to 3 ICLK		1376	
007F FFCEh	FLASH	E2 DataFlash Blank Check Status Register	DFLBCSTAT	16	16	2 to 3 FCLK	2 to 3 ICLK		1376	
007F FFE8h	FLASH	Peripheral Clock Notification Register	PCKAR	16	16	2 to 3 FCLK	2 to 3 ICLK		1377	

Note: This table lists the I/O registers in both 5-V and 3-V package specifications. The I/O registers of each product correspond to the functions listed in Table 1.2. For details, see Table 1.2, Comparison of Functions of Different RX634 Group Products.
Note: The CEC, RCR0, and RCR1 are not available in 5-V packages.
Note 1. Odd addresses cannot be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 26.4 lists register allocation for 16-bit access.
Note 2. Odd addresses cannot be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMOCNTL register.

6. Resets

6.1 Overview

The nine types of reset are as follows: RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, deep software standby 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
Deep software standby reset	Deep software standby mode is canceled by an interrupt.
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 details on the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDA) and section 41, 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	Deep Software Standby 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)	○	○	—	—	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	○	—	—	—	—	○	—
Registers related to the 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)	○	○	○	○	○	—	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	○	—	—	○	—
Voltage-monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	○	—	—	—
Registers related to the voltage monitor function 2 LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	○	○	○	○	○	○	—	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	○	—	○	—
Deep software standby reset detect flag (RSTSR0.DPSRSTF)	○	○	○	○	○	○	○	—	—
Software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	○	○	—
Register related to main clock oscillator (MOFCR)	○	○	○	○	○	○	○	—	○
Pin state	○	○	○	○	○	○	○	—	○
Registers related to the low power-consumption function (DPSBYCR, DPSIER0, 2, DPSIFR0, 2, DPSIEGR0, 2)	○	○	○	○	○	○	○	—	○
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○	○	○

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

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
	DPSRSTF	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0*1	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
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSRSTF	Deep Software Standby Reset Flag	0: Deep software standby mode cancelation not requested by an interrupt. 1: Deep software standby mode cancelation requested by an interrupt.	R/(W) *2

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 VCC voltage has fallen below Vdet0.
[Setting condition]

- When Vdet0-level VCC voltage is detected.

[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 VCC voltage has fallen below Vdet1.
[Setting condition]

- When Vdet1-level VCC voltage is detected.

[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 VCC voltage has fallen below Vdet2.
[Setting condition]

- When Vdet2-level VCC voltage is detected.

[Clearing conditions]

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

DPSRSTF Flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that deep software standby mode has been canceled by an interrupt and that an internal reset (deep software standby reset) occurred.
[Setting condition]

- When deep software standby mode is canceled by an interrupt.
  For details, see section 11, Low Power Consumption.

[Clearing conditions]

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

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 RES# pin reset.
[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]

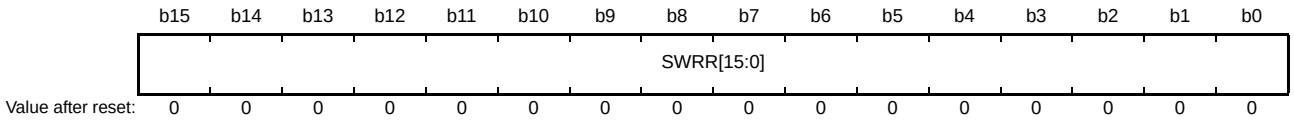
- When a software reset occurs.

[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



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

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

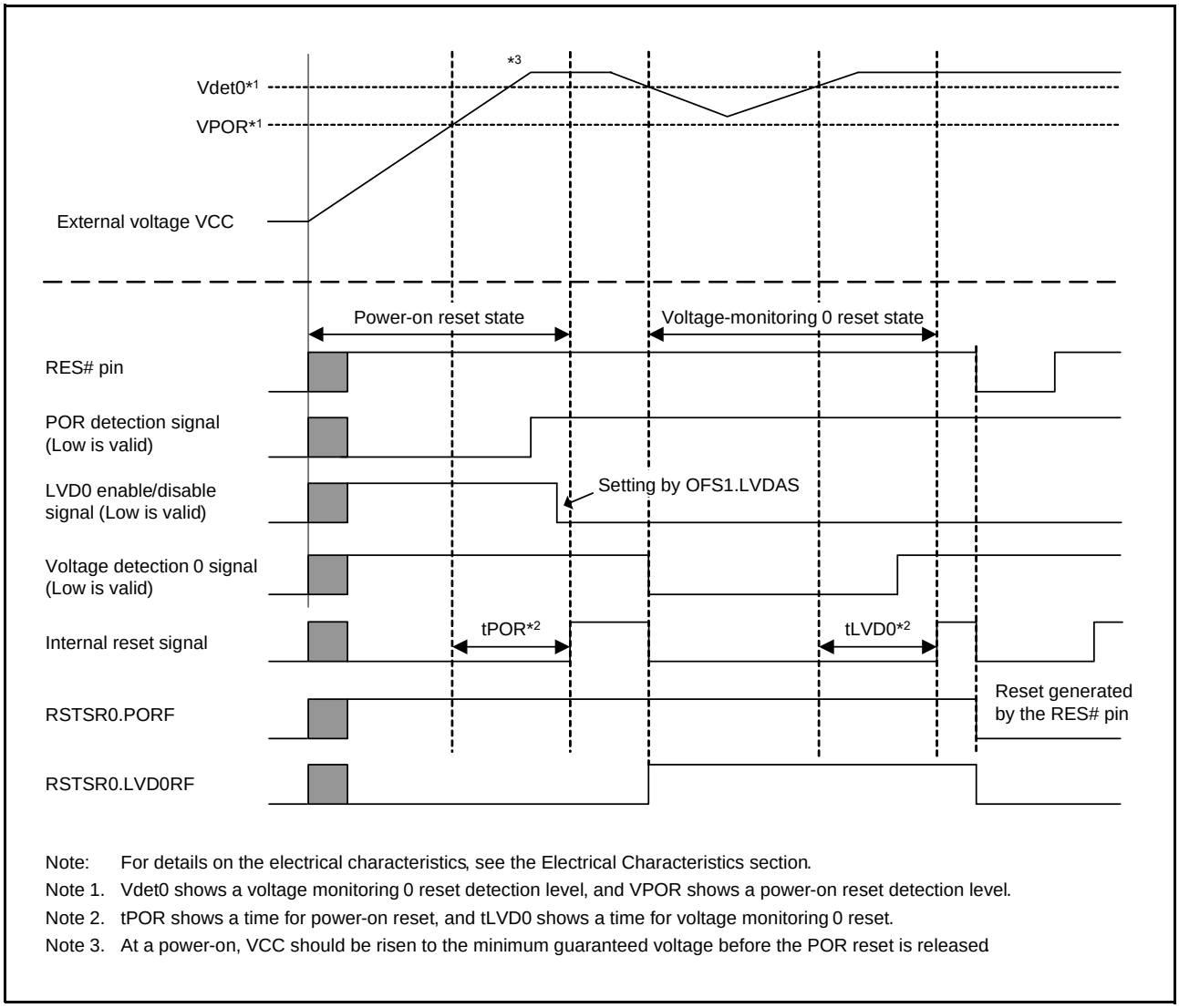
6.3 Operation

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 this MCU enters a reset state.
In order to unfailingly reset this MCU, 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 41, 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. 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 a stabilization period of the external power supply and this MCU circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by a RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 level selection (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.
Release from the voltage monitoring 0 reset state occurs when VCC rises above Vdet0 and the LVD0 reset time (tLVD0) elapses.
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 the Electrical Characteristics section.
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.

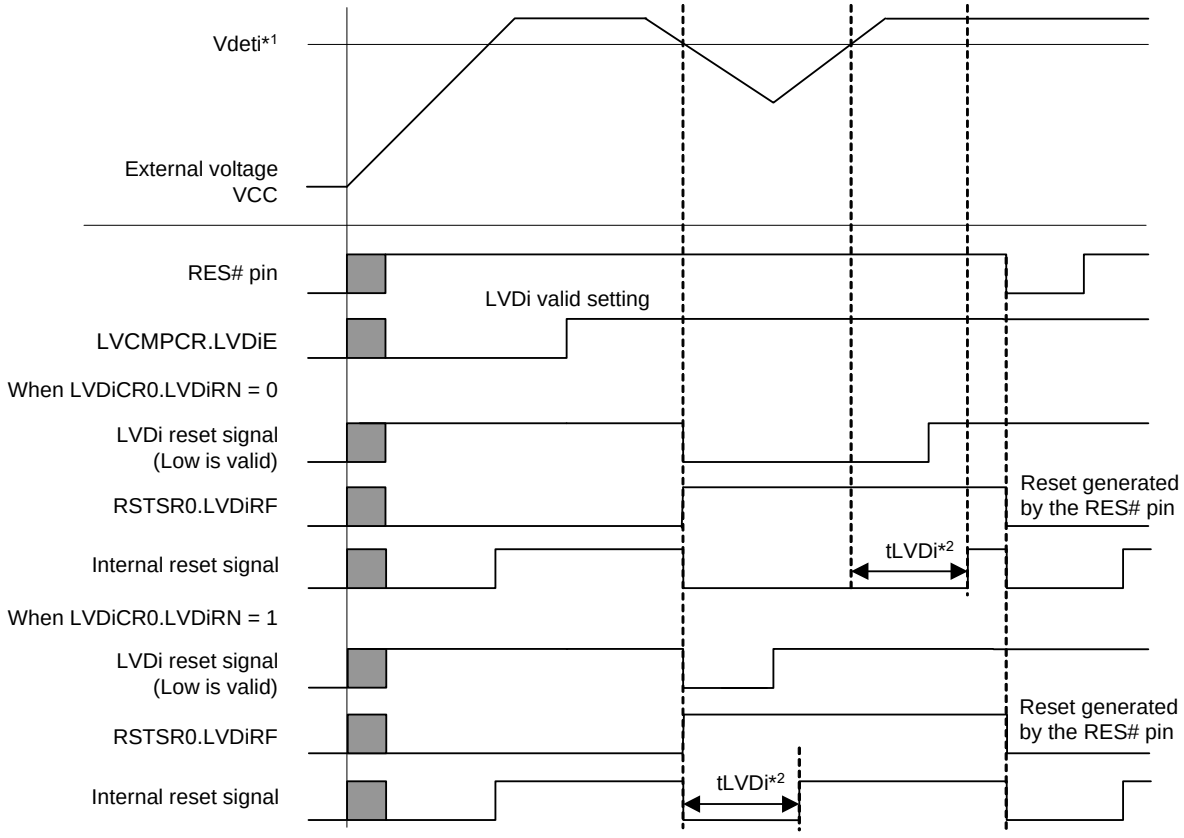
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of 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 LVD1 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 LVD1 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 LDV2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection select register (LDVLVLR).

Table 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 the Electrical Characteristics section.
Note 1. Vdeti shows a detection level of voltage monitoring 1 reset and voltage monitoring 2 reset.
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 Deep Software Standby Reset

This is an internal reset generated when deep software standby mode is canceled by an interrupt.
When a deep software standby mode cancelation source is generated, a deep software standby reset is generated. The deep software standby reset is canceled after tDSBY (return time after deep software standby mode cancelation) has elapsed. At the same time, deep software standby mode is also canceled.
When tDSBYWT (wait time after deep software standby mode cancelation) has elapsed after deep software standby mode has been canceled, the internal reset is canceled and the CPU starts the reset exception handling.
For details of the deep software standby reset, see section 11, Low Power Consumption.

6.3.5 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 29, Independent Watchdog Timer (IWDTa).

6.3.6 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the watchdog timer reset from the watchdog timer can be selected by settings in the WDT reset control register (WDTRCR) 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 28, Watchdog Timer (WDTA).

6.3.7 Software Reset

The software reset is an internal reset generated by the software reset circuit.
When A501h is written to the software reset register (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.

6.3.8 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.

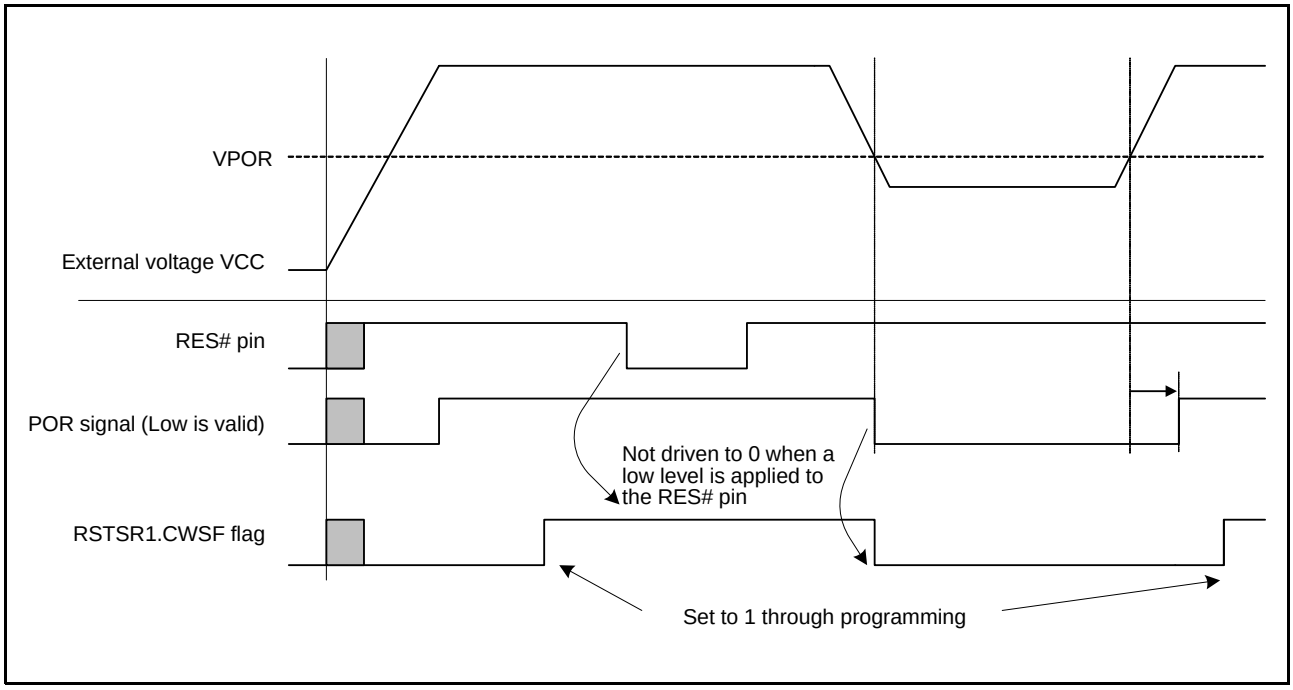


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.9 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.

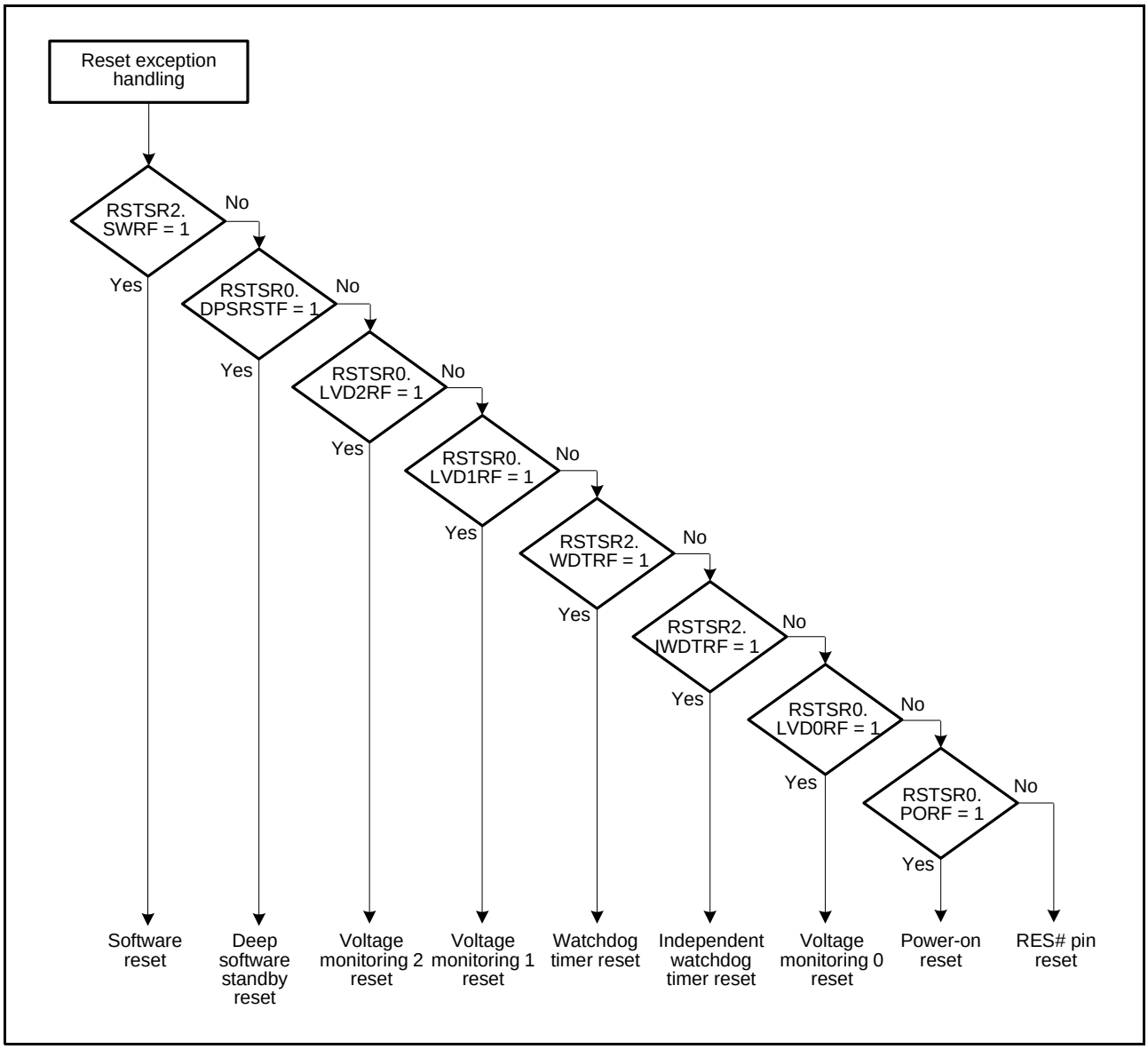


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory

7.1 Overview

Option-setting memory refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the ROM.
Figure 7.1 shows the option-setting memory area.

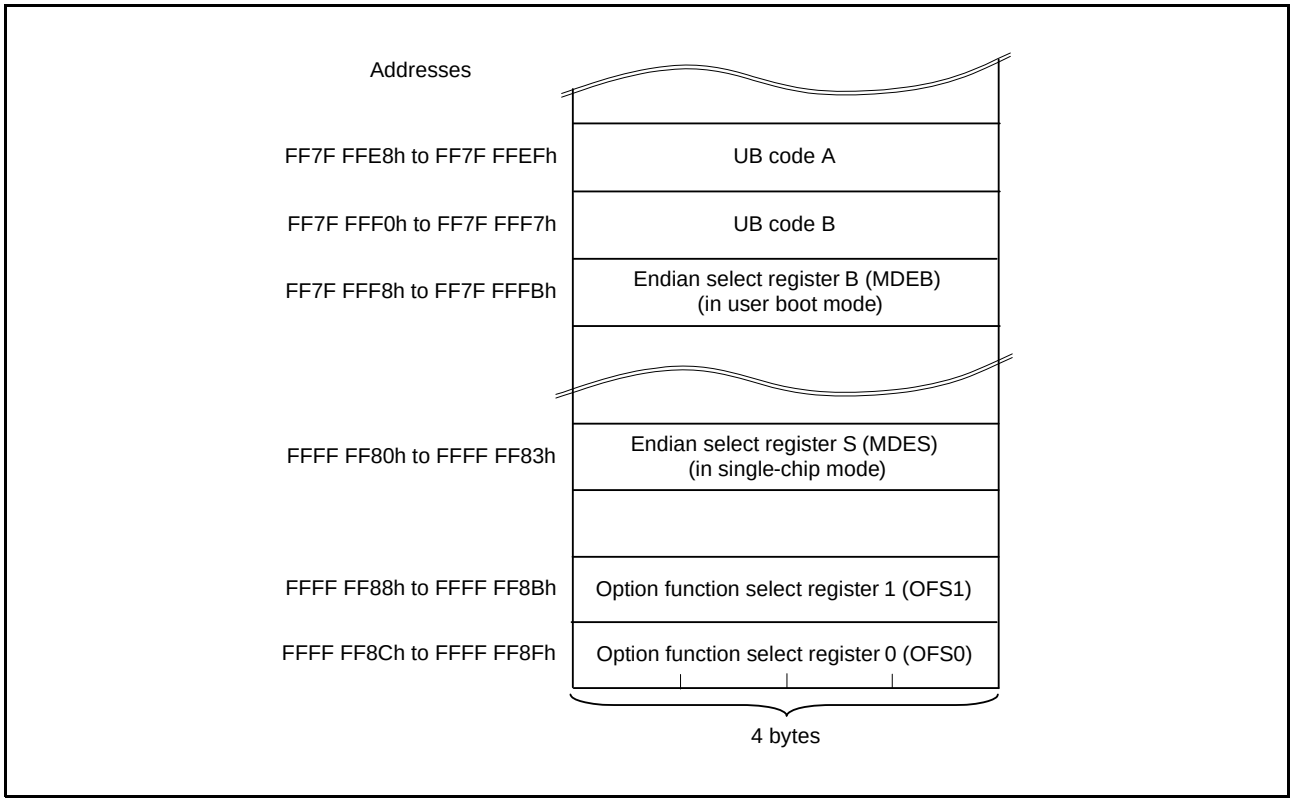


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): FFFF FF8Ch

	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 to 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 Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: × 1 (Cycle period: 131 ms) 0 0 1 0: × 1/16 (Cycle period: 2.10 s) 0 0 1 1: × 1/32 (Cycle period: 4.19 s) 0 1 0 0: × 1/64 (Cycle period: 8.39 s) 1 1 1 1: × 1/128 (Cycle period: 16.8 s) 0 1 0 1: × 1/256 (Cycle period: 33.6 s) 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 is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns to 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, deep software standby, or all-module clock stop mode	R
b16, b15	—	Reserved	When reading, these bits return to 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

Bit	Symbol	Bit Name	Description	R/W
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
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/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 is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.
When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ineffective in user boot mode, and the value becomes FFFF FFFFh.

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 LOCO 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, see section 29, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT 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 LOCO 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 LOCO clock cycles for the IWDT.
For details, see section 29, 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 29, 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 29, 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. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 29, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects to stop counting when entering sleep, software standby, deep software standby, or all-module clock stop mode.
For details, see section 29, 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, see section 28, 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, see section 28, 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 28, 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 28, 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. Either a watchdog timer reset or a non-maskable interrupt request is selectable.

For details, refer to section 28, Watchdog Timer (WDTA).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): FFFF FF88h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDAS	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	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
b31 to b3	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.

When erasing the block including the OFS1 register, the setting in the OFS1 register is ineffective, and the OFS1 register value becomes FFFF FFFFh.

The setting in the OFS1 register is ineffective in user boot mode, and the value becomes FFFF FFFFh.

LVDAS Bit (Voltage Detection 0 Circuit Start)

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

7.2.3 Endian Select Register B (MDEB), Endian Select Register S (MDES)

Address(es): FF7F FFF8h: MDEB (in user boot mode)
FFFF FF80h: MDES (in single-chip mode)

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	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
b31 to b3	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

The MDEn (n = B, S) register selects the endian for the CPU. In user boot mode, the endian select register B (MDEB) at address FF7F FFF8h is used to select the endian. In single-chip mode, the endian select register S (MDES) at address FFFF FF80h is used.
MDEn is allocated in the ROM. Set the register at the same time as writing the program. After writing to the register once, do not write to it again.
When erasing the block including the MDEn register, the MDEn register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.
The endian is determined by the value at address FF7F FFF8h in the user boot area when operating in user boot mode, and by the value at address FFFF FF80h in the user area when operating in single-chip mode.

7.3 UB Codes

UB codes A and B are required if user boot mode is to be employed. The MCU will start up in user boot mode on release from the reset state if the four conditions below are satisfied.

- UB code A is 5573 6572h and 426F 6F74h.
- UB code B is FFFF FF07h and 0008 C04Ch.
- The low level is being input on the MD pin.
- The high level is being input on the PC7 pin.

7.3.1 UB Code A

UB code A consists of two 32-bit words. Set UB code A to 5573 6572h and 426F 6F74h. Do not set other values for the code.
Figure 7.2 shows the structure of UB code A in memory. Set UB code A in 32-bit units.

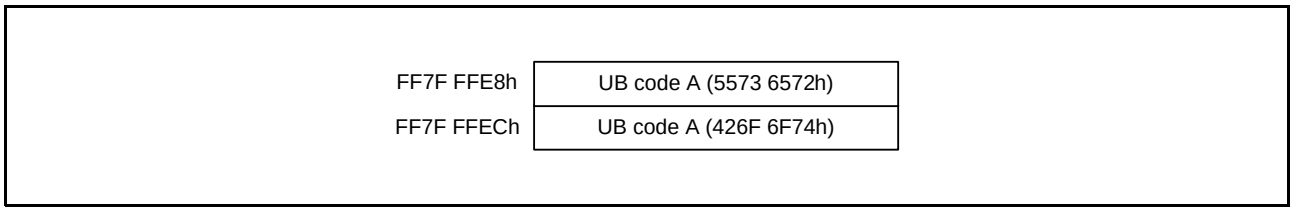


Figure 7.2 UB Code A Structure

7.3.2 UB Code B

UB code B consists of two words, i.e. 32 bits. Set UB code B to FFFF FF07h and 0008 C04Ch. Do not set other values for the code.
Figure 7.3 shows the structure of UB code B in memory. Set UB code B in 32-bit units.

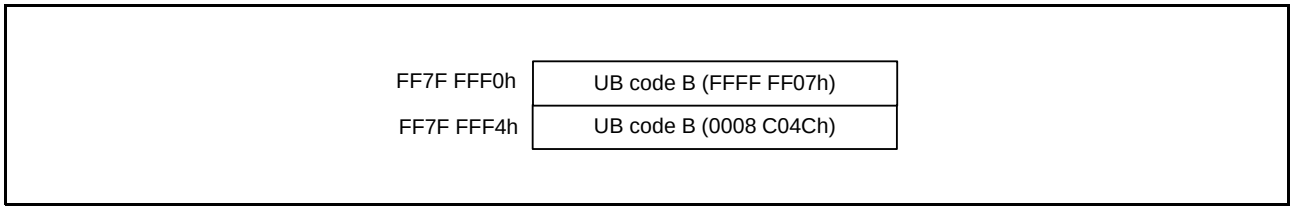


Figure 7.3 UB Code B Structure

7.4 Usage Note

7.4.1 Setting Example of Option-Setting Memory

Since the option-setting memory is allocated in the ROM, values cannot be written by executing instructions. Write appropriate values when writing the program. An example of the settings is shown below.

- To set ffff fff8h in the OFS0 register
  .org 0ffff ff8ch
  .lword 0ffffff8h

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

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

In voltage detection 0, whether to enable or disable the reset of voltage monitoring 0 can be selected after the reset using the option function select register 1 (OFS1).
In voltage detection 1 and voltage detection 2, the detection voltage is set using the voltage detection level select register (LVDLVLR).
Reset of voltage monitoring 0, reset/interrupt of voltage monitoring 1, and reset/interrupt of 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
	Detected event	Voltage drops past Vdet0	Voltage rises or drops past Vdet1	Voltage rises or drops past Vdet2
	Detection voltage	One level fixed	Specify voltage using LVDLVLR.LVD1LVL[3:0] bits	Specify voltage 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	Non-maskable interrupt
			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: 1, 2, 4, 8)	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)

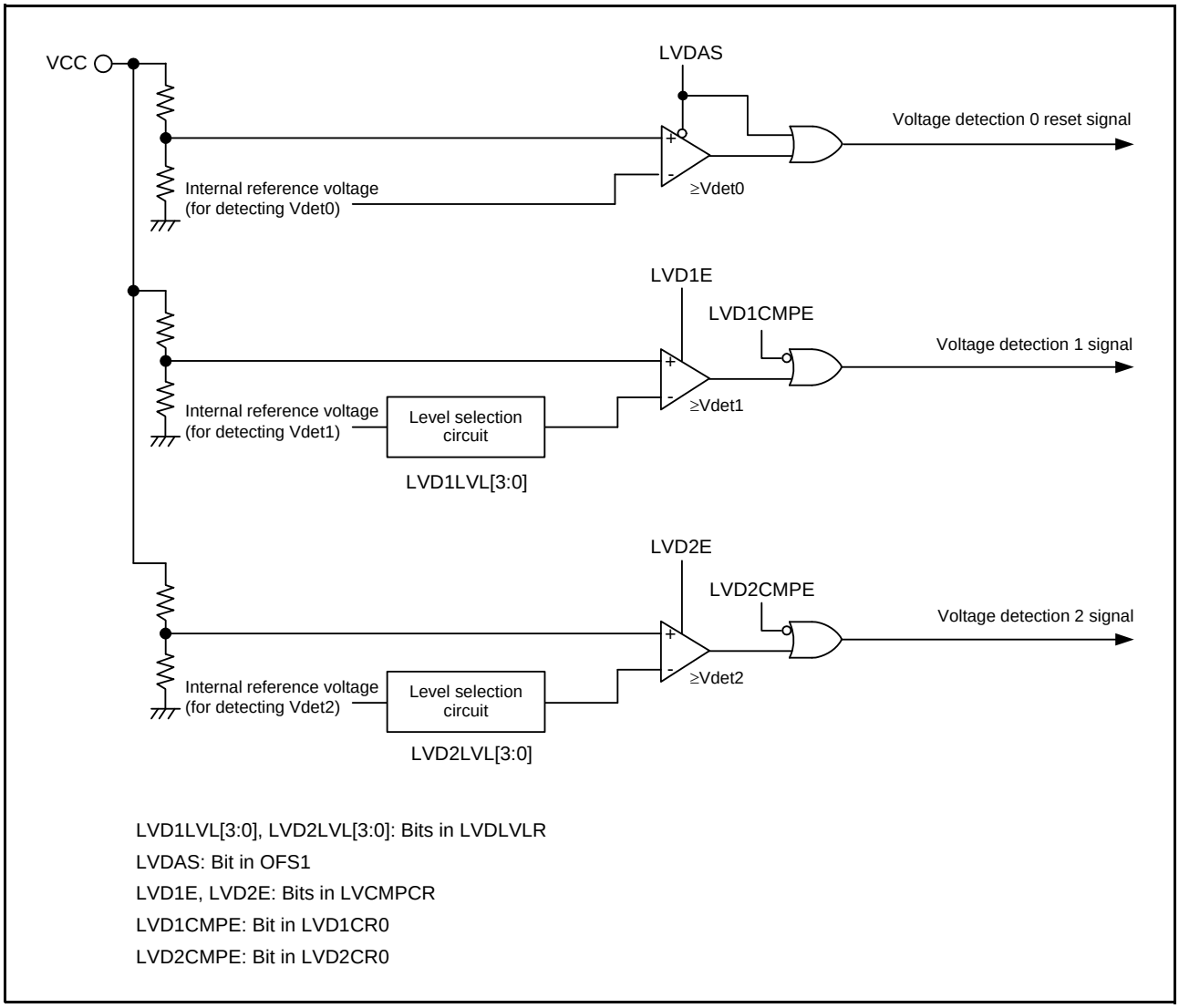


Figure 8.1 Block Diagram of Voltage Detection Circuit

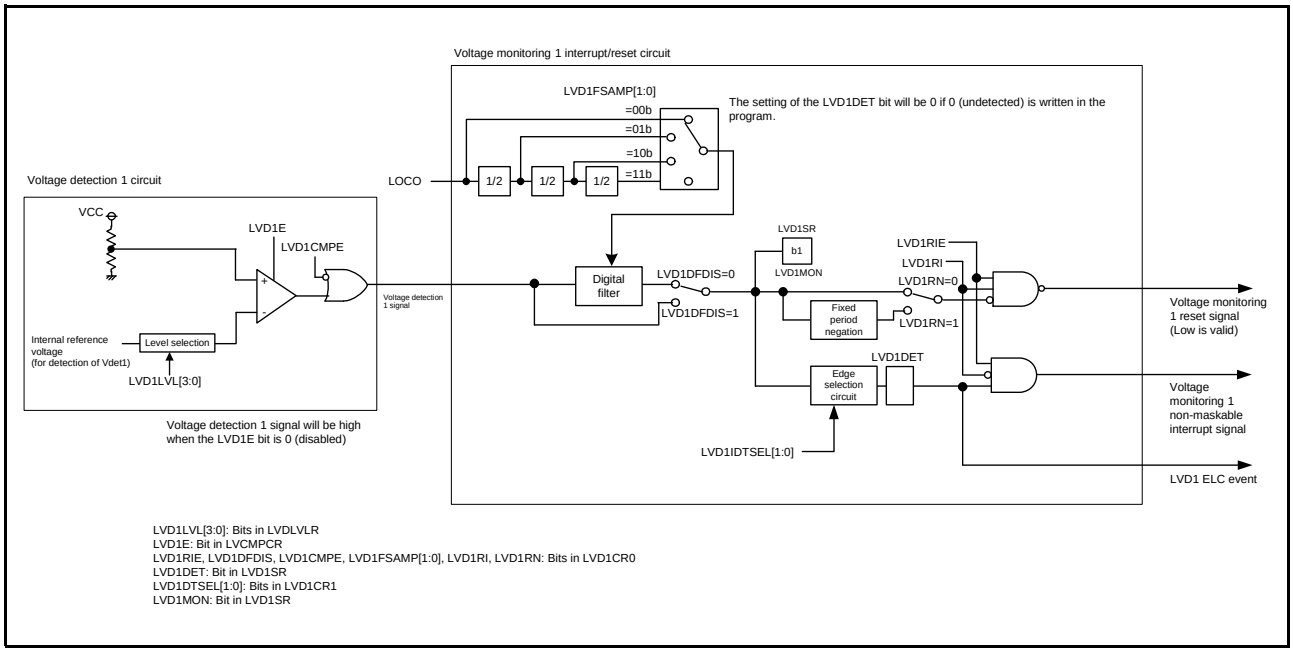


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

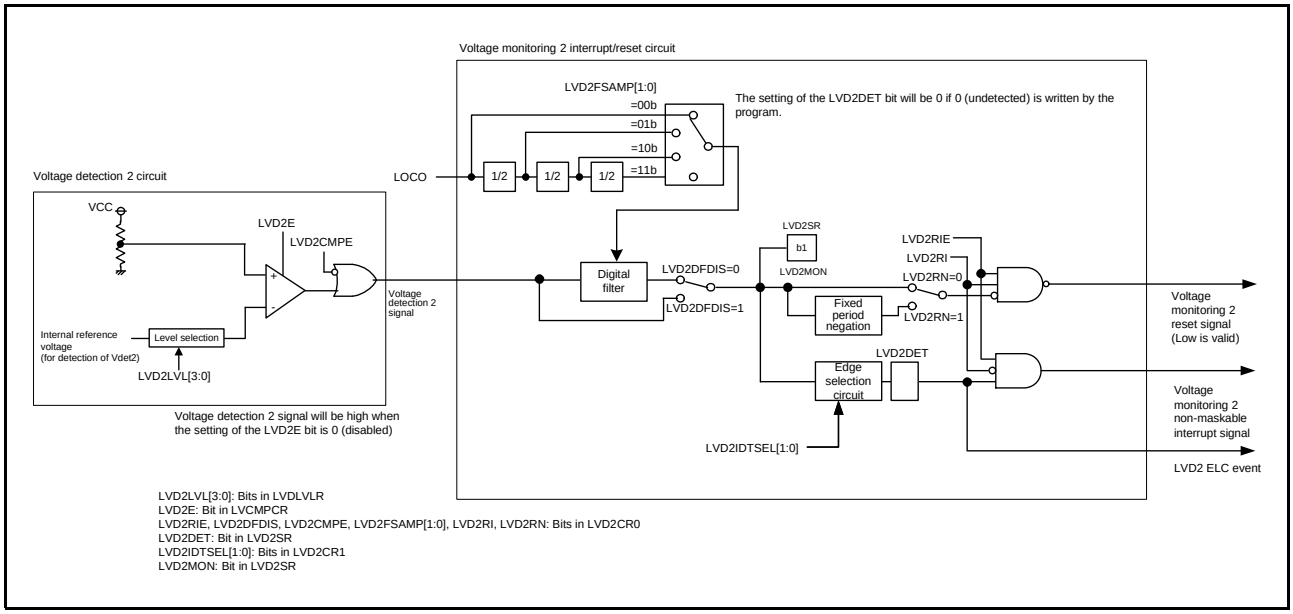


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

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
	—	—	—	—	—	—	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/ ELC Event 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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

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: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 peripheral module 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).
When LVD1CR0 and LVD1CR1 have been modified, the LVD1DET flag may become 1.
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
	—	—	—	—	—	—	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/ ELC Event 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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

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: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 peripheral module 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).
When LVD2CR0 and LVD2CR1 have been modified, the LVD2DET flag may become 1.
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE 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

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

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:	1	x	x	x	1	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	[3-V product] b3 b2 b1 b0 1 0 0 0: 2.90 V 1 0 0 1: 2.85 V 1 0 1 0: 2.88 V Do not set otherwise. [5-V product] b3 b2 b1 b0 1 0 0 0: 4.77 V 1 0 0 1: 4.23 V 1 0 1 0: 4.50 V Do not set otherwise.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	[3-V product] b3 b2 b1 b0 1 0 0 0: 2.90 V 1 0 0 1: 2.85 V 1 0 1 0: 2.88 V Do not set otherwise. [5-V product] b3 b2 b1 b0 1 0 0 0: 4.77 V 1 0 0 1: 4.23 V 1 0 1 0: 4.50 V Do not set otherwise.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

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.

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: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 LOCO frequency	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt enabled when Vdet1 is crossed 1: Voltage monitoring 1 reset enabled when the voltage falls 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

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (enabling the voltage detection 1 circuit) and the LVD1CMPE bit is set to 1 (enabling output of the results of comparison by voltage monitoring 1).
During programming or erasure of flash memory, please do not generate neither a voltage monitoring 1 interrupt nor voltage monitoring 1 reset.

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 or deep 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 disabled).

LVD1RI Bit (Voltage Monitoring 1 Circuit Mode Select)

When the LVD1RI bit is 1 (voltage monitoring 1 reset enabled) or when the LVD2CR0.LVD2RI bit is 1 (voltage monitoring 2 reset enabled), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter deep software standby mode, set the LVD1RI bit to 0 (voltage monitoring 1 interrupt enabled) and the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt 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 or deep 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: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 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 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

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

The LVD2RIE bit is enabled when the LVCMPCR.LVD2E bit is set to 1 (enabling the voltage detection 2) and the LVD2CMPE bit is set to 1 (enabling output of the results of comparison by voltage monitoring 2).
During programming or erasure of flash memory, please do not generate neither a voltage monitoring 2 interrupt nor voltage monitoring 2 reset.

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 or deep 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).

LVD2RI Bit (Voltage Monitoring 2 Circuit Mode Select)

When the LVD2RI it is 1 (voltage monitoring 2 reset enabled) or when the LVD1CR0.LVD1RI bit is 1 (voltage monitoring 1 reset enabled), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter to deep software standby mode, set the LVD2RI bit to 0 (voltage monitoring 2 interrupt enabled) and the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt 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 or deep 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	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
2	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.	Set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter).
3	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitoring 1).	
4	Wait for at least one cycle of the LOCO.	—
5	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).	—
6	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	— (waiting is not required)
7	Set the LVCMPCR.LVD1E bit to 1 (enabling the voltage detection 1 circuit).	

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.

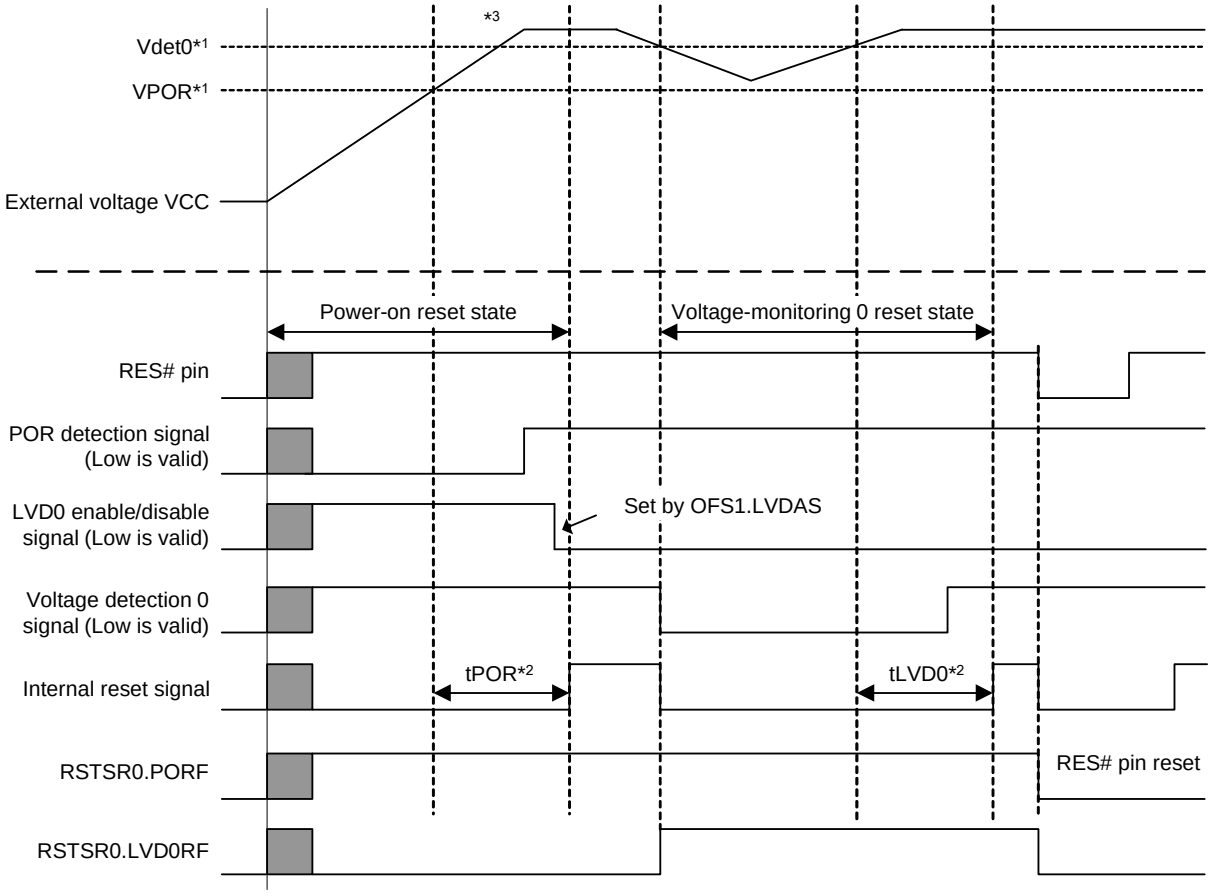
Table 8.3 Procedures for Setting up Monitoring against Vdet2

Step	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.	
2	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.	Set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter).
3	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitoring 2).	
4	Wait for at least one cycle of the LOCO.	—
5	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).	—
6	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	— (waiting is not required)
7	Set the LVCMPCR.LVD2E bit to 1 (enabling the voltage detection 2 circuit).	

8.4 Reset from Voltage Monitoring 0

When using the reset from voltage monitoring 0, clear the voltage detection 0 circuit start bit (OFS1.LVDAS) to 0 (enabling the voltage monitoring 0 reset after a reset).

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



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. VPOR indicates the detection level for a power-on reset and Vdet0 indicates the detection level for a voltage monitoring 0 reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 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 Monitoring 1

Table 8.4 lists the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

Furthermore, set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitoring 1 circuit in software standby or deep software standby mode.

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

	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1*1	Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.		Set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter).	
3 *1, *2	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitoring 1 interrupt).	• Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitoring 1 reset). • Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitoring 1 interrupt).	• Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitoring 1 reset). • Select the type of the reset negation by setting the LVD1CR0.LVD1RN bit.
4	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—
5	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitoring 1).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).		— (waiting is not required)	
9	Clear the LVD1SR.LVD1DET flag to 0.	—	Clear the LVD1SR.LVD1DET flag to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (enabling the voltage monitoring 1 interrupt or reset).			
11*1	Set the LVCMPCR.LVD1E bit to 1 (enabling the voltage detection 1 circuit).			

Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or of the LVD1CR1.LVD1IDTSEL[1:0] bits, or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 11.

Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset to Stop Voltage Monitoring

Step	Setting Bits Related to the Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output, and Voltage Monitoring 1 Reset
1*1	Clear the LVCMPCR.LVD1E bit to 0 (disabling the voltage detection 1 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD1CR0.LVD1RIE bit to 0 (disabling the voltage monitoring 1 interrupt or reset).
4	Clear the LVD1CR0.LVD1CMPE bit to 0 (disabling output of the results of comparison by voltage monitoring 1).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD1E, LVD1CR0.LVD1CMPE, and LVD1RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or the LVD1CR1.LVD1IDTSEL[1:0] bits, or when restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

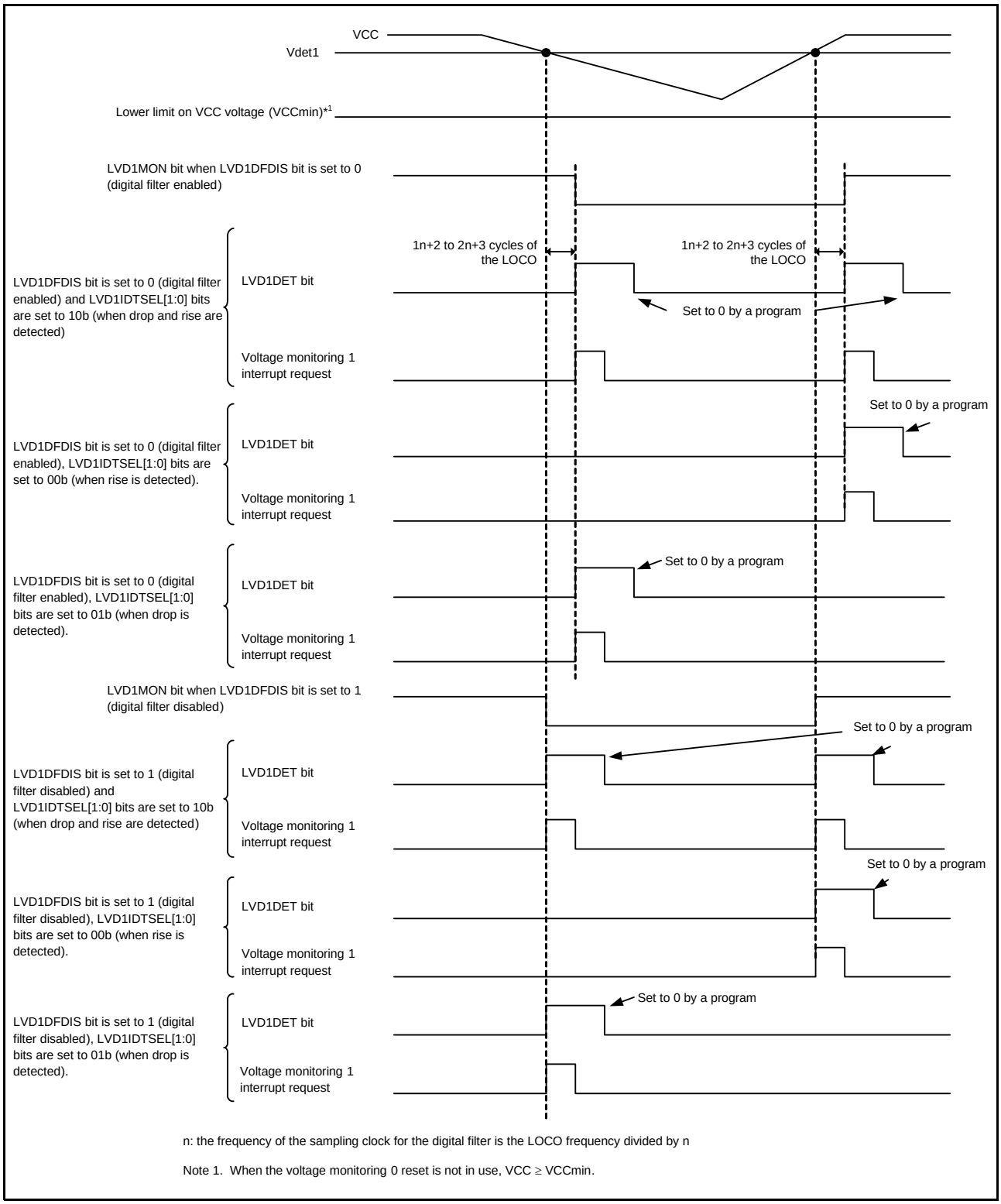


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitoring 2

Table 8.6 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset so that voltage monitoring operates.Table 8.7 shows the procedure for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitoring 2 interrupt. For the operation of the voltage monitoring 2 reset, see Figure 6.2 in section 6, Resets.

Furthermore, set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitoring 2 circuit in software standby or deep software standby mode.

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

	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
Step	Voltage Monitoring 2 Interrupt, Voltage Monitoring 2 ELC Event Output	Voltage Monitoring 2 Reset	Voltage Monitoring 2 Interrupt, Voltage Monitoring 2 ELC Event Output	Voltage Monitoring 2 Reset
1*1	Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.		Set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter).	
3 *1,*2	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitoring 2 interrupt).	• Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitoring 2 reset). • Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitoring 2 interrupt).	• Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitoring 2 reset). • Select the type of the reset negation by setting the LVD2CR0.LVD2RN bit.
4	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—
5	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitoring 2).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least 2n + 3 cycles of the LOCO (where n = 1, 2, 4, 8, and the sampling clock for the digital filter is the LOCO frequency-divided by n).		— (waiting is not required)	
9	Clear the LVD2SR.LVD2DET flag to 0.	—	Clear the LVD2SR.LVD2DET flag to 0.	—
10	Set the LVD2CR0.LVD2RIE bit to 1 (enabling the voltage monitoring 2 interrupt or reset).			
11*1	Set the LVCMPCR.LVD2E bit to 1 (enabling the voltage detection 2 circuit).			

Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or of the LVD2CR1.LVD2IDTSEL[1:0] bits, or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 11.

Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

Table 8.7 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset to Stop Voltage Monitoring

Step	Setting Bits Related to the Voltage Monitoring 2 Interrupt, Voltage Monitoring 2 ELC Event Output, and Voltage Monitoring 2 Reset
1*1	Clear the LVCMPCR.LVD2E bit to 0 (disabling the voltage detection 2 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD2CR0.LVD2RIE bit to 0 (disabling the voltage monitoring 2 interrupt or reset).
4	Clear the LVD2CR0.LVD2CMPE bit to 0 (disabling output of the results of comparison by voltage monitoring 2).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD2E, LVD2CR0.LVD2CMPE, and LVD2RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or the LVD2CR1.LVD2IDTSEL[1:0] bits, or when restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

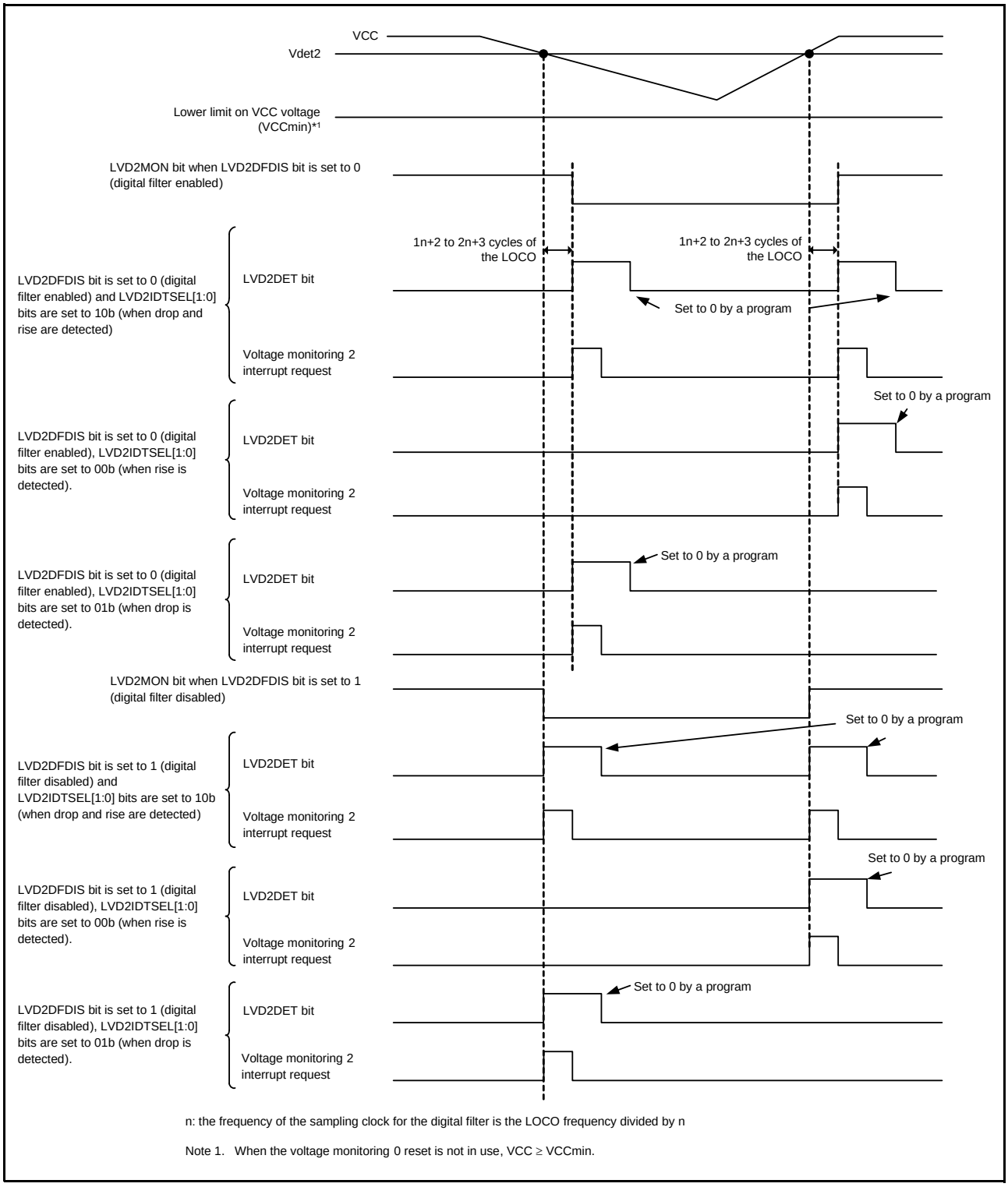


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

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

(1) Vdet1 passage detection event output
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 1 circuit comparison result output are enabled.

(2) Vdet2 passage detection event output
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 before 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 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt request signal 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.
It is possible to output voltage monitoring 1 and 2 interrupts in software standby and deep software standby modes. The event signals for the ELC, however, are output as follows:

- When the events of passing 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 the 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.
- If events of passing Vdet1/Vdet2 are detected in deep software standby mode, no event signals are generated for the ELC.

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, DMAC, DTC, ROM, and RAM. • Generates the peripheral module clock (PCLKB) to be supplied to the peripheral module. • Generates the S12AD clock (PCLKD) to be supplied to the S12AD. • Generates the FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the external bus clock (BCLK) to be supplied to the external bus. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT. • Generates the CEC clock (CECCLK) to be supplied to the CEC. • Generates the RCR clock (RCRCLK) to be supplied to the RCR. • Generates the JTAG-dedicated clock (JTAGTCK) to be supplied to the JTAG.
Operating frequency	• ICLK: 54 MHz (max) • PCLKB: 32 MHz (max) • PCLKD: 54 MHz (max) • FCLK: 4 MHz to 32 MHz (for programming and erasing the ROM and E2 DataFlash) 32 MHz (for reading from the E2 DataFlash) • BCLK: 54 MHz (max) • BCLK pin output: 27 MHz (max) • CACCLK: Same as the clock from respective oscillators. • IWDTCLK: 125 kHz • CECCLK: Same as the clock from respective oscillators. • RCRCLK: Same as the clock from respective oscillators. • JTAGTCK: 10 MHz (max)
Main clock oscillator	• Resonator frequency: 8 MHz to 20 MHz • External clock input frequency: 20 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal resonator • Connection pin: EXTAL, XTAL • Oscillation stop detection function: When an oscillation stop of the main clock is detected, the system clock source is switched to LOCO, and MTU pins are driven to high-impedance.
PLL circuit	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 MHz to 20 MHz • Frequency multiplication ratio: Selectable from 8, 10, 12, 16, 20, 24, 25, and 50 • VCO oscillation frequency: 104 MHz to 200 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 125 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 125 kHz
External clock input (TCK) for JTAG	Input clock frequency: 10 MHz (max)
Control of output on the BCLK pin	• BCLK clock output or high-level output is selectable • BCLK or BCLK/2 is selectable

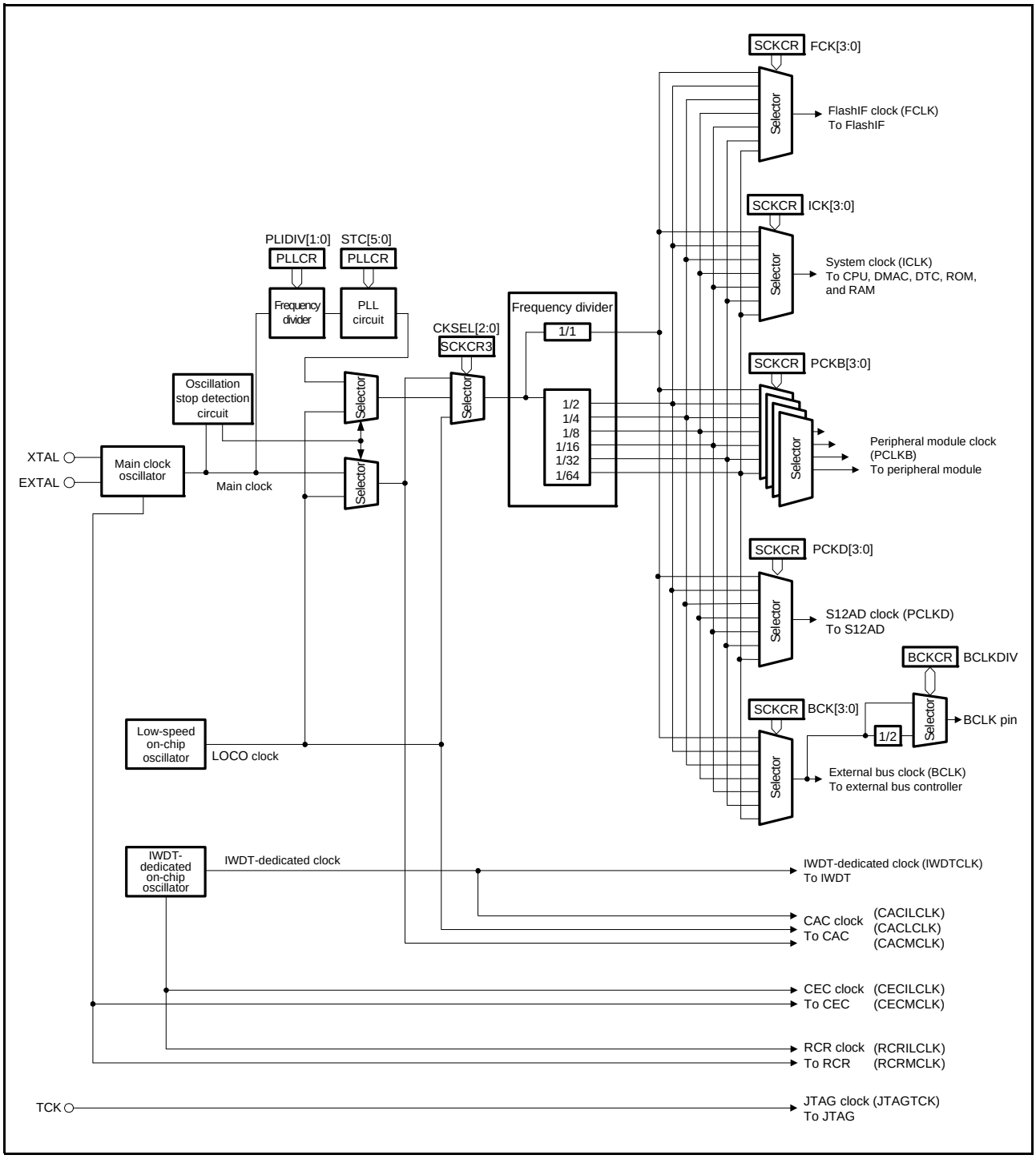


Figure 9.1 Block Diagram of Clock Generation Circuit

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	
TCK	Input	This pin is used to input the clock for the JTAG.
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).

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]				PSTOP1	—	—	—	BCK[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
	—	—	—	—	PCKB[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]	S12AD Clock (PCLKD) Select*1, *3, *7	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
b7 to b4	—	Reserved	Set these bits to 0010b.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select*1 to *3, *7	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	—	Reserved	Set these bits to 0010b.	R/W
b19 to b16	BCK[3:0]	External Bus Clock (BCLK) Select*3 to *5, *7	b19 b16 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
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	PSTOP1	BCLK Pin Output Control*5	0: BCLK pin output is enabled. 1: BCLK pin output is disabled. (Fixed high)	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select*2 to *4, *6, *7	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

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	FCK[3:0]	FlashIF Clock (FCLK) Select*3, *6,*7	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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Make a setting such that the frequencies satisfy the relation PCLKB:PCLKD = N:1, where N is an integer.
Note 2. Make a setting such that the frequencies satisfy the relation ICLK:PCLKB = N:1 or 1:N, where N is an integer.
Note 3. The setting for division by one is prohibited if the PLL is selected.
Note 4. Do not make a setting such that the ICLK runs at a lower frequency than the external bus clock.
Note 5. When operation of the external bus clock is selected, the P53 I/O port pin function is not available because it is multiplexed on the same pin as the BCLK pin function.
Note 6. Make a setting such that the frequencies satisfy the relation ICLK:FCLK = N:1 or 1:N, where N is an integer.
Note 7. The setting for division by one or two is prohibited if the SCKCR3.CKSEL[2:0] bits are set to 010b (the main clock oscillator is selected)

SCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in operating power control register (OPCCR.OPCMTSF) is 1 (during transition)
- ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode or E2 DataFlash P/E mode) (i = 0 to 3 or D)
- Time period from WAIT instruction issuance for a transition to sleep mode, to return from sleep mode to normal operating mode

9.2.2 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 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SCKCR3 should not be modified in the following cases:

- The operating power control mode transition status flag in operating power control register (OPCCR.OPCMTSF) is 1 (during transition)
- ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode or E2 DataFlash P/E mode) (i = 0 to 3 or D)
- Time period from WAIT instruction issuance 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 clock (PCLKB), S12AD clock (PCLKD), FlashIF clock (FCLK), and external bus clock (BCLK) from low-speed on-chip oscillator (LOCO), the main clock oscillator, and the PLL circuit.
Transitions to clock sources which are not in operation are prohibited.

9.2.3 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]						—	—	—	—	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	0	0	1	1	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: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	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 0 0 1 1 1: ×8 0 0 1 0 0 1: ×10 0 0 1 0 1 1: ×12 0 0 1 1 1 1: ×16 0 1 0 0 1 1: ×20 0 1 0 1 1 1: ×24 0 1 1 0 0 0: ×25 1 1 0 0 0 1: ×50 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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 the PLL input signal is within the range of 4 to 20 MHz.

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 VCO oscillation frequency for the PLL (104 MHz to 200 MHz).

9.2.4 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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Only set this register after setting the PLL wait control register according to the procedure in section 11, Low Power Consumption.

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.
After the setting of the PLLEN bit has been changed so that the PLL operates, only start using the PLL clock after the PLL clock oscillation stabilization waiting time (tPLLWT1 or tPLLWT2) has elapsed.
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.

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

Writing 1 to the PLLEN bit (PLL is stopped) is prohibited while the PLL clock is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
Writing 0 to the PLLEN bit (PLL is operating) is prohibited while low-speed operating mode 1 or low-speed operating mode 2 is selected by the operating power control bits in the operating power control register (OPCCR.OPCM[2:0]).

9.2.5 External Bus Clock Control Register (BCKCR)

Address(es): 0008 0030h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BCLKDIV	BCLK Pin Output Select	0: BCLK 1: 1/2 BCLK	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

BCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in operating power control register (OPCCR.OPCMTSF) is 1 (during transition)
- ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode or E2 DataFlash P/E mode) (i = 0 to 3 or D)
- Time period from WAIT instruction issuance for a transition to sleep mode, to return from sleep mode to normal operating mode

BCLKDIV Bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.
Either the BCLK clock with the frequency selected by the BCK[3:0] bits in SCKCR or the BCLK clock divided by 2 can be selected.

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Only set this register after setting the main clock oscillator wait control register according to the procedure in section 11, Low Power Consumption.

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 and main clock oscillator forced oscillation bit in the main clock oscillator forced oscillation control register (MOFCR.MOFXIN). The main clock oscillator can be started by setting the MOSTP bit to operating or by setting the MOFXIN bit to forced oscillation. When the MOFXIN bit is set to forced oscillation, the oscillator operates even in software standby mode or deep software standby mode.

When changing the value of the MOSTP bit or MOFCR.MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When a crystal oscillator is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the main clock oscillation stabilization waiting time (crystal; tMAINOSCWT) has elapsed.

When an external clock is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the EXTAL external clock input waiting time (tEXWT) has elapsed.

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 PLL operation.

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

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited 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]).

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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 waiting time (tLOCOWT) has elapsed.

That is, a fixed time for stabilization of oscillation is required after the setting for LOCO operation. 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 or deep software standby.
- When a transition to software standby or deep 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).

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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 so that the IWDT-dedicated on-chip oscillator operates, supply of the clock within the LSI only starts after a time of waiting for stabilization of the LOCO (tLOCOWT) has elapsed. If the IWDT-dedicated clock is to be used, only start using the oscillator after this waiting time (tLOCOWT) has elapsed. Ensure that oscillation by the IWDT-dedicated on-chip oscillator is stable before executing a WAIT instruction to place the chip on software standby or deep software standby.

9.2.9 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 POE2a. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE2a.	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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

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 cleared 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 or deep software standby mode. To make a transition to software standby mode or deep software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

9.2.10 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: The main clock oscillation stop has not been detected. 1: The main clock oscillation stop has been detected.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 3 system clock cycles for the bit to be read as 0.

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 bit is not cleared to 0 even though the main clock oscillation is restarted. The OSTDF bit is cleared 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 bit is cleared to 0 while the main clock oscillation is stopped, the OSTDF bit becomes 0 and then returns to 1.

When 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 bit cannot be modified to 0. The OSTDF bit should be set to 0 after switching the clock source to other sources than the main clock oscillator and PLL.

[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.

9.2.11 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MOFXIN	Main Clock Oscillator Forced Oscillation	0: Oscillator is not controlled by this bit. 1: The main clock oscillator is forcedly oscillated.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MOFXIN Bit (Main Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the main clock oscillator.

When changing the value of the MOSCCR.MOSTP bit or MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

9.2.12 Main Clock Supply Control Register (MOSCR)

Address(es): 000A 0C00h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MOSE	Main Clock Supply Control Enable	0: Main clock supply to CEC and RCR is stopped 1: Main clock supply to CEC and RCR is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

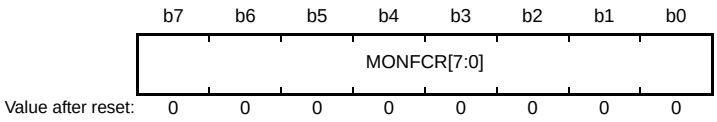
MOSE Bit (Main Clock Supply Control Enable)

This bit controls the main clock supply to the CEC and RCR.

When rewriting the MOSE bit, ensure that oscillation by the main clock is stable. Do not rewrite the MOFCR.MOFXIN bit while the MOSE bit is 1. When rewriting the MOFCR.MOFXIN bit, set the MOSE bit to 0, and ensure that the main clock supply to the CEC and RCR is stopped.

9.2.13 Main Clock Noise Filter Control Register (MONFCR)

Address(es): 000A 0C02h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	MONFCR[7:0]	Main Clock Noise Filter Control	Value other than A5h: Noise filter function enabled A5h: Noise filter function disabled	R/W

The MONFCR register is used to enable or disable the noise filter function for main clock input to be supplied to the CEC and RCR. Set this register to A5h to disable the noise filter function. Setting a value other than A5h enables the noise filter function. Rewrite this register while the MOSCR.MOSE bit is set to 0 (main clock supply to CEC and RCR is stopped).

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 resonator.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, 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 resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

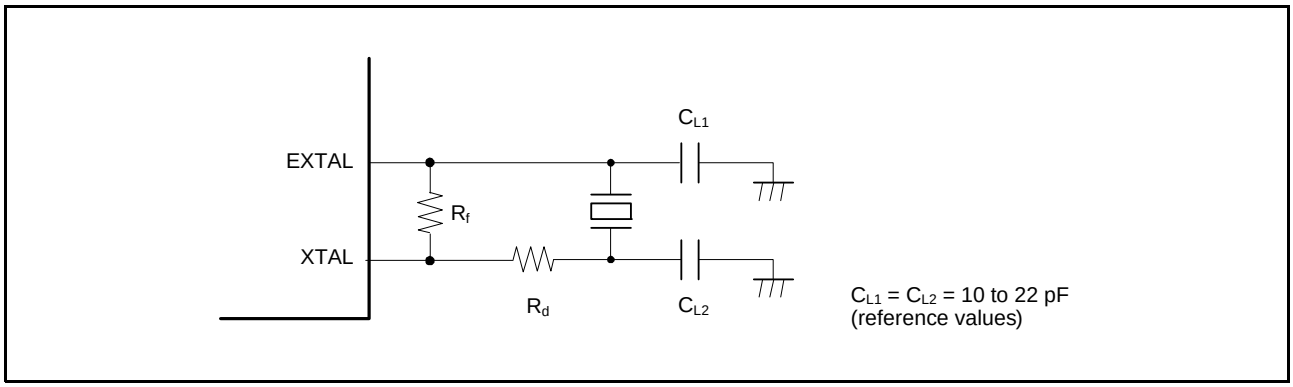


Figure 9.2 Example of Crystal Resonator Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	8	10	12.5
Rd (Ω)	200	100	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.

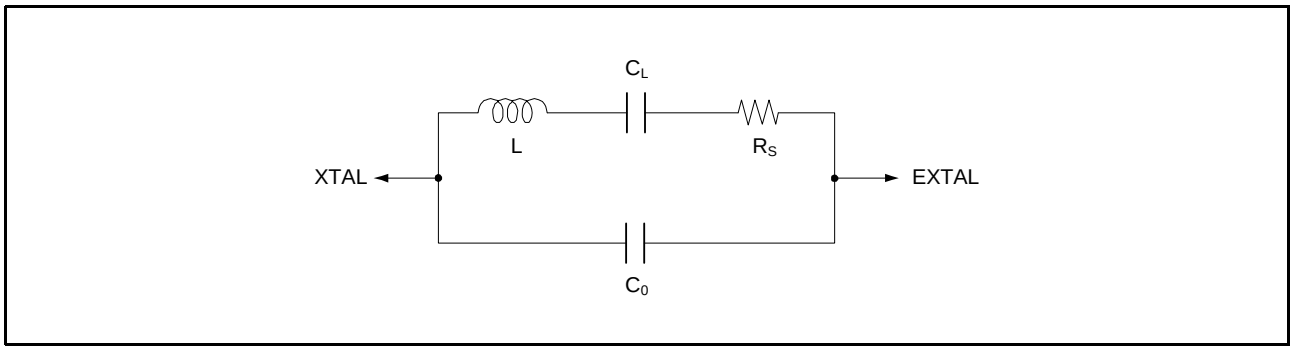


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Resonator Characteristics (Reference Values)

Frequency (MHz)	8	10	12.5
R_S max (Ω)	80	70	60
C_0 max (pF)		7	

9.3.2 External Clock Input

Figure 9.4 shows examples of connection of external clock input. To leave the XTAL pin open, make the parasitic capacitance less than 5 pF.

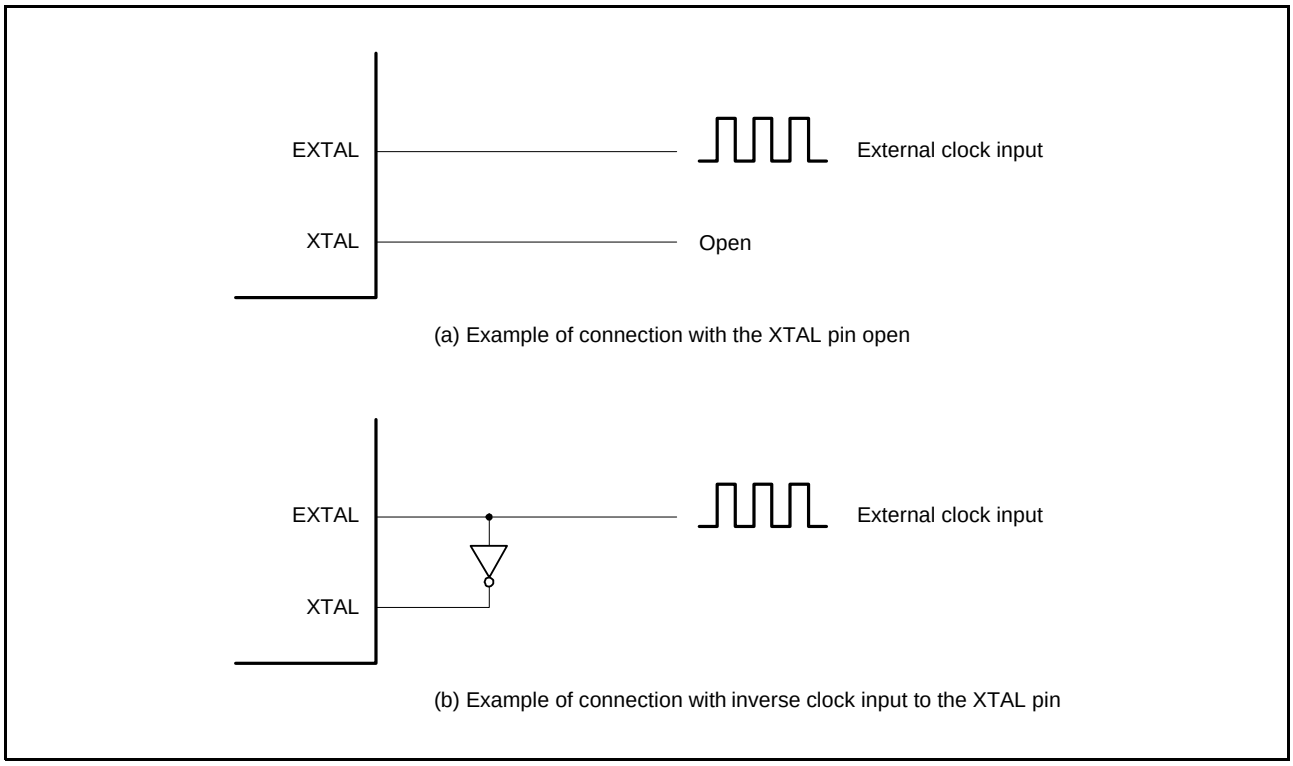


Figure 9.4 Equivalent Circuit for Crystal Resonator

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) or that of the main clock oscillator forced oscillation bit (MOFCR.MOFXIN) is 1 (forcing the main clock oscillator to 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.
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, see section 22, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 23, Port Output Enable 2 (POE2a).
In this MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (see Table 41.36 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, if an oscillation stop is detected with the main clock or PLL clock selected as the system clock source, 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). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock or PLL clock again when the OSTDF flag is cleared to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be cleared to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and clear the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the 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 a specified oscillation settling time has elapsed.
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 or deep software standby mode.
The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, PLL clock, and CAC main clock (CACMCLK), which are provided as the system clock sources.
The system clock (ICLK) frequency during the low-speed clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock select bits (SCKCR.ICK[3:0])

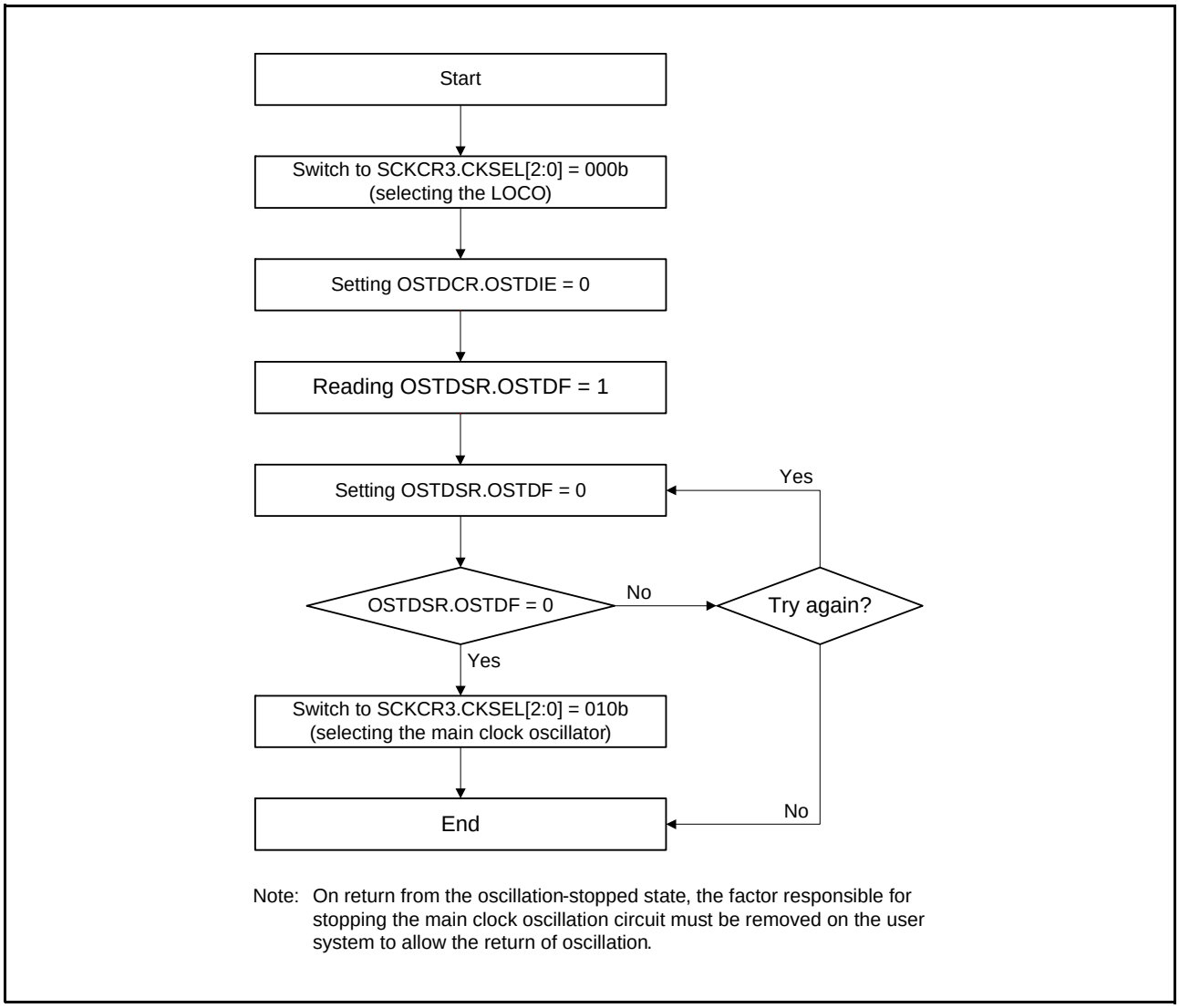


Figure 9.5 Flow of Recovery from Detection of Oscillator Stop

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 2 (POE2a). On accepting the notification of the oscillation stop, the POE2a sets the OSTST high-impedance flag in input level control/status register 3 (ICSR3.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR3.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.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, see section 14, Interrupt Controller (ICUb).

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, LOCO clock, PLL clock, dedicated clock for the IWDT, and the external clock for JTAG. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DMAC, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of peripheral modules: Peripheral module clock (PCLKB)
(3) Operating clock for the S12AD: S12AD clock (PCLKD)
(4) Operating clock of the FlashIF: FlashIF clock (FCLK)
(5) Clock for the external bus controller and external pin output: External bus clock (BCLK)
(6) Operating clock for the CAC module: CAC clock (CACCLK)
(7) Operating clock for the IWDT: IWDT-dedicated clock (IWDTCLK)
(8) Operating clock for the CEC module: CEC clock (CECCLK)
(9) Operating clock for the RCR module: RCR clock (RCRCLK)
(10) Operating clock for the JTAG: JTAG clock (JTAGTCK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits, the clock source selected by the SCKCR3.CKSEL[2:0] bits, and the bits that select the frequency of the PLL circuit (PLLCR.STC[5:0] and PLIDIV[1:0]). If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

9.6.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, ROM, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.6.2 Peripheral Module Clock

The peripheral module clock (PCLKB) is the operating clocks for use by peripheral modules.
The PCLKB frequencies are specified by the PCKB[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.6.3 S12AD Clock

The S12AD clock (PCLKD) refers to the clock for the S12AD.
The settings of the SCKCR.PCKD[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0], and PLIDIV[1:0] bits determine the frequency of the PCLKD.

9.6.4 FlashIF 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 ROM and E2 DataFlash, and reading from the E2 DataFlash.
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.

9.6.5 External Bus Clock

The external bus clock (BCLK) is an operating clock for the external bus controller. It is also output externally from the BCLK pin for the external connection bus. When the external bus is enabled, P53 that is function-multiplexed with the BCLK pin cannot be used as an I/O port.
BCLK can be output from the BCLK pin by setting the SCKCR.PSTOP1 bit to 0 and setting the external bus enable bit in the system control register 0 (SYSCR0.EXBE) to 1. Make sure that modification of the SYSCR0.EXBE bit to 1 must be performed while the PSTOP1 bit in SCKCR is 1.
When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the BCLK pin.
The BCLK frequency is specified by the BCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

9.6.6 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC.
There are three types of CACCLK: CACLCLK generated by the main clock oscillator, CACLCLK generated by the low-speed on-chip oscillator, and CACILCLK generated by the IWDT-dedicated on-chip oscillator.

9.6.7 IWDT-Dedicated Clock

The IWDT-dedicated low-speed clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.6.8 CEC Clocks

The CEC clock (CECCLK) is the operating clock for the CEC module.
There are two types of CECCLK: CECMCLK generated by the main clock oscillator, and CECILCLK generated by the IWDT-dedicated on-chip oscillator.

9.6.9 RCR Clocks

The RCR clock (RCRCLK) is the operating clock for the RCR module.
There are two types of RCRCLK: RCRMCLK generated by the main clock oscillator, and RCRILCLK generated by the IWDT-dedicated on-chip oscillator.

9.6.10 JTAG Clock

The JTAG-dedicated clock (JTAGTCK) is the operating clock for the JTAG.
JTAGTCK is generated by the external clock for JTAG (TCK).

9.7 Pin Settings When an Oscillator is Connected

(1) Main clock

Clear the main clock oscillator stop bit (MOSTP in MOSCCR) to 0 so that the clock runs, or set the main clock oscillator forced oscillation bit (MOFXIN in MOFCR) to 1 so that it is forcibly made to oscillate.

9.8 Usage Notes

9.8.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clock (PCLKB), S12AD clock (PCLKD), FlashIF clock (FCLK), and external bus clock (BCLK) 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 PCLKB. 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 system clock (ICLK) and external bus clock (BCLK).
ICLK ≥ BCLK

(3) Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.

(4) To secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then read the value from the register, and then perform the subsequent processing.

9.8.2 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.

9.8.3 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 9.6 to prevent electromagnetic induction from interfering with correct oscillation.

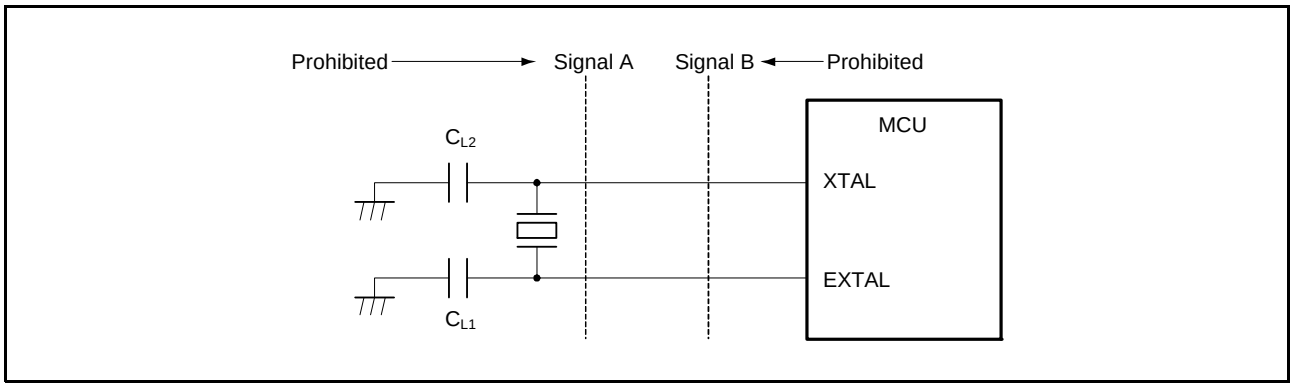


Figure 9.6 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

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 • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • LOCO clock • IWDTCLK clock • 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.

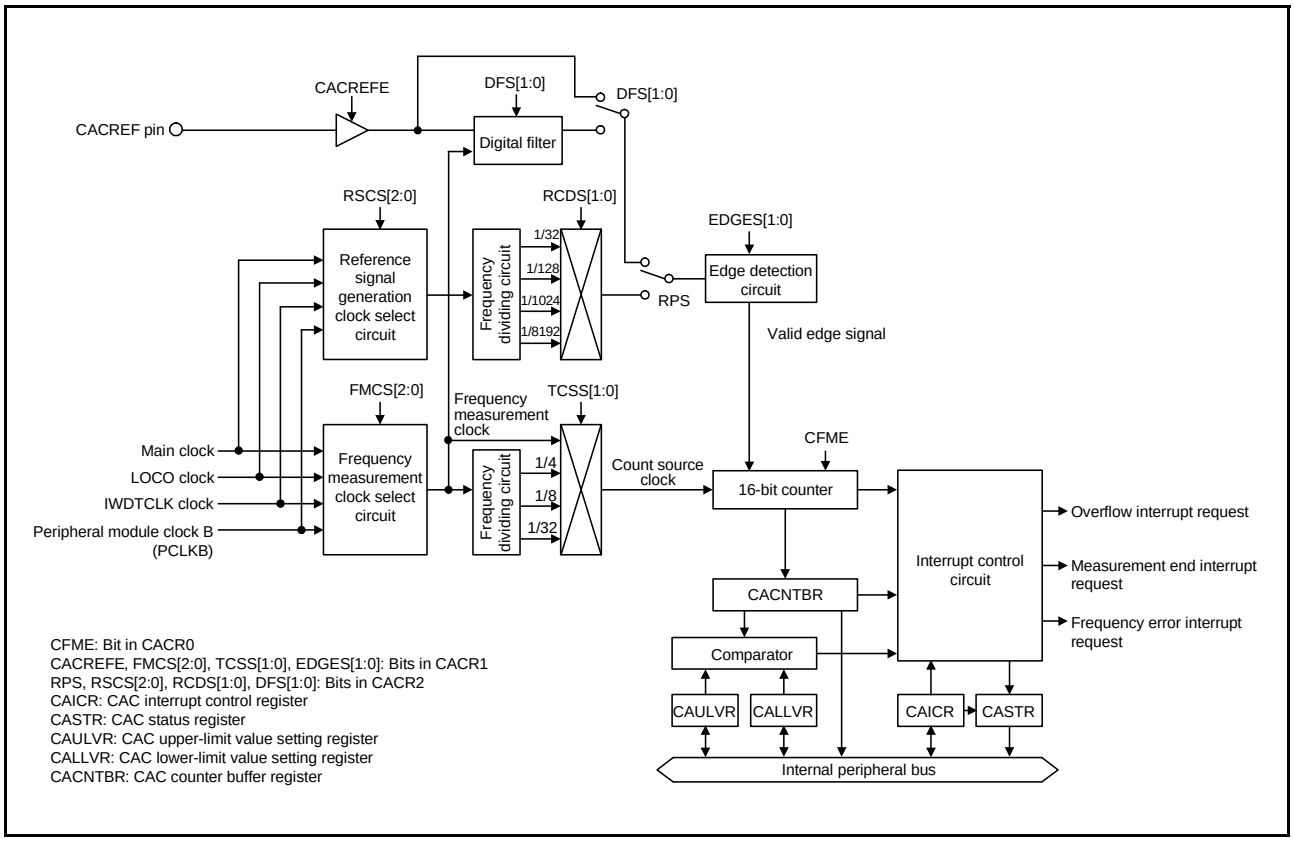


Figure 10.1 CAC Block Diagram

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): 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

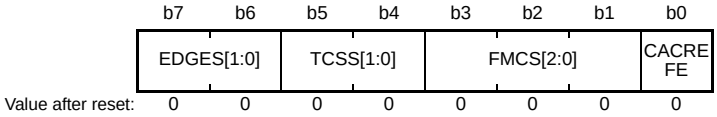
CFME Bit (Clock Frequency Measurement Enable)

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): 0008 B001h



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 1: LOCO clock 1 0 0: IWDTCLK clock 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 1. 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): 0008 B002h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

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 1: LOCO clock 1 0 0: IWDTCLK clock 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 frequency measuring clock. 1 0: The sampling clock for the digital filter is the frequency measuring clock divided by 4. 1 1: The sampling clock for the digital filter is the frequency measuring clock divided by 16.	R/W

Note 1. 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): 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): 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): 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

CAULVR 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 CACNTBR 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): 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

CALLVR 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 CACNTBR 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): 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

CACNTBR 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.

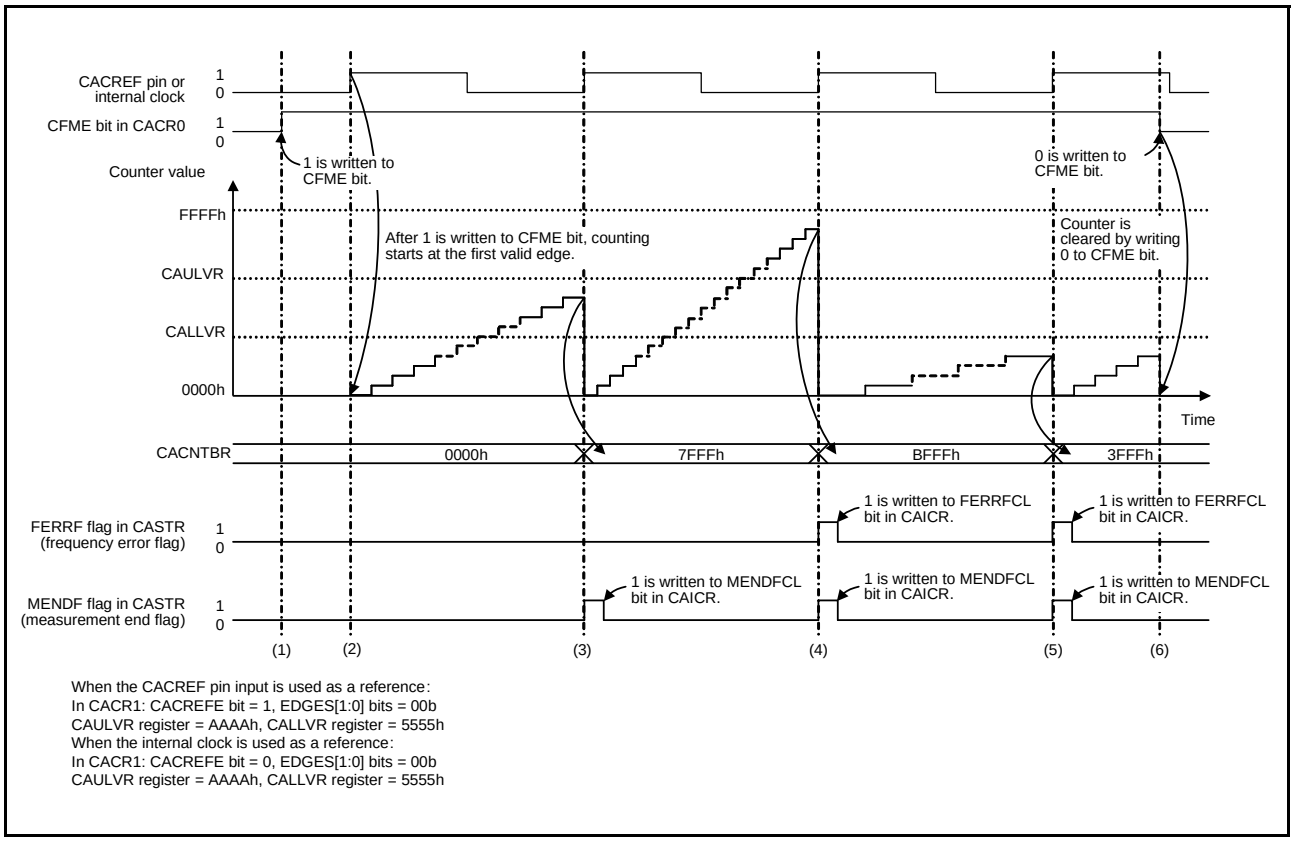


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (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 (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 (CACR1.EDGES[1:0] = 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 (CACR1.EDGES[1:0] = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1 because the

clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(6) While the CFME bit in CACR0 is 1, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 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 CACNTBR 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, BCLK output control, stopping modules, functions for low power consumption in normal operation, 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 canceling each mode.

After a reset, this LSI enters the normal program execution state, but modules except for the DMACA, 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 (PCLKB), external bus clock (BCLK), and flash interface clock (FCLK).*1
BCLK output control function	BCLK output or high-level output can be selected.*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 • Deep software standby mode
Function for lower operating power consumption	• Power consumption can be reduced in normal operation, sleep mode, and all-module clock stop mode by selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage. • Three operating power control modes High-speed operating mode Low-speed operating mode 1 Low-speed operating mode 2

Note 1. For details, see section 9, Clock Generation Circuit.

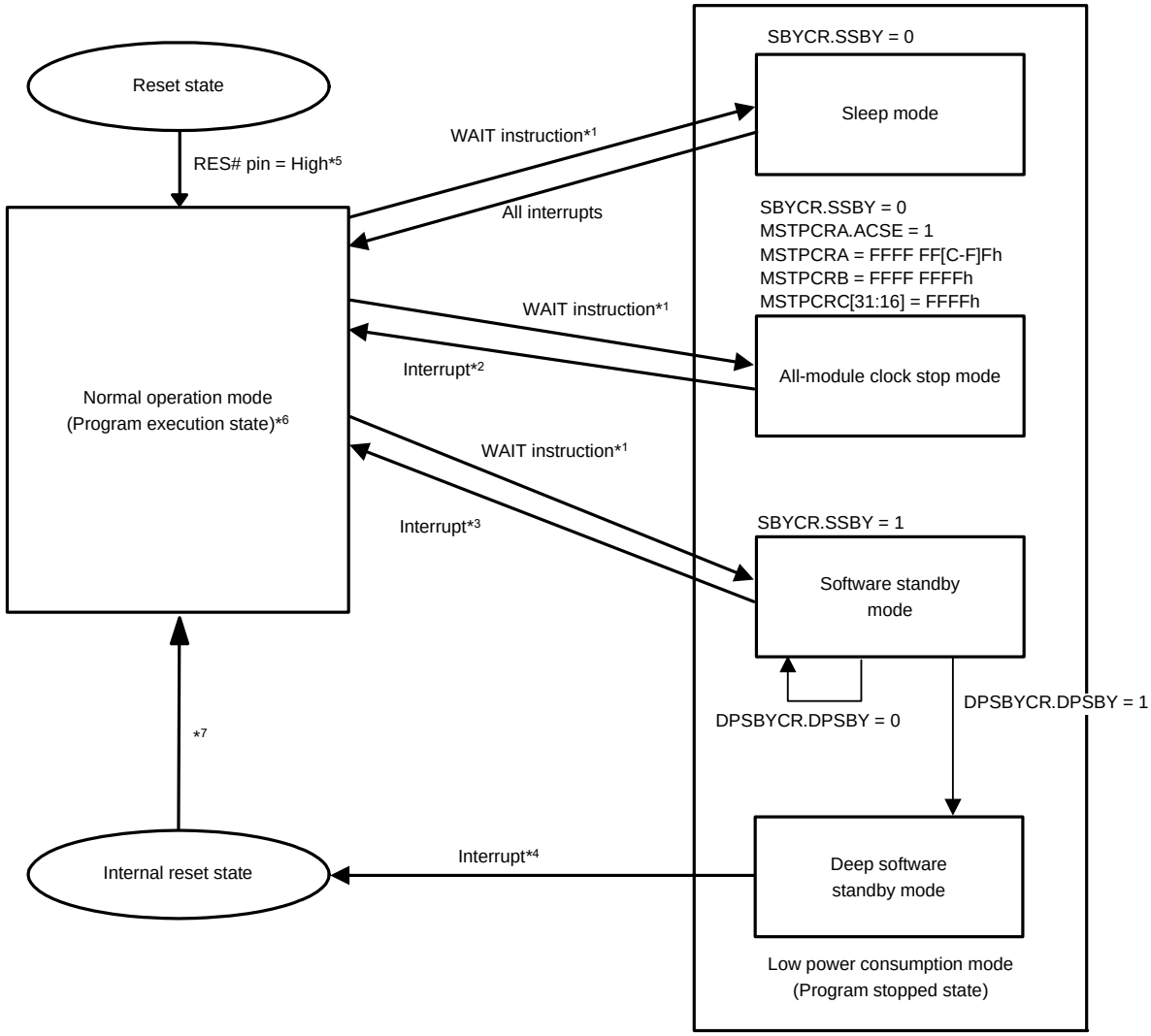
Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Canceling method other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after cancellation*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Main clock oscillator	Operating possible	Operating possible	Operating possible*5	Operating possible*5
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
Dedicated on-chip oscillator for the IWDT	Operating possible*6	Operating possible*6	Operating possible*6	Stopped (Undefined)*6
PLL	Operating possible	Operating possible	Stopped	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM1 (0001 0000h to 0001 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM0 (0000 0000h to 0000 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*7
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Independent watchdog timer (IWDT)	Operating possible*6	Operating possible*6	Operating possible*6	Stopped (Undefined)*6
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*8	Stopped (Retained)	Stopped (Undefined)
CEC transmission/reception circuit (CEC)	Operating possible	Operating possible*9	Operating possible*9	Stopped (Undefined)
Remote control signal receiver (RCR)	Operating possible	Operating possible*9	Operating possible*9	Stopped (Undefined)
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible*10, *11
Power-on reset circuit	Operating	Operating	Operating	Operating*11
Peripheral modules	Operating possible	Stopped (Retained)*12	Stopped (Retained)*12	Stopped (Undefined)*12
I/O ports	Operating	Retained*13, *14	Retained*14, *15	Retained*15

“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 IRQ12) or any of peripheral interrupts (the 8-bit timer, IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, voltage monitoring 1, voltage monitoring 2, and oscillator-stopped detection interrupts).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ12) or any of peripheral interrupts (the IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI, IRQ0-DS to IRQ7-DS, SCL0-DS, or, SDA0-DS) or any of peripheral interrupts (the IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts). However, these interrupts are enabled only when the corresponding bit in the deep standby interrupt enable registers i (DPSIERi) (i = 0, 2) is set to 1. Pins that have “-DS” appended to their names can be used as triggers for release from deep software standby.
Note 4. 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 5. Operation or stopping can be selected by the main clock oscillator forced oscillation bit (MOFXIN) in the main clock oscillator forced oscillation control register (MOFCR).
Note 6. 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. If the OFS0.IWDTSLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby 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). If the IWDTCSTPR.SLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode.
Note 7. Retention or undefined is selectable by the setting of the deep cut bits (DEEPCUT[1:0]) in the deep standby control register (DPSBYCR).
Note 8. 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 9. Operation is possible when the clock source of the operating clock is the main clock or IWDT-dedicated on-chip oscillator clock.
Note 10. If the voltage monitoring 1 circuit mode selection bit in the voltage monitoring 1 circuit control register 0 (LVD1CR0.LVD1RI) or the voltage monitoring 2 circuit mode selection bit in the voltage monitoring 2 circuit control register 0 (LVD2CR0.LVD2RI) is 1, the transition is to software standby mode rather than deep software standby mode.
Note 11. When the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]) are set to 11b and the LSI enters deep software standby mode, the voltage detection circuit stops and the low power consumption function is enabled.

Note 12. When a transition is made to all-stop module clock stop mode, software standby mode, or deep software standby mode, the 12-bit A/D converter partially enters the wait state for operation. To place the 12-bit A/D converter fully in the standby state, set the MSTPCRA.MSTPA24 bit to 1.
Note 13. If pin P53 is being used for the BCLK signal, operation continues with as-is output of BCLK. While the 8-bit timer is operated, the related pins continue operating.
Note 14. While the CEC transmission/reception circuit (CEC) and the remote control signal receiver circuit (RCR) are operated, related pins continue operating.
Note 15. Retention of levels or placement in the high-impedance state is selectable for the address bus and bus control signals (CS0# to CS3#, RD#, WR0#, WR1#, WR#, BC0#, BC1#, and ALE) by the output port enable bit (OPE) in the standby control register (SBYCR).



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 IRQ12) or any of peripheral interrupts (the 8-bit timer, IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, 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 IRQ12) or any of peripheral interrupts (the IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, voltage-monitoring 1, and voltage-monitoring 2 interrupts).

Note 4. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI, IRQ0-DS to IRQ7-DS, SCL0-DS, or SDA0-DS) or any of peripheral interrupts (the voltage-monitoring 1, and voltage-monitoring 2 interrupts). However, these interrupts are enabled only when the corresponding bit in the deep standby interrupt enable registers I (DPSIERi) (i = 0, 2) is set to 1.

Note 5. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode.

Note 6. 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 sleep mode, the clock source will be the same before and at the time of the transition.

Note 7. When an interrupt request as indicated in Note 4 above is generated, an internal reset (deep software standby reset) is generated over a fixed period. Release from deep software standby mode accompanies release from the internal reset state. The transition from deep software standby mode is to normal operating mode, and reset exception processing starts with the LOCO as the source of the operating clock.

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. The voltage-monitoring 1 reset and voltage-monitoring 2 reset cannot be used to cancel deep software standby mode.

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	OPE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state. 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state.	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

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS3#, RD#, WR0#, WR1#, WR#, BC0#, BC1#, and ALE) in software standby mode or deep software standby mode, or to set the output to the high-impedance state.

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 LSI enters software standby mode after execution of the WAIT instruction. When the LSI returns to normal 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 (OSTDCR.OSTDE) is 1, setting of the SSBY bit is invalid. Even if the SSBY bit is 1, the LSI will enter sleep mode or all module clock stop mode on execution of the WAIT instruction.

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	—	MSTPA29	MSTPA28	MSTPA27	—	—	MSTPA24	—	—	—	—	MSTPA19	—	MSTPA17	—
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
	MSTPA15	MSTPA14	MSTPA13	—	MSTPA11	—	MSTPA9	—	—	—	MSTPA5	MSTPA4	—	—	—	—
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	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: The module-stop state is canceled 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: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b8 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 2 Module Stop	Target module: MTU (MTU0 to MTU5) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b12	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit 0 (Unit 0) Module Stop	Target module: TPU unit 0 (TPU0 to TPU5) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: The module-stop state is canceled 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: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-bit A/D Converter (Unit 1) Module Stop	Target module: S12AD1 0: The module-stop state is canceled 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
b19	MSTPA19	D/A Converter Module Stop	Target module: DA 0: The module-stop state is canceled 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	MSTPA24	12-Bit A/D Converter Control Block Module Stop	Target module: S12AD control block 0: The module-stop state is canceled 1: Transition to the module-stop state is made	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, make sure that 1 has been written to this bit.	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: The module-stop state is canceled 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, make 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: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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, and MSTPCRC satisfying specified conditions, the LSI enters all-module clock stop mode. For details, see section 11.6.2, 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.

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
	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	MSTPB26	MSTPB25	MSTPB24	MSTPB23	—	MSTPB21	MSTPB20	—	—	MSTPB17	MSTPB16
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
	—	—	—	—	—	—	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	This bit is read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCId Module Stop	Target module: SCId (SCI12) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: The module-stop state is canceled 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	ELC Module Stop	Target module: ELC 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b15 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b16	MSTPB16	Serial Peripheral Interface 1 Module Stop	Target module: RSPI1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b19, b18	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b20	MSTPB20	I^2C Bus Interface 1 Module Stop	Target module: RIIC1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: The module-stop state is canceled 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: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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
	—	MSTPC30	MSTPC29	MSTPC28	MSTPC27	MSTPC26	MSTPC25	MSTPC24	—	—	—	—	MSTPC19	—	—	MSTPC16
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC1	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	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0000 FFFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b1	MSTPC1	RAM1 Module Stop*1	Target module: RAM1 (0001 0000h to 0001 FFFFh) 0: RAM1 operating 1: RAM1 stopped	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	MSTPC16	I^2C Bus Interface 3 Module Stop	Target module: RIIC3 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b18, b17	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop	Target module: CAC 0: The module-stop state is canceled 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 Communication Interface 11 Module Stop	Target module: SCI11 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b25	MSTPC25	Serial Communication Interface 10 Module Stop	Target module: SCI10 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b26	MSTPC26	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b28	MSTPC28	RCR1 Module Stop	Target module: RCR1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b29	MSTPC29	RCR0 Module Stop	Target module: RCR0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b30	MSTPC30	CEC Module Stop	Target module: CEC 0: The module-stop state is canceled 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

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. The MSTPC1 or MSTPC0 bit should not be set to 1 during access to the corresponding on-chip RAM. The corresponding RAM should not be accessed while the MSTPC1 or 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 rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 0 0: High-speed operating mode 1 1 0: Low-speed operating mode 1 1 1 1: Low-speed operating mode 2 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	• Read 0: Transition completed 1: During transition • Write The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

OPCCR is used to reduce power consumption in normal operating mode, sleep mode, and all-module clock stop mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

OPCCR should not be modified in the following cases:

- When the operating power control mode transition status flag (OPCMTSF) is 1 (operating power control mode switching is in progress)
- When the ROM P/E mode entry bit i in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (i = 0 to 3, D)
- Period from the time of WAIT instruction execution for a sleep mode transition, to return from sleep mode to normal operation

Writing to the flash memory while it is being programmed or erased is impossible because write access to the OPCCR register is not allowed.

For the procedure to use in shifting to operating power control mode, refer to section 11.5, Function for Lower Operating Power Consumption.

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and all-module clock stop mode.

Table 11.3 shows the operating power control modes along with the operating frequency ranges, operating voltage ranges, and power consumption.

Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption

Operating Power Control Mode	OPCM[2:0] Bits	Operating Frequency Range					Operating Voltage Range		Power Consumption
		Flash Memory Read				Flash Memory P/E	Flash Memory Read	Flash Memory P/E	
		ICLK	FCLK	PCLKB	BCLK	FCLK			
High-speed operating mode	000b	54 MHz max	32 MHz max	32 MHz max	54 MHz max	4 to 32 MHz	3-V package: 2.7 to 3.6 V 5-V package: 4.0 to 5.5V	3-V package: 2.7 to 3.6 V 5-V package: 4.0 to 5.5V	High
Low-speed operating mode 1	110b	1 MHz max	1 MHz max	1 MHz max	1 MHz max	P/E disabled	3-V package: 2.7 to 3.6 V 5-V package: 4.0 to 5.5V	P/E disabled	
Low-speed operating mode 2	111b	32 to 125 kHz	32 to 125 kHz	125 kHz max	125 kHz max	P/E disabled	3-V package: 2.7 to 3.6 V 5-V package: 4.0 to 5.5V	P/E disabled	Low

Each operating power control mode is described below.

- High-speed operating mode

This mode allows high-speed operation.
During reading of the flash memory (FLASH), the maximum operating frequency of ICLK and BCLK is 54 MHz, and that of FCLK and PCLKB is 32 MHz. During flash memory programming/erasure (P/E), the FCLK can be operated at an operating frequency from 4 to 32 MHz. The operating voltage is in the range of 2.7 to 3.6 V in the 3-V package and 4.0 to 5.5 V in the 5-V package both for flash memory read and P/E. After a reset is canceled, operation is started from this mode.

- Low-speed operating mode 1

This mode reduces power consumption for low-speed operation.
During reading of the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKB and BCLK is 1 MHz. The operating voltage is in the range of 2.7 to 3.6 V in the 3-V package and 4.0 to 5.5 V in the 5-V package.
In low-speed operating mode 1, P/E operation of flash memory is disabled, and writing to set the PLLCR2.PLLEN bit to 0 (PLL operation) is prohibited.
In this mode, lower power consumption is possible than in high-speed operating mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

- Low-speed operating mode 2

As compare to low-speed operating mode 1, this mode reduces power consumption for low-speed operation.
During reading of the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKB and BCLK is 125 kHz, and the minimum operating frequency of ICLK and FCLK is 32 kHz. The operating voltage is in the range of 2.7 to 3.6 V in the 3-V package and 4.0 to 5.5 V in the 5-V package.
The following restrictions apply when low-speed operating mode 2 is selected:

- P/E operations for flash memory are prohibited.
- Reading of data flash is prohibited.
- Using the PLL is prohibited.
- Using the oscillation stop detection function of the main clock oscillator is prohibited.
- Using the clock source switching function of the ELC is prohibited.

In this mode, lower power consumption is possible than in low-speed operating mode 1 when the same operation is performed under the same conditions (operating frequency, operating voltage).

When the PLL stop control bit (PLLCR2.PLLEN) in PLL control register 2 is 0 (PLL operation), writing 110b (low-speed operating mode 1) and 111b (low-speed operating mode 2) to the OPCM[2:0] bits is not possible.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched.
When a write access is attempted to change the operating power control mode, the OPCMTSF flag is set to 1. The flag becomes 0 after a transition to the changed control mode is completed. Make sure that the OPCMTSF flag is 0 (completed operating power control mode transition) before the next processing.

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 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 at sleep mode cancellation is disabled 1: Clock source switching at sleep mode cancellation is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

RSTCKCR is used to control clock source switching at cancellation of 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) 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]).
When return from sleep mode is made while clock source switching at sleep mode cancellation is enabled (RSTCKCR.RSTCKEN is 1), and operating power control mode select bits (OPCCR.OPCM[2:0]) are set so as to select low-speed operating mode 1 (110b) or low-speed operating mode 2 (111b), the OPCCR.OPCM[2:0] bits are automatically switched to high-speed mode (000b).
Do not use the clock source switching function at returning from sleep mode and that of the ELC at the same time.
To enable the clock source switching function at returning from sleep mode, set the RSTCKEN bit to 1 while that of the ELC is disabled. To enable the clock source switching function of the ELC, make sure that the RSTCKEN bit i s 0.

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used when sleep mode is canceled.
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 when sleep mode is canceled.
When sleep mode is canceled, 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 main clock, or PLL selected as the clock source, the RSTCKEN bit should not be set to 1.

11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Waiting Time	b4 b0 0 0 0 0 0: Waiting time = 2 cycles 0 0 0 0 1: Waiting time = 4 cycles 0 0 0 1 0: Waiting time = 8 cycles 0 0 0 1 1: Waiting time = 16 cycles 0 0 1 0 0: Waiting time = 32 cycles 0 0 1 0 1: Waiting time = 64 cycles 0 0 1 1 0: Waiting time = 512 cycles 0 0 1 1 1: Waiting time = 1024 cycles 0 1 0 0 0: Waiting time = 2048 cycles 0 1 0 0 1: Waiting time = 4096 cycles 0 1 0 1 0: Waiting time = 16384 cycles 0 1 0 1 1: Waiting time = 32768 cycles 0 1 1 0 0: Waiting time = 65536 cycles 0 1 1 0 1: Waiting time = 131072 cycles 0 1 1 1 0: Waiting time = 262144 cycles 0 1 1 1 1: Waiting time = 524288 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MOSCWTCR is used to control the time to wait until supplying the output of the main clock oscillator to the internal circuits. Supply of the main clock signal within the MCU starts after counting of the number of cycles of the main clock specified in the MOSCWTCR register plus 16384.

Set the MSTS[4:0] bits so that the waiting time is at least as long as the main clock oscillator stabilization time ($t_{MAINOSC}$). Only use the main clock after the main clock oscillation settling time ($t_{MAINOSCWT}$) has elapsed. Regarding the main clock oscillation settling time, refer to Table 41.18.

The waiting time is not required when the main clock is being externally input to the main clock oscillator. Only write new values to MOSCWTCR while the MOSCCR.MOSTP bit is 1; do not write to MOSCWTCR if this is not the case.

Example: When oscillation is at 12 MHz and the crystal resonator is to be used after 10 ms (= 10000 µs) of stabilization time

To satisfy the relation waiting time ≥ $t_{MAINOSC} \times f_{MAIN}$ = 10000 [µs] × 12 [MHz] = 120000 [cycles], set the MSTS[4:0] bits to 01101b (131072 cycles).

At this time, when the main clock stabilization time is calculated from the formula in Table 41.18,

$$t_{MAINOSCWT} = t_{MAINOSC} + \frac{n + 16384}{f_{MAIN}}$$

$$= 10000\ [\mu s] + \frac{131072\ [cycles] + 16384}{12\ [MHz]}$$

$$= 22288\ [\mu s]$$

22288 µs are required from the time the main clock starts oscillating until it is usable.

11.2.8 PLL Wait Control Register (PLLWTCR)

Address(es): 0008 00A6h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSTS[4:0]	PLL Waiting Time	b4 b0 0 0 0 0 0: Waiting time = 16 cycles 0 0 0 0 1: Waiting time = 32 cycles 0 0 0 1 0: Waiting time = 64 cycles 0 0 0 1 1: Waiting time = 512 cycles 0 0 1 0 0: Waiting time = 1024 cycles 0 0 1 0 1: Waiting time = 2048 cycles 0 0 1 1 0: Waiting time = 4096 cycles 0 0 1 1 1: Waiting time = 16384 cycles 0 1 0 0 0: Waiting time = 32768 cycles 0 1 0 0 1: Waiting time = 65536 cycles 0 1 0 1 0: Waiting time = 131072 cycles 0 1 0 1 1: Waiting time = 262144 cycles 0 1 1 0 0: Waiting time = 524288 cycles 0 1 1 0 1: Waiting time = 1048576 cycles 0 1 1 1 0: Waiting time = 2097152 cycles 0 1 1 1 1: Waiting time = 4194304 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

PLLWTCR is used to control the time to wait until the output of the PLL is supplied to the internal circuits.
Supply of the PLL clock signal within the MCU starts after counting of the number of cycles of the PLL clock specified in the PLLWTCR register + 131072.
Set the PSTS[4:0] bits so that the waiting time is greater than the PLL clock stabilization time (tPLL1 or tPLL2). Only use the PLL clock after waiting for the PLL clock oscillation stabilization waiting time (tPLLWT1 or tPLLWT2) to elapse after oscillation of the PLL has started. See Table 41.18 for the PLL clock oscillation stabilization waiting time.
New values can only be written to the PLLWTCR register while the PLLCR2.PLLEN bit is 1 (stopping the PLL). Do not write to the register if this is not the case.

Example: When oscillation is at 12 MHz, the crystal resonator is to be used after 10 ms (= 10000 µs) of stabilization time, and the PLL is to oscillate at 200 MHz

- When the PLL starts operating after oscillation of the main clock has become stable

To satisfy the relation waiting time ≥ $t_{PLL1} \times f_{PLL}$ = 500 [µs] × 200 [MHz] = 100000 [cycles], set the PSTS[4:0] bits to 01010b (131072 cycles).
At this time, when the PLL clock stabilization time is calculated from the formula in Table 41.18,

$$t_{PLLWT1} = t_{PLL1} + \frac{n + 131072}{f_{PLL}}$$

$$= 500\ [\mu s] + \frac{131072\ [cycles] + 131072}{200\ [MHz]}$$

$$= 1810.72\ [\mu s]$$

1811 µs of waiting are required from the time the PLL clock starts oscillating until it is usable.

- When the PLL starts operating before oscillation of the main clock has become stable

To satisfy the relation waiting time ≥ ($t_{MAINOSC}$ + t_{PLL1}) × t_{PLL} = (10000 [µs] + 500 [µs]) × 200 [MHz] = 2100000 [cycles], set the PSTS[4:0] bits to 01111b (4194304 cycles).

At this time, when the PLL clock stabilization time is calculated from the formula in Table 41.18,

$$t_{PLLWT2} = t_{PLL2} + \frac{n + 131072}{f_{PLL}}$$

$$= t_{MAINOSC} + t_{PLL1} + \frac{n + 131072}{f_{PLL}}$$

$$= 10000\ [\mu s] + 500\ [\mu s] + \frac{4194304\ [cycles] + 131072}{200\ [MHz]}$$

$$= 32126.88\ [\mu s]$$

about 32.13 ms of waiting are required from the time the PLL clock starts oscillating until it is usable.

11.2.9 Deep Standby Control Register (DPSBYCR)

Address(es): 0008 C280h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DEEPCUT [1:0]	Deep Cut	b1 b0 0 0: Power is supplied to the RAM (RAM0*1) in deep software standby mode 0 1: Power is not supplied to the RAM (RAM0*1) in deep software standby mode 1 0: Setting prohibited 1 1: Power is not supplied to the RAM (RAM0*1) in deep software standby mode. In addition, LVD is stopped and the low power consumption function in a power-on reset circuit is enabled.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Deep software standby mode and I/O port retention are canceled simultaneously. 1: The I/O port state is retained even after deep software standby mode is canceled. Then, writing 0 to the IOKEEP bit cancels the I/O port retention.	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 1 0: Transition to software standby mode is made after the WAIT instruction is executed 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. For the RAM address space, see Table 11.2.

DPSBYCR is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DEEPCUT[1:0] Bits (Deep Cut)

The DEEPCUT[1:0] bits control the internal power supply to the RAM in deep software standby mode. In addition, these bits control the state of LVD and power-on reset circuit in deep software standby mode.
The internal power supply of RAM0 can be controlled by the setting of the DEEPCUT[1:0] bits.
When an LVD interrupt is used in deep software standby mode, the DEEPCUT[1:0] bits must be set to 00b or 01b.
For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power consumption function of the power-on reset circuit is enabled.
The internal power supply of RAM1 is stopped in deep software standby mode regardless of the setting of the DEEPCUT[1:0] bits.

IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports keep retaining the same states from software standby mode. The IOKEEP bit specifies whether to keep retaining the I/O port states from deep software standby mode even after deep software standby mode is canceled, or to cancel retaining the I/O port states.

DPSBY Bit (Deep Software Standby)

The DPSBY bit controls transitions to deep software standby mode.

When the WAIT instruction is executed while the SBYCR.SSBY and DPSBY bits are both 1, the LSI enters deep software standby mode through software standby mode.
The DPSBY bit remains 1 when deep software standby mode is canceled by certain pins which are sources of external interrupts (NM1, IRQ0-DS to IRQ7-DS, SCL0-DS, and SDA0-DS) or a peripheral interrupt (voltage monitoring 1 or voltage monitoring 2). Write 0 to this bit to clear it.
The setting of the DPSBY bit becomes invalid when the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP is 0 (counting continues) or the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
Instead, even when the SBYCR.SSBY bit is 1 and the DPSBY bit 1, the transition after the execution of a WAIT instruction is to software standby mode.
The setting of the DPSBY bit becomes invalid when voltage monitoring 1 reset is enabled by the voltage monitoring 1 circuit mode select bit (LVD1CR0.LVD1RI = 1) or when a voltage monitoring 2 reset is selected by the voltage monitoring 2 circuit mode bit (LVD2CR0.LVD2RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WAIT instruction is to software standby mode.

11.2.10 Deep Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Canceling deep software standby mode by the IRQ0-DS pin is disabled 1: Canceling deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Canceling deep software standby mode by the IRQ1-DS pin is disabled 1: Canceling deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Canceling deep software standby mode by the IRQ2-DS pin is disabled 1: Canceling deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Canceling deep software standby mode by the IRQ3-DS pin is disabled 1: Canceling deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Canceling deep software standby mode by the IRQ4-DS pin is disabled 1: Canceling deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Canceling deep software standby mode by the IRQ5-DS pin is disabled 1: Canceling deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Canceling deep software standby mode by the IRQ6-DS pin is disabled 1: Canceling deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Canceling deep software standby mode by the IRQ7-DS pin is disabled 1: Canceling deep software standby mode by the IRQ7-DS pin is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER0 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER0 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR0 being set to 1. However, when DPSIEGR0 is 0, the rising edge is not detected, resulting in DPSIFR0 not being set to 1.

11.2.11 Deep Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): 0008 C284h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRIICCIE	DRIICDIE	DNMIE	—	—	DLVD2IE	DLVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IE	LVD1 Deep Standby Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage monitoring 1 signal 1: Enable canceling deep software standby mode by the voltage monitoring 1 signal	R/W
b1	DLVD2IE	LVD2 Deep Standby Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage monitoring 2 signal 1: Enable canceling deep software standby mode by the voltage monitoring 2 signal	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIE	NMI Pin Enable	0: Canceling deep software standby mode by the NMI pin is disabled 1: Canceling deep software standby mode by the NMI pin is enabled	R/W*1
b5	DRIICDIE	SDA0-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SDA0-DS signal is disabled 1: Canceling deep software standby mode by the SDA0-DS signal is enabled	R/W
b6	DRIICCIE	SCL0-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SCL0-DS signal is disabled 1: Canceling deep software standby mode by the SCL0-DS signal is enabled	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

DPSIER2 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER2 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR2 being set to 1. However, when DPSIEGR2 is 0, the rising edge is not detected, resulting in DPSIFR2 not being set to 1.

11.2.12 Deep Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): 0008 C286h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ0-DS pin is generated 1: A cancel request by the IRQ0-DS pin is generated	R/(W) *1
b1	DIRQ1F	IRQ1-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ1-DS pin is generated 1: A cancel request by the IRQ1-DS pin is generated	R/(W) *1
b2	DIRQ2F	IRQ2-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ2-DS pin is generated 1: A cancel request by the IRQ2-DS pin is generated	R/(W) *1
b3	DIRQ3F	IRQ3-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ3-DS pin is generated 1: A cancel request by the IRQ3-DS pin is generated	R/(W) *1
b4	DIRQ4F	IRQ4-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ4-DS pin is generated 1: A cancel request by the IRQ4-DS pin is generated	R/(W) *1
b5	DIRQ5F	IRQ5-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ5-DS pin is generated 1: A cancel request by the IRQ5-DS pin is generated	R/(W) *1
b6	DIRQ6F	IRQ6-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ6-DS pin is generated 1: A cancel request by the IRQ6-DS pin is generated	R/(W) *1
b7	DIRQ7F	IRQ7-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ7-DS pin is generated 1: A cancel request by the IRQ7-DS pin is generated	R/(W) *1

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR0 is cleared to 00h.
To clear DPSIFR0 to 00h after modifying DPSIER0, wait for at least six PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.
DPSIFR0 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 0 to 7)

These flags indicate that a cancel request by the IRQn-DS pin has been generated.
[Setting condition]

- A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0

11.2.13 Deep Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): 0008 C288h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRIICCIF	DRIICDIF	DNMIF	—	—	DLVD2IF	DLVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IF	LVD1 Deep Standby Cancel Flag	0: No cancel request by the voltage monitor 1 signal is generated 1: A cancel request by the voltage monitor 1 signal is generated	R/(W) *1
b1	DLVD2IF	LVD2 Deep Standby Cancel Flag	0: No cancel request by the voltage monitor 2 signal is generated 1: A cancel request by the voltage monitor 2 signal is generated	R/(W) *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIF	NMI Deep Standby Cancel Flag	0: No cancel request by the NMI pin is generated 1: A cancel request by the NMI pin is generated	R/(W) *1
b5	DRIICDIF	SDA0-DS Deep Standby Cancel Flag	0: No cancel request by the SDA0-DS signal is generated 1: A cancel request by the SDA0-DS signal is generated	R/(W) *1
b6	DRIICCIF	SCL0-DS Deep Standby Cancel Flag	0: No cancel request by the SCL0-DS signal is generated 1: A cancel request by the SCL0-DS signal is generated	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.
Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR2 is cleared to 00h.
To clear DPSIFR2 to 00h after modifying DPSIER2, wait for at least six PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.
DPSIFR2 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DLVDmIF Flag (LVDm Deep Standby Cancel Flag) (m = 1 or 2)

This flag indicates that a cancel request by the voltage monitor m signal has been generated.
[Setting condition]

- A cancel request is generated by the voltage monitoring m signal that is selected in DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

DNMIF Flag (NMI Deep Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.
[Setting condition]

- A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICDIF Flag (SDA0-DS Deep Standby Cancel Flag)

This flag indicates that a cancel request by the SDA0-DS signal has been generated.

[Setting condition]

- A cancel request by the SDA0-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICCIF Flag (SCL0-DS Deep Standby Cancel Flag)

This flag indicates that a cancel request by the SCL0-DS signal has been generated.

[Setting condition]

- A cancel request by the SCL0-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

11.2.14 Deep Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): 0008 C28Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.15 Deep Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): 0008 C28Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRIICCEG	DRIICDEG	DNMIEG	—	—	DLVD2EG	DLVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1EG	LVD1 Edge Select	0: A cancel request is generated when VCC < Vdet1 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DLVD2EG	LVD2 Edge Select	0: A cancel request is generated when VCC < Vdet2 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet2 (rise) is detected	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DRIICDEG	SDA0-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DRIICCEG	SCL0-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.16 Deep Standby Backup Register (DPSBKRy) (y = 0 to 31)

Address(es): 0008 C2A0h to 0008 C2BFh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

DPSBKRy is a 32-byte readable/writable register to store data during deep software standby mode.
The value of this register is retained even in deep software standby mode where RAM data is not retained.
DPSBKRy is not initialized, and the register value is undefined immediately after power-on.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], and PCKB[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, ROM, and RAM operate on the operating clock specified by the ICK[3:0] bits.

Peripheral modules operate on the operating clock specified by the PCKB[3:0] bit.

The flash memory interface operates on the operating clock specified by the FCK[3:0] bits.
The external bus operates on the operating clock specified by the BCK[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 C, i = 31 to 0) in MSTPCRA to MSTPCRC is set to 1, the specified module-stops operating and enters the module-stop state, but the CPU continues to operate independently. Clearing the MSTPmi bit to 0 cancels the module-stop state, allowing the module to restart operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After a reset is canceled, all modules other than 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 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal operation, sleep mode and all-module clock stop mode.

11.5.1 Setting Operating Power Consumption Control Mode

Examples of the procedures for switching operating power consumption control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

Example: From high-speed operation mode to low-speed operation mode 1

(High-speed operation in the operating power consumption control mode used before mode-switching)
↓
Set to switch from the main clock to the LOCO clock (clock source and frequency division ratio)
↓
Write to OPCCR
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
(Low-speed operation in the switched operating power consumption control mode)

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

Example: From low-speed operation mode 2 to high-speed operation mode

(Low-speed operation in the operating power consumption control mode used before mode-switching)
↓
Write to OPCCR
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
Set to switch from the LOCO clock to the main clock (clock source and frequency division ratio)
↓
(High-speed operation in the switched operating power consumption control mode)

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 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. 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 to be used for recovery from sleep mode to the CPU.*2
(3) Set the priority*2 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*2 for the 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, Interrupt Controller (ICUb).

11.6.1.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, the reset signal from the RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Canceling by an interrupt
  When an interrupt occurs, sleep mode is canceled and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), sleep mode is not canceled.
- Canceling by the RES# pin reset
  When the RES# pin is driven low, the LSI 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 handling.
- Canceling by a power-on reset
  Sleep mode is canceled by a power-on reset.
- Canceling by a voltage monitoring reset
  Sleep mode is canceled by a voltage monitoring reset from the voltage detection circuit.
- Canceling by the independent watchdog timer reset
  Sleep mode is canceled 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 sleep mode cannot be canceled by the independent watchdog timer reset.

Note 1. For details, see section 14, Interrupt Controller (ICUb).
Note 2. For details, see section 2, CPU.

11.6.1.3 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). In addition, for details on the oscillation settling time, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.6.2 All-Module Clock Stop Mode

11.6.2.1 Transition to All-Module Clock Stop Mode

After setting the MSTPCRA.ACSE bit to 1 and placing modules controlled by MSTPCRA, MSTPCRB, and MSTPCRC registers in the module-stop state (MSTPCRA = FFFF FF[C-F]Fh, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = 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, POE*6, IWDT, CEC*2, RCR*2, 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. 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 to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*5 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*5 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. The CEC and RCR operate when the clock source of the operating clock is the main clock or IWDT-dedicated on-chip oscillator.
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 DMACA are not activated.
Note 4. For details, see section 2, CPU.
Note 5. For details, see section 14, Interrupt Controller (ICUb).
Note 6. When a POE interrupt source condition is satisfied while the setting to enable POE 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 POE interrupt is generated after recovery.

11.6.2.2 Canceling All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ12), a peripheral interrupt (8-bit timer*1, IWDT*2, CEC (INTDAA, INTCEA, INTERRA)*3, RCR*3, 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*4 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*5 of the CPU) or a maskable interrupt has been set up as a trigger to activate the DTC or DMAC, the interrupt will not cancel all-module clock stop mode.

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. The CEC and RCR operate when the clock source of the operating clock is the main clock or IWDT-dedicated on-chip oscillator.
Note 4. For details, see section 14, Interrupt Controller (ICUb).
Note 5. For details, see section 2, CPU.

11.6.3 Software Standby Mode

11.6.3.1 Transition to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1 and the DPSBYCR.DPSBY bit cleared to 0, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions*3, and all 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. Whether the address bus and bus control signals are placed in the high-impedance or the output state is retained can be specified by the SBYCR.OPE bit. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.

Clear 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. 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) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination to be used for recovery from software standby mode to the CPU.*2
(3) Set the priority*2 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*2 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, Interrupt Controller (ICUb).
Note 3. The CEC and RCR operate when the clock source of the operating clock is the main clock or IWDT-dedicated on-chip oscillator.

11.6.3.2 Canceling Software Standby Mode

Release from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ12), peripheral interrupts (the IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, 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 operating before the transition to software standby are restarted. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Release due to an interrupt
  When an interrupt request from the NMI, IRQ0 to IRQ12, IWDT, CEC (INTDAA, INTCEA, INTERRA), RCR, 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 set by the MOSCWTCR.MSTS[4:0], SOSCWTCR.SSTS[4:0], or PLLWTCR.PSTS[4:0] bits has elapsed, the chip is released from software standby and starts interrupt exception processing.
- Release due to a reset on the RES# pin reset
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the LSI 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.
- 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.
- 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.
- 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.

11.6.3.3 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 software standby mode is canceled 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, software standby mode is canceled 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 (ICUb).

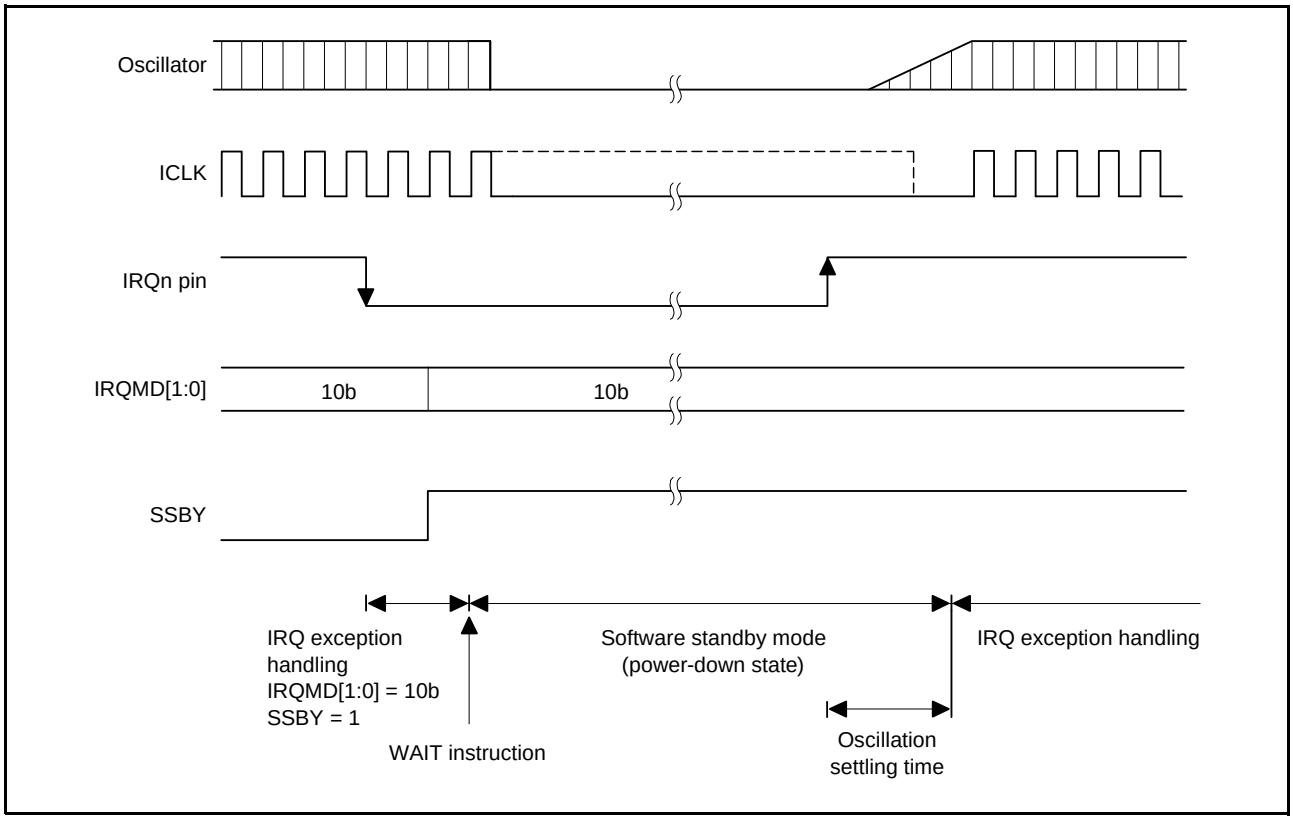


Figure 11.2 Example of Software Standby Mode Application

11.6.4 Deep Software Standby Mode

11.6.4.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode*1 is made. At this time, when the DPSBYCR.DPSBY bit is set to 1, a transition to deep software standby mode is made. On deep software standby mode, the CPU, internal peripheral modules (except for parts of the SCL0-DS and SDA0-DS), RAM1*2, and all functions of the oscillators are stopped; furthermore, since the internal supply of power for these modules is stopped, power consumption is markedly reduced. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for parts of the SCL0-DS and SDA0-DS) become undefined. All data in the RAM1*2 become undefined, regardless of the setting of the DPSBYCR.DEEPCUT[1:0] bits.
Data in the RAM0*2 are preserved if the setting of the DEEPCUT[1:0] bits is 00b. If the setting of the DEEPCUT[1:0] bits is 01b, the internal supply of power to the RAM0*2 is cut off, reducing power consumption. Data in the RAM0*2 become undefined at this time. If the setting of the DEEPCUT[1:0] bits is 11b, the internal supply of power to the RAM0*2 is cut off, the LVD is stopped, and the low-power-consumption function of the power-on reset circuit is enabled, so power consumption is further reduced. For details, see section 41, Electrical Characteristics.
When the WDT is in use, since the oscillators and power supply to the WDT are stopped by the transition to deep software standby mode, counting also stops.
Power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep 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 but not to deep software standby mode 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 but not to deep software standby mode while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
When the voltage monitoring 1 reset function (LVD1CR0.LVD1RI = 1) or voltage monitoring 2 reset function (LVD2CR0.LVD2RI = 1) is selected for the voltage detection circuit, a transition to deep software standby mode cannot be made, but to software standby.
The I/O port states remain unchanged from software standby mode.

Note 1. Conditions on the DTC, DMAC, and IWDT for transition to software standby mode should be met before the WAIT instruction is executed. For details, see section 11.6.3, Software Standby Mode.
Note 2. The RAM address space is divided into the RAM0 area and RAM1 area. For the RAM address space, see Table 11.2.

11.6.4.2 Canceling Deep Software Standby Mode

Release from deep software standby mode is initiated by any of the external pin interrupt source pins (the NMI, IRQ0-DS to IRQ12-DS, SCL0-DS, or SDA0-DS), peripheral interrupts (the voltage monitoring 1 and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage monitoring 0 reset.

(1) Canceling by an external interrupt pin or internal interrupt signal
Cancellation of deep software standby mode is controlled by DPSIERn (n = 0, 2) and DPSIFRn (n = 0, 2). When a deep software standby canceling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. At this time, if the canceling source is enabled in DPSIERn, deep software standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0, 2). The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ12-DS, SCL0-DS, SDA0-DS, voltage monitoring 1, and voltage monitoring 2 interrupts.
When a deep software standby mode canceling source is generated, the internal power supply and LOCO clock oscillation begin, and then the internal reset (deep software standby reset) is generated for the entire LSI.
A stable LOCO clock is then supplied to the entire LSI and deep software standby reset is canceled. At the same time, deep software standby mode is canceled and the reset exception handling starts.
When deep software standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Release due to a reset on the RES# pin
Deep software standby mode is canceled when the low level is applied to the RES# pin.
At this time, the RES# pin should be held low according to the specifications described in section 41, Electrical Characteristics. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from deep 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 0 reset
Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring 0 reset.

11.6.4.3 Pin States when Deep Software Standby Mode is Canceled

In deep software standby mode, the I/O ports retain the same states from software standby mode. The inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled. Upon cancellation of deep software standby mode, the reset exception handling starts immediately. The following shows the states of I/O ports at this time.
Whether to initialize the I/O ports or to retain the I/O port states at the time of software standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
  I/O ports are initialized by an internal reset generated when deep software standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1
  Although the inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled, I/O ports retain their states from software standby mode regardless of the LSI internal state. At this time, the I/O port states remain unchanged from software standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the LSI operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by an internal reset generated when deep software standby mode is canceled.

11.6.4.4 Example of Deep Software Standby Mode Application

Figure 11.3 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and deep software standby mode is canceled at the rising edge of the IRQn-DS pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). Then, after the DPSIEGR0.DIRQnEG (n = 0 to 7) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WAIT instruction is executed. Thus a transition to deep software standby mode is made.
After that, deep software standby mode is canceled at the rising edge of the IRQ-DS pin.

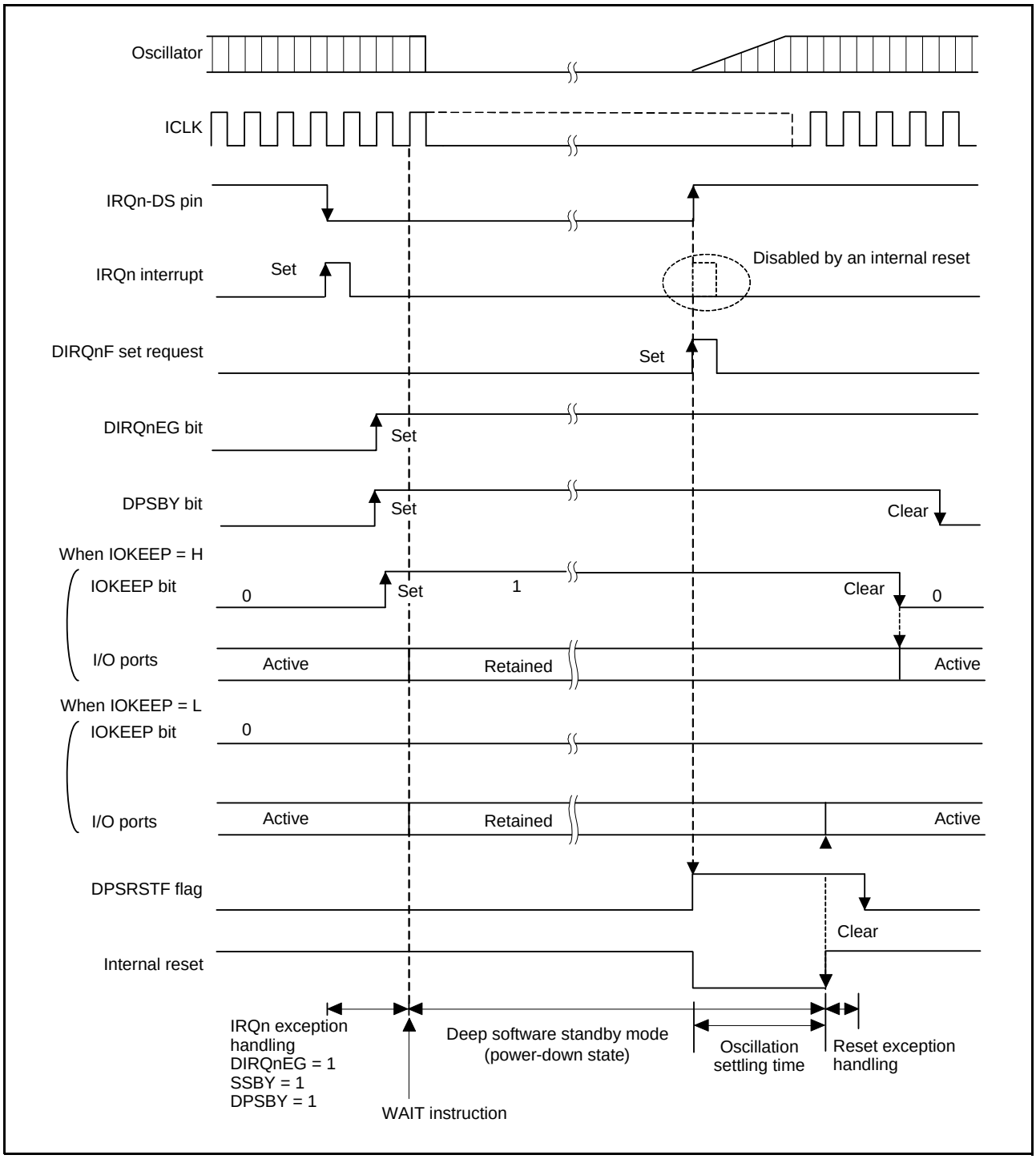


Figure 11.3 Example of Deep Software Standby Mode Application

11.6.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.4 shows an example of a flowchart to use deep software standby mode.
In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES# pin or by the cancellation of deep software standby mode.
In the case of a reset by the RES# pin, a transition to deep software standby mode is made after the required register settings have been made.
In the case of a reset by the cancellation of deep software standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings have been made.

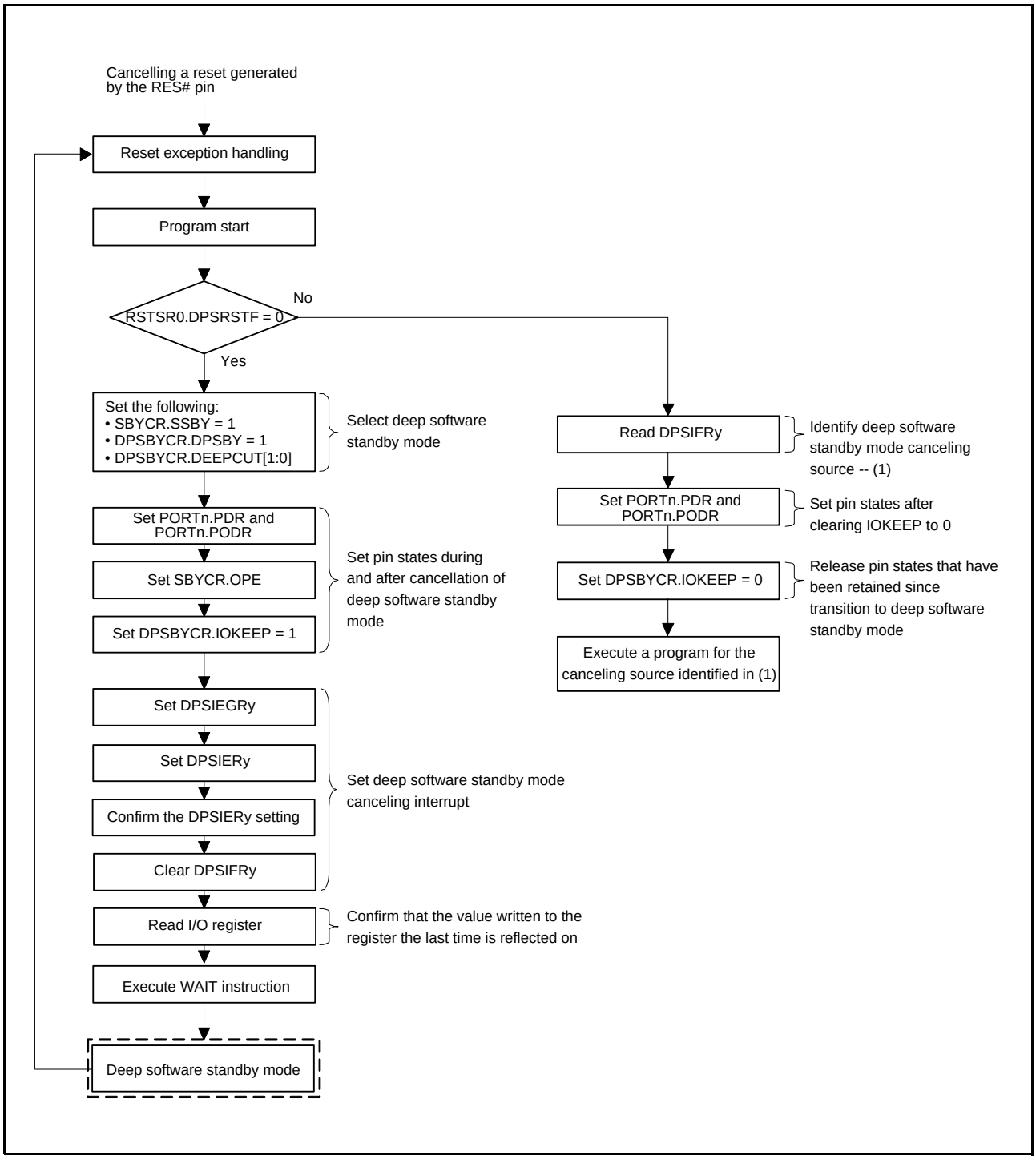


Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode. Therefore, the supply current is not reduced while output signals are held high.

11.7.2 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, clear 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 (DMACA) and section 18, Data Transfer Controller (DTCa).

11.7.3 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module-stop state. Therefore, if the module-stop state is made 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.

11.7.4 Write Access to MSTPCRA, MSTPCRB, and MSTPCRC

Write accesses to MSTPCRA, MSTPCRB, and MSTPCRC should be made only by the CPU.

11.7.5 Input Buffer Control by DIRQnE Bit (n = 0 to 7)

Setting the DPSIER0.DIRQnE (n = 0 to 7) bit to 1 enables the input buffer of the IRQ0-DS to IRQ7-DS pins. Therefore, note that, although inputs to these pins are sent to the DPSIFR0.DIRQnF (n = 0 to 7) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

11.7.6 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. 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, execute the WAIT instruction after confirming that the last write to the register has completed.

11.7.7 Rewrite the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMACA 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.

11.7.8 Canceling All-Module Clock Stop Mode

If the ICLK is set so as to be slower than the PCLKB, a TMR interrupt cannot be used to cancel all-module clock stop mode. To use the TMR interrupt as the all-module clock stop mode cancelling source, change the ICLK so as to be faster than the PCLKB before all-module clock stop mode is entered.

11.7.9 Points for Caution on Return from Software Standby

When interrupts IRQ0 to IRQ12 are not set as triggers for release from software standby, the corresponding input buffers become invalid on software standby, so the input signal within the LSI is fixed to the high level. Accordingly, depending on the state of the pin, a transition to software standby may set the interrupt status flag (ICU.IRi.IR) to 1.
When placing the chip on software standby, execute the WAIT instruction after setting the IERi.IENj bits to prohibit whichever of interrupts IRQ0 to IRQ12 are not to be used as triggers for release from standby. Moreover, after return from software standby, clear the interrupt status flags.

11.7.10 Point for Caution when Shifting from Low-Speed Operation Mode to Software Standby Mode

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

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, SCKCR3, PLLCR, PLLCR2, BCKCR, MOSCCR, LOCOCR, ILOCOCR, OSTDCR, OSTDSR
PRC1	• Registers related to the operating modes: SYSCR0, SYSCR1 • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR, RSTCKCR, MOSCWTCR, PLLWTCR, DPSBYCR, DPSIER0, DPSIER2, DPSIFR0, DPSIFR2, DPSIEGR0, DPSIEGR2 • Registers related to clock generation circuit: MOFCR • 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	Function	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, 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. Write data is not retained.

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 RX 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.

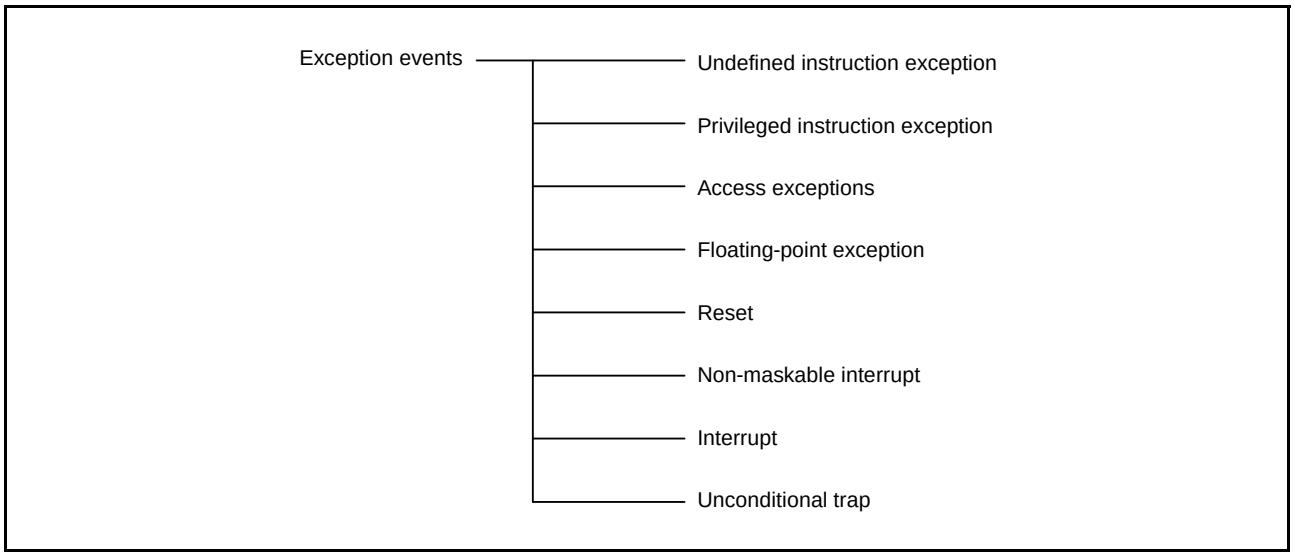


Figure 13.1 Types of Exception Events

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 Floating-Point Exception

Floating-point exceptions include the five exception events (overflow, underflow, inexact, division-by-zero, and invalid operation) specified in the IEEE754 standard and another floating-point exception that is generated on detection of unimplemented processing. The exception handling of floating-point exceptions is prohibited when the EX, EU, EZ, EO, or EV bit in FPSW is 0.

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.

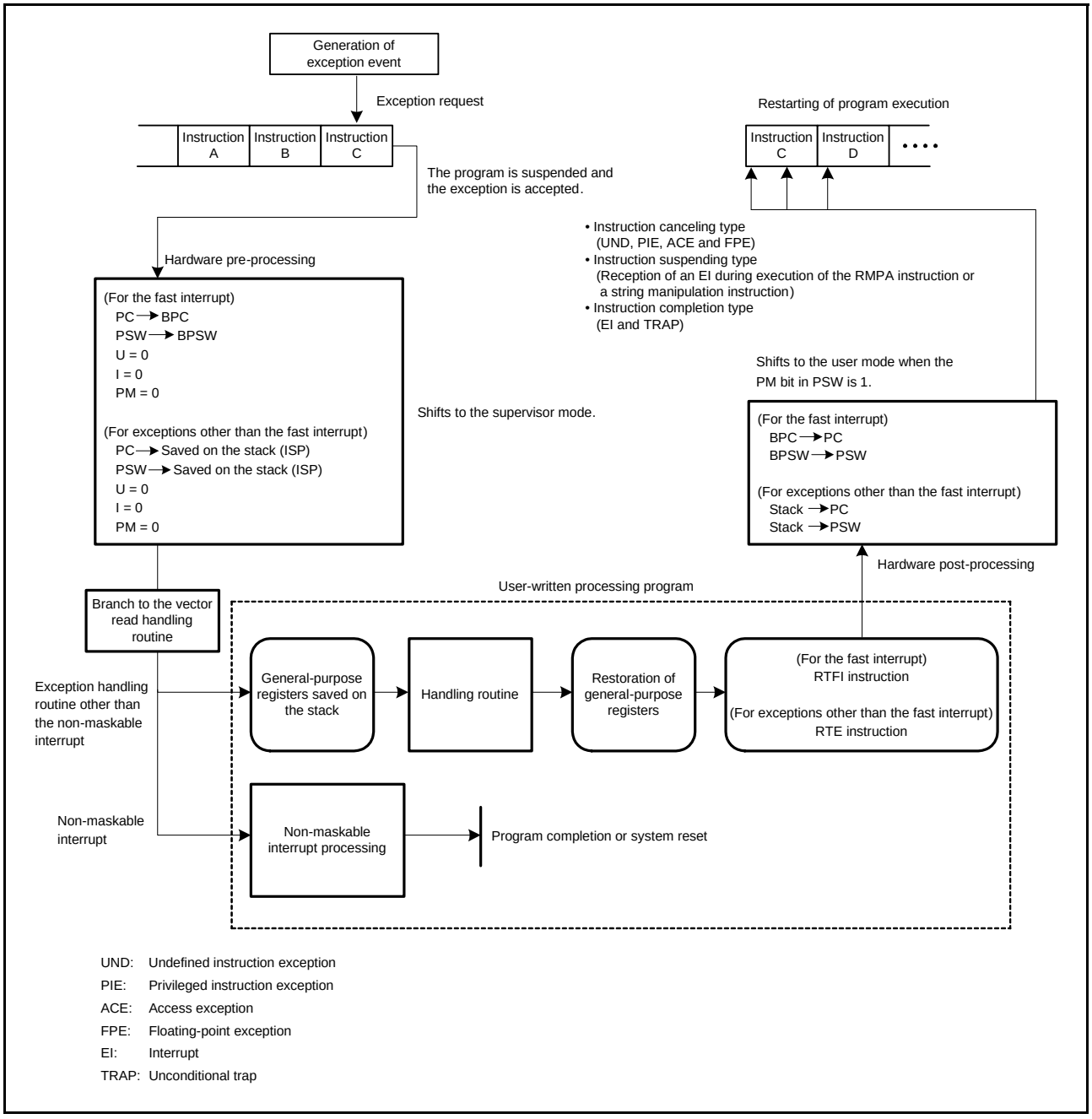


Figure 13.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RX 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 RX CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.

General purpose registers and control registers other than the PC and PSW that are to be used within the exception

handling routine must be saved on the stack by a user program at the start of the exception handling routine.
On completion of processing by an exception handling routine, registers saved on the stack 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 RX CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

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
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.

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		Fixed vector table	Stack
Privileged instruction exception		Fixed vector table	Stack
Access exception		Fixed vector table	Stack
Floating-point exception		Fixed vector table	Stack
Reset		Fixed vector table	Nowhere
Non-maskable interrupt		Fixed vector table	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Relocatable vector table (INTB)	Stack
Unconditional trap		Relocatable 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 values in FPSW are not saved by hardware pre-processing. Therefore, if floating-point instructions are to be used within an exception handling routine, the user must save these values 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

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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFDCh.
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 in 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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFD0h.
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 in 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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFFFFD4h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.4 Floating-Point Exception

1. The value in 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 cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFE4h.
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 cleared 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 address FFFF FFF8h.
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 cleared 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 relocatable 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 in 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 cleared 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 relocatable vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the relocatable 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 restores 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
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Return is impossible
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	Floating-point exception

14. Interrupt Controller (ICUb)

14.1 Overview

The interrupt controller receives interrupt signals from peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC and DMAC.

Table 14.1 lists the specifications of the interrupt controller, and Figure 14.1 shows a block diagram of the interrupt controller.

Table 14.1 Specifications of Interrupt Controller

Item		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules.
	External pin interrupts	• Interrupts from pins IRQ0 to IRQ12 • Number of sources: 13 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source. • Digital filter function: Supported
	Software interrupt	• Interrupt generated by writing to a register • One interrupt source
	Event link interrupt	The ELSR18I or ELSR19I interrupt is generated by an ELC event
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC/DMAC control	The DTC and DMAC can be activated by interrupt sources.*1
Non-maskable interrupts	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter function: Supported
	Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
	WDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	IWDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	Voltage monitoring 1 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1)
	Voltage monitoring 2 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 2 (LVD2)
Return from power-down modes		• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • All-module clock stop mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ12 interrupts, TMR interrupts. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ12 interrupts.

Note 1. For the DTC and DMAC activation sources, refer to Table 14.3, Interrupt Vector Table.

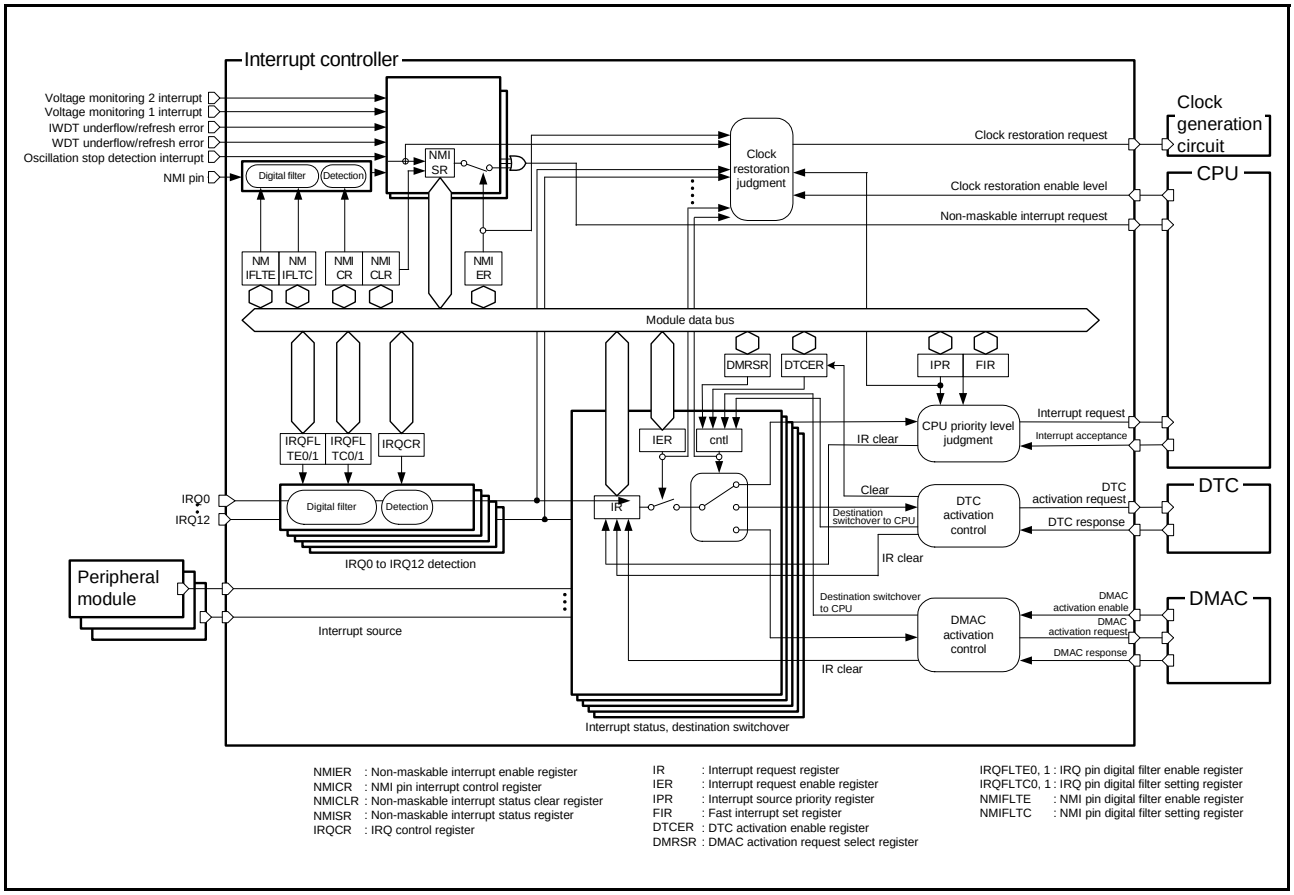


Figure 14.1 Block Diagram of Interrupt Controller

Table 14.2 lists the input/output pins of the interrupt controller.

Table 14.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ12	Input	External interrupt request pins

14.2 Register Descriptions

14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): 0008 7010h to 0008 70FDh

	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	These bits are read as 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

This bit is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQi pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits (i = 0 to 12). For detection of the various interrupt sources, see Table 14.3, Interrupt Vector Table.

(1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC or DMAC.

(2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing condition]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high.
Do not write 0 or 1 to the IR flag while level detection is selected.

14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): 0008 7202h to 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: Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

IENj Bits (Interrupt Request Enable j) (j = 7 to 0)

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag. Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 14.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bits, see Table 14.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 14.4.3, Selecting Interrupt Request Destinations.

14.2.3 Interrupt Source Priority Register n (IPRn) (n = 000 to 250)

Address(es): 0008 7300h to 0008 73FAh

	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	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the interrupt is specified as a fast interrupt, it can be issued even if the priority level is level 0.

For the correspondence between interrupt sources and IPRn registers, see Table 14.3, Interrupt Vector Table.

IPR[3:0] Bits (Interrupt Priority Level Select)

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect activation requests to the DTC and DMAC.

The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (IERm.IENj bit = 0).

14.2.4 Fast Interrupt Set Register (FIR)

Address(es): 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	Specify the vector number of an interrupt source to be a fast interrupt.	R/W
b14 to b8	—	Reserved	These bits are read as 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 fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC or DMAC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0).

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits specify the vector number of an interrupt source that uses the fast interrupt function.

FIEN Bit (Fast Interrupt Enable)

This bit enables the fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt.
When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register. When using the fast interrupt for returning from the software standby mode, see section 14.6.3, Return from Software Standby Mode.
If the setting of the IERm.IENj (m = 02h to 1Fh, j = 7 to 0) bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 14.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, see section 13, Exception Handling, and section 14.4.6, Fast Interrupt.

14.2.5 Software Interrupt Activation Register (SWINTR)

Address(es): 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 Activation	This bit is read as 0. Writing 1 issues a software interrupt request. Writing 0 to this bit has no effect.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written.

SWINT Bit (Software Interrupt Activation)

When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when the DTC activation enable register 027 (DTCER027) is set to 0, an interrupt to the CPU is generated.

If 1 is written to the SWINT bit when the DTC activation enable register 027 (DTCER027) is set to 1, a DTC activation request is issued.

14.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): 0008 711Bh to 0008 71FCh

	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 Activation Enable	0: DTC activation is disabled 1: DTC activation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

An interrupt source that has been selected as a source for DMAC activation should not be specified as a source for DTC activation. See Table 14.3, Interrupt Vector Table, for the interrupt sources that are selectable as sources for DTC activation.

DTCE Bit (DTC Activation Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the source for the DTC activation.

[Setting condition]

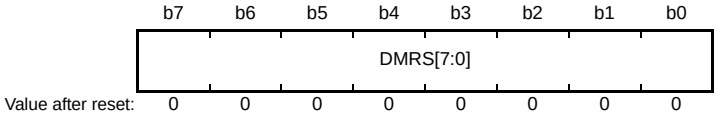
- When 1 is written to the DTCE bit

[Clearing conditions]

- When the specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- When 0 is written to the DTCE bit

14.2.7 DMAC Activation Request Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): DMRSR0 0008 7400h, DMRSR1 0008 7404h, DMRSR2 0008 7408h, DMRSR3 0008 740Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DMRS[7:0]	DMAC Activation Source Select	These bits specify the vector number for the DMAC activation request.	R/W

To specify the same interrupt source for multiple DMRSRm registers is disabled. The interrupt source that has been selected for the DMRSRm activation should not be specified as the source for the DTC activation. Otherwise, the correct operation is not guaranteed.

DMRS[7:0] Bits (DMAC Activation Source Select)

The vector number of the interrupt source for DMAC activation is specified in 8 bits. Do not set the vector numbers that are not assigned for the DMAC activation.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

Write to the DMRSRm register while the DMA transfer enable bit of the DMA transfer enable register (DMACm.DMCNT.DTE) is cleared to 0.

14.2.8 IRQ Control Register i (IRQCRi) (i = 0 to 12)

Address(es): 0008 7500h to 0008 750Ch

	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	These bits are read as 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense 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	These bits are read as 0. The write value should be 0.	R/W

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

IRQMD[1:0] Bits (IRQ Detection Sense Select)

These bits select the detection sensing method of external pin interrupt sources IRQ0 to IRQ12.
For the external pin interrupt detection setting, see section 14.4.8, External Pin Interrupts.

14.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7510h

	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

FLTENi Bits (IRQi Digital Filter Enable) (i = 0 to 7)

These bits enable the digital filter used for the external pin interrupt sources IRQ0 to IRQ7.

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 IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.10 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): 0008 7511h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FLTENi Bits (IRQi Digital Filter Enable) (i = 8 to 12)

These bits enable the digital filter used for the external pin interrupt sources IRQ8 to IRQ12.

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.

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.11 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): 0008 7514h

	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

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ0 to IRQ7. The sampling clock cycle can be selected from among the PCLKB (every cycle), PCLKB/8 (once every eight cycles), PCLKB/32 (once every 32 cycles), and PCLKB/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.12 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): 0008 7516h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 8 to 12)

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ8 to IRQ12. The sampling clock cycle can be selected from among the PCLKB (every cycle), PCLKB/8 (once every eight cycles), PCLKB/32 (once every 32 cycles), and PCLKB/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.13 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

The NMISR register monitors the status of a non-maskable interrupt source. Writing to the NMISR register is ignored. The setting in the non-maskable interrupt enable register (NMIER) does not affect the status flags in NMISR.
Before the end of the non-maskable interrupt handler, read the NMISR register and confirm the generation status of other non-maskable interrupts. Be sure to confirm that all of the bits in the NMISR register are set to 0 before the end of the handler.

NMIST Flag (NMI Status Flag)

This flag indicates the NMI pin interrupt request.
The NMIST flag is read-only, and cleared by the NMICLR.NMICLR bit.
[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the oscillation stop detection interrupt request.
The OSTST flag is read-only, and cleared by the NMICLR.OSTCLR bit.
[Setting condition]

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

This flag indicates the WDT underflow/refresh error interrupt request.
The WDTST flag is read-only, and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

- When the WDT underflow/refresh error interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

This flag indicates the IWDT underflow/refresh error interrupt request.
The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.
[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 1 interrupt.
The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.
[Setting condition]

- When the voltage monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 2 interrupt.
The LVD2ST flag is read-only, and cleared by the NMICLR.LVD2CLR bit.
[Setting condition]

- When the voltage monitoring 2 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

14.2.14 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

This bit enables the WDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

This bit enables the voltage monitoring 1 interrupt.
A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.
Writing 0 to this bit is disabled.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

This bit enables the voltage monitoring 2 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

14.2.15 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): 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	This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect.	R/(W) *1
b1	OSTCLR	OST Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b2	WDTCLR	WDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.WDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b3	IWDTCLR	IWDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b4	LVD1CLR	LVD1 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b5	LVD2CLR	LVD2 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD2ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag.
This bit is read as 0.

OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag.
This bit is read as 0.

WDTCLR Bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag.
This bit is read as 0.

IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag.
This bit is read as 0.

LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag.
This bit is read as 0.

LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag.
This bit is read as 0.

14.2.16 NMI Pin Interrupt Control Register (NMICR)

Address(es): 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	These bits are read as 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	These bits are read as 0. The write value should be 0.	R/W

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

14.2.17 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): 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	These bits are read as 0. The write value should be 0.	R/W

NFLTEN Bit (NMI Digital Filter Enable)

This bit enables the digital filter used for the NMI pin interrupt.

The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

The NMI pin level is sampled at the sampling clock cycle specified with the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.18 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): 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	These bits are read as 0. The write value should be 0.	R/W

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

These bits select the cycle of the digital filter sampling clock for the NMI pin interrupt.

The sampling clock cycle can be selected from among the PCLKB (every cycle), PCLKB/8 (once every eight cycles), PCLKB/32 (once every 32 cycles), and PCLKB/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.7, Digital Filter.

14.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a four-byte vector address from the vector table.

14.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.
Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 14.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.
Table 14.3 lists details of the interrupt vectors. Details of the headings in Table 14.3 are listed below.

Item	Description
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
Form of interrupt detection	“Edge” or “level” as the method for detection of the interrupt
CPU interrupt	“○” in this column indicates usability as a CPU interrupt.
DTC activation	“○” in this column indicates usability as a request for DTC activation.
DMAC activation	“○” in this column indicates usability as a request for DMAC activation.
sstb return	“○” in this column indicates usability as a request for return from software-standby mode.
sacs return	“○” in this column indicates usability as a request for return from all-module clock-stop mode.
IER	Name of the interrupt request enable register (IER) and bit corresponding to the vector number
IPR	Name of the interrupt source priority register (IPR) corresponding to the interrupt source
DTCER	Name of the DTC activation enable register (DTCER) corresponding to the DTC activation source

Table 14.3 Interrupt Vector Table (1/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	1	0004h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	2	0008h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	3	000Ch	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	4	0010h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	5	0014h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	6	0018h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	7	001Ch	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	8	0020h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	9	0024h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	10	0028h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	11	002Ch	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	12	0030h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	13	0034h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	14	0038h	—	×	×	×	×	×	—	—	—
—	For an unconditional trap	15	003Ch	—	×	×	×	×	×	—	—	—
BSC	BUSERR	16	0040h	Level	○	×	×	×	×	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	×	×	×	×	×	—	—	—
—	Reserved	18	0048h	—	×	×	×	×	×	—	—	—
—	Reserved	19	004Ch	—	×	×	×	×	×	—	—	—
—	Reserved	20	0050h	—	×	×	×	×	×	—	—	—
FCU	FIFERR	21	0054h	Level	○	×	×	×	×	IER02.IEN5	IPR001	—
—	Reserved	22	0058h	—	×	×	×	×	×	—	—	—
FCU	FRDYI	23	005Ch	Edge	○	×	×	×	×	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	×	×	×	×	×	—	—	—
—	Reserved	25	0064h	—	×	×	×	×	×	—	—	—
—	Reserved	26	0068h	—	×	×	×	×	×	—	—	—
ICU	SWINT	27	006Ch	Edge	○	○	×	×	×	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	○	○	○	×	×	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	○	○	○	×	×	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	○	○	○	×	×	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	○	○	○	×	×	IER03.IEN7	IPR007	DTCER031
CAC	FERRF	32	0080h	Level	○	×	×	×	×	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	○	×	×	×	×	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	○	×	×	×	×	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	×	×	×	×	×	—	—	—
—	Reserved	36	0090h	—	×	×	×	×	×	—	—	—
—	Reserved	37	0094h	—	×	×	×	×	×	—	—	—
—	Reserved	38	0098h	—	×	×	×	×	×	—	—	—
—	Reserved	39	009Ch	—	×	×	×	×	×	—	—	—
—	Reserved	40	00A0h	—	×	×	×	×	×	—	—	—
—	Reserved	41	00A4h	—	×	×	×	×	×	—	—	—
—	Reserved	42	00A8h	—	×	×	×	×	×	—	—	—
—	Reserved	43	00ACh	—	×	×	×	×	×	—	—	—

Table 14.3 Interrupt Vector Table (2/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
RSPI0	SPEI0	44	00B0h	Level	○	×	×	×	×	IER05.IEN4	IPR044	—
	SPRI0	45	00B4h	Edge	○	○	○	×	×	IER05.IEN5		DTCER045
	SPTI0	46	00B8h	Edge	○	○	○	×	×	IER05.IEN6		DTCER046
	SPII0	47	00BCh	Level	○	×	×	×	×	IER05.IEN7		—
RSPI1	SPEI1	48	00C0h	Level	○	—	—	—	—	IER06.IEN0	IPR048	—
	SPRI1	49	00C4h	Edge	○	○	○	—	—	IER06.IEN1		DTCER049
	SPTI1	50	00C8h	Edge	○	○	○	—	—	IER06.IEN2		DTCER050
	SPII1	51	00CCh	Level	○	—	—	—	—	IER06.IEN3		—
—	Reserved	52	00D0h	—	×	×	×	×	×	—	—	—
—	Reserved	53	00D4h	—	×	×	×	×	×	—	—	—
—	Reserved	54	00D8h	—	×	×	×	×	×	—	—	—
—	Reserved	55	00DCh	—	×	×	×	×	×	—	—	—
—	Reserved	56	00E0h	—	×	×	×	×	×	—	—	—
DOC	DOPCF	57	00E4h	Level	○	×	×	×	×	IER07.IEN1	IPR057	—
—	Reserved	58	00E8h	—	×	×	×	×	×	—	—	—
—	Reserved	59	00ECh	—	×	×	×	×	×	—	—	—
—	Reserved	60	00F0h	—	×	×	×	×	×	—	—	—
—	Reserved	61	00F4h	—	×	×	×	×	×	—	—	—
—	Reserved	62	00F8h	—	×	×	×	×	×	—	—	—
—	Reserved	63	00FCh	—	×	×	×	×	×	—	—	—
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
CEC	INTDAA	77	0134h	Edge	○	○	×	○	○	IER09.IEN5	IPR077	DTCER077
	INTCEA	78	0138h	Edge	○	○	×	○	○	IER09.IEN6	IPR078	DTCER078
	INTERRA	79	013Ch	Edge	○	○	×	○	○	IER09.IEN7	IPR079	DTCER079
—	Reserved	80	0140h	—	×	×	×	×	×	—	—	—
—	Reserved	81	0144h	—	×	×	×	×	×	—	—	—
—	Reserved	82	0148h	—	×	×	×	×	×	—	—	—
—	Reserved	83	014Ch	—	×	×	×	×	×	—	—	—
—	Reserved	84	0150h	—	×	×	×	×	×	—	—	—
—	Reserved	85	0154h	—	×	×	×	×	×	—	—	—
—	Reserved	86	0158h	—	×	×	×	×	×	—	—	—
—	Reserved	87	015Ch	—	×	×	×	×	×	—	—	—
—	Reserved	88	0160h	—	×	×	×	×	×	—	—	—
—	Reserved	89	0164h	—	×	×	×	×	×	—	—	—
—	Reserved	90	0168h	—	×	×	×	×	×	—	—	—

Table 14.3 Interrupt Vector Table (3/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
—	Reserved	91	016Ch	—	×	×	×	×	×	—	—	—
—	Reserved	92	0170h	—	×	×	×	×	×	—	—	—
—	Reserved	93	0174h	—	×	×	×	×	×	—	—	—
RCR0	RCRI0	94	0178h	Edge	○	—	—	○	○	IER0B.IEN6	IPR094	—
RCR1	RCRI1	95	017Ch	Edge	○	—	—	○	○	IER0B.IEN7	IPR095	—
—	Reserved	96	0180h	—	×	×	×	×	×	—	—	—
—	Reserved	97	0184h	—	×	×	×	×	×	—	—	—
RIIC1	EEI1	98	0188h	Level	○	—	—	—	—	IER0C.IEN2	IPR098	—
	RXI1	99	018Ch	Edge	○	○	○	—	—	IER0C.IEN3	IPR099	DTCER099
	TXI1	100	0190h	Edge	○	○	○	—	—	IER0C.IEN4	IPR100	DTCER100
	TEI1	101	0194h	Level	○	—	—	—	—	IER0C.IEN5	IPR101	—
S12AD	S12ADI0	102	0198h	Edge	○	○	○	×	×	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	○	○	○	×	×	IER0C.IEN7	IPR103	DTCER103
—	Reserved	104	01A0h	—	×	×	×	×	×	—	—	—
—	Reserved	105	01A4h	—	×	×	×	×	×	—	—	—
ELC	ELSR18I	106	01A8h	Edge	○	○	○	×	×	IER0D.IEN2	IPR106	DTCER106
	ELSR19I	107	01ACh	Edge	○	○	○	×	×	IER0D.IEN3	IPR107	DTCER107
—	Reserved	108	01B0h	—	×	×	×	×	×	—	—	—
—	Reserved	109	01B4h	—	×	×	×	×	×	—	—	—
—	Reserved	110	01B8h	—	×	×	×	×	×	—	—	—
CEC	INTDA	111	01BCh	Edge	○	○	○	—	—	IER0D.IEN7	IPR111	DTCER111
	INTCE	112	01C0h	Edge	○	○	○	—	—	IER0E.IEN0		DTCER112
	INTERR	113	01C4h	Edge	○	—	—	—	—	IER0E.IEN1		—
MTU0	TGIA0	114	01C8h	Edge	○	○	○	×	×	IER0E.IEN2	IPR114	DTCER114
	TGIB0	115	01CCh	Edge	○	○	×	×	×	IER0E.IEN3		DTCER115
	TGIC0	116	01D0h	Edge	○	○	×	×	×	IER0E.IEN4		DTCER116
	TGID0	117	01D4h	Edge	○	○	×	×	×	IER0E.IEN5		DTCER117
	TCIV0	118	01D8h	Edge	○	×	×	×	×	IER0E.IEN6	IPR118	—
	TGIE0	119	01DCh	Edge	○	×	×	×	×	IER0E.IEN7		—
	TGIF0	120	01E0h	Edge	○	×	×	×	×	IER0F.IEN0		—
MTU1	TGIA1	121	01E4h	Edge	○	○	○	×	×	IER0F.IEN1	IPR121	DTCER121
	TGIB1	122	01E8h	Edge	○	○	×	×	×	IER0F.IEN2		DTCER122
	TCIV1	123	01ECh	Edge	○	×	×	×	×	IER0F.IEN3	IPR123	—
	TCIU1	124	01F0h	Edge	○	×	×	×	×	IER0F.IEN4		—
MTU2	TGIA2	125	01F4h	Edge	○	○	○	×	×	IER0F.IEN5	IPR125	DTCER125
	TGIB2	126	01F8h	Edge	○	○	×	×	×	IER0F.IEN6		DTCER126
	TCIV2	127	01FCh	Edge	○	×	×	×	×	IER0F.IEN7	IPR127	—
	TCIU2	128	0200h	Edge	○	×	×	×	×	IER10.IEN0		—
MTU3	TGIA3	129	0204h	Edge	○	○	○	×	×	IER10.IEN1	IPR129	DTCER129
	TGIB3	130	0208h	Edge	○	○	×	×	×	IER10.IEN2		DTCER130
	TGIC3	131	020Ch	Edge	○	○	×	×	×	IER10.IEN3		DTCER131
	TGID3	132	0210h	Edge	○	○	×	×	×	IER10.IEN4		DTCER132
	TCIV3	133	0214h	Edge	○	×	×	×	×	IER10.IEN5	IPR133	—

Table 14.3 Interrupt Vector Table (4/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
MTU4	TGIA4	134	0218h	Edge	○	○	○	×	×	IER10.IEN6	IPR134	DTCER134
	TGIB4	135	021Ch	Edge	○	○	×	×	×	IER10.IEN7		DTCER135
	TGIC4	136	0220h	Edge	○	○	×	×	×	IER11.IEN0		DTCER136
	TGID4	137	0224h	Edge	○	○	×	×	×	IER11.IEN1		DTCER137
	TCIV4	138	0228h	Edge	○	○	×	×	×	IER11.IEN2	IPR138	DTCER138
MTU5	TGIU5	139	022Ch	Edge	○	○	×	×	×	IER11.IEN3	IPR139	DTCER139
	TGIV5	140	0230h	Edge	○	○	×	×	×	IER11.IEN4		DTCER140
	TGIW5	141	0234h	Edge	○	○	×	×	×	IER11.IEN5		DTCER141
TPU0	TGI0A	142	0238h	Edge	○	○	○	×	×	IER11.IEN6	IPR142	DTCER142
	TGI0B	143	023Ch	Edge	○	○	×	×	×	IER11.IEN7		DTCER143
	TGI0C	144	0240h	Edge	○	○	×	×	×	IER12.IEN0		DTCER144
	TGI0D	145	0244h	Edge	○	○	×	×	×	IER12.IEN1		DTCER145
	TCI0V	146	0248h	Edge	○	×	×	×	×	IER12.IEN2	IPR146	—
TPU1	TGI1A	147	024Ch	Edge	○	○	○	×	×	IER12.IEN3	IPR147	DTCER147
	TGI1B	148	0250h	Edge	○	○	×	×	×	IER12.IEN4		DTCER148
	TCI1V	149	0254h	Edge	○	×	×	×	×	IER12.IEN5	IPR149	—
	TCI1U	150	0258h	Edge	○	×	×	×	×	IER12.IEN6		—
TPU2	TGI2A	151	025Ch	Edge	○	○	○	×	×	IER12.IEN7	IPR151	DTCER151
	TGI2B	152	0260h	Edge	○	○	×	×	×	IER13.IEN0		DTCER152
	TCI2V	153	0264h	Edge	○	×	×	×	×	IER13.IEN1	IPR153	—
	TCI2U	154	0268h	Edge	○	×	×	×	×	IER13.IEN2		—
TPU3	TGI3A	155	026Ch	Edge	○	○	○	×	×	IER13.IEN3	IPR155	DTCER155
	TGI3B	156	0270h	Edge	○	○	×	×	×	IER13.IEN4		DTCER156
	TGI3C	157	0274h	Edge	○	○	×	×	×	IER13.IEN5		DTCER157
	TGI3D	158	0278h	Edge	○	○	×	×	×	IER13.IEN6		DTCER158
	TCI3V	159	027Ch	Edge	○	×	×	×	×	IER13.IEN7	IPR159	—
TPU4	TGI4A	160	0280h	Edge	○	○	○	×	×	IER14.IEN0	IPR160	DTCER160
	TGI4B	161	0284h	Edge	○	○	×	×	×	IER14.IEN1		DTCER161
	TCI4V	162	0288h	Edge	○	×	×	×	×	IER14.IEN2	IPR162	—
	TCI4U	163	028Ch	Edge	○	×	×	×	×	IER14.IEN3		—
TPU5	TGI5A	164	0290h	Edge	○	○	○	×	×	IER14.IEN4	IPR164	DTCER164
	TGI5B	165	0294h	Edge	○	○	×	×	×	IER14.IEN5		DTCER165
	TCI5V	166	0298h	Edge	○	×	×	×	×	IER14.IEN6	IPR166	—
	TCI5U	167	029Ch	Edge	○	×	×	×	×	IER14.IEN7		—
—	Reserved	168	02A0h	—	×	×	×	×	×	—	—	—
—	Reserved	169	02A4h	—	×	×	×	×	×	—	—	—
POE	OEI1	170	02A8h	Level	○	×	×	×	×	IER15.IEN2	IPR170	—
	OEI2	171	02ACh	Level	○	×	×	×	×	IER15.IEN3	IPR171	—
—	Reserved	172	02B0h	—	×	×	×	×	×	—	—	—
—	Reserved	173	02B4h	—	×	×	×	×	×	—	—	—
TMR0	CMIA0	174	02B8h	Edge	○	○	×	×	○	IER15.IEN6	IPR174	DTCER174
	CMIB0	175	02BCh	Edge	○	○	×	×	○	IER15.IEN7		DTCER175
	OVI0	176	02C0h	Edge	○	×	×	×	○	IER16.IEN0		—
TMR1	CMIA1	177	02C4h	Edge	○	○	×	×	○	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	○	○	×	×	○	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	○	×	×	×	○	IER16.IEN3		—

Table 14.3 Interrupt Vector Table (5/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
TMR2	CMIA2	180	02D0h	Edge	○	○	×	×	○	IER16.IEN4	IPR180	DTCER180
	CMIB2	181	02D4h	Edge	○	○	×	×	○	IER16.IEN5		DTCER181
	OVI2	182	02D8h	Edge	○	×	×	×	○	IER16.IEN6		—
TMR3	CMIA3	183	02DCh	Edge	○	○	×	×	○	IER16.IEN7	IPR183	DTCER183
	CMIB3	184	02E0h	Edge	○	○	×	×	○	IER17.IEN0		DTCER184
	OVI3	185	02E4h	Edge	○	×	×	×	○	IER17.IEN1		—
SCI2	ERI2	186	02E8h	Level	○	×	×	×	×	IER17.IEN2	IPR186	—
	RXI2	187	02ECh	Edge	○	○	○	×	×	IER17.IEN3		DTCER187
	TXI2	188	02F0h	Edge	○	○	○	×	×	IER17.IEN4		DTCER188
	TEI2	189	02F4h	Level	○	×	×	×	×	IER17.IEN5		—
SCI3	ERI3	190	02F8h	Level	○	×	×	×	×	IER17.IEN6	IPR190	—
	RXI3	191	02FCh	Edge	○	○	○	×	×	IER17.IEN7		DTCER191
	TXI3	192	0300h	Edge	○	○	○	×	×	IER18.IEN0		DTCER192
	TEI3	193	0304h	Level	○	×	×	×	×	IER18.IEN1		—
SCI4	ERI4	194	0308h	Level	○	×	×	×	×	IER18.IEN2	IPR194	—
	RXI4	195	030Ch	Edge	○	○	○	×	×	IER18.IEN3		DTCER195
	TXI4	196	0310h	Edge	○	○	○	×	×	IER18.IEN4		DTCER196
	TEI4	197	0314h	Level	○	×	×	×	×	IER18.IEN5		—
DMAC	DMAC0I	198	0318h	Edge	○	○	×	×	×	IER18.IEN6	IPR198	DTCER198
	DMAC1I	199	031Ch	Edge	○	○	×	×	×	IER18.IEN7	IPR199	DTCER199
	DMAC2I	200	0320h	Edge	○	○	×	×	×	IER19.IEN0	IPR200	DTCER200
	DMAC3I	201	0324h	Edge	○	○	×	×	×	IER19.IEN1	IPR201	DTCER201
RIIC3	EEI3	202	0328h	Level	○	—	—	—	—	IER19.IEN2	IPR202	—
	RXI3	203	032Ch	Edge	○	○	○	—	—	IER19.IEN3	IPR203	DTCER203
	TXI3	204	0330h	Edge	○	○	○	—	—	IER19.IEN4	IPR204	DTCER204
	TEI3	205	0334h	Level	○	—	—	—	—	IER19.IEN5	IPR205	—
SCI7	ERI7	206	0338h	Level	○	×	×	×	×	IER19.IEN6	IPR206	—
	RXI7	207	033Ch	Edge	○	○	○	×	×	IER19.IEN7		DTCER207
	TXI7	208	0340h	Edge	○	○	○	×	×	IER1A.IEN0		DTCER208
	TEI7	209	0344h	Level	○	×	×	×	×	IER1A.IEN1		—
SCI10	ERI10	210	0348h	Level	○	×	×	×	×	IER1A.IEN2	IPR210	—
	RXI10	211	034Ch	Edge	○	○	○	×	×	IER1A.IEN3		DTCER211
	TXI10	212	0350h	Edge	○	○	○	×	×	IER1A.IEN4		DTCER212
	TEI10	213	0354h	Level	○	×	×	×	×	IER1A.IEN5		—
SCI0	ERI0	214	0358h	Level	○	×	×	×	×	IER1A.IEN6	IPR214	—
	RXI0	215	035Ch	Edge	○	○	○	×	×	IER1A.IEN7		DTCER215
	TXI0	216	0360h	Edge	○	○	○	×	×	IER1B.IEN0		DTCER216
	TEI0	217	0364h	Level	○	×	×	×	×	IER1B.IEN1		—
SCI1	ERI1	218	0368h	Level	○	×	×	×	×	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	○	○	○	×	×	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	○	○	○	×	×	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	○	×	×	×	×	IER1B.IEN5		—
SCI5	ERI5	222	0378h	Level	○	×	×	×	×	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	○	○	○	×	×	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	○	○	○	×	×	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	○	×	×	×	×	IER1C.IEN1		—

Table 14.3 Interrupt Vector Table (6/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	DMAC	sstb Return	sacs Return	IER	IPR	DTCER
SCI6	ERI6	226	0388h	Level	○	×	×	×	×	IER1C.IEN2	IPR226	—
	RXI6	227	038Ch	Edge	○	○	○	×	×	IER1C.IEN3		DTCER227
	TXI6	228	0390h	Edge	○	○	○	×	×	IER1C.IEN4		DTCER228
	TEI6	229	0394h	Level	○	×	×	×	×	IER1C.IEN5		—
SCI8	ERI8	230	0398h	Level	○	×	×	×	×	IER1C.IEN6	IPR230	—
	RXI8	231	039Ch	Edge	○	○	○	×	×	IER1C.IEN7		DTCER231
	TXI8	232	03A0h	Edge	○	○	○	×	×	IER1D.IEN0		DTCER232
	TEI8	233	03A4h	Level	○	×	×	×	×	IER1D.IEN1		—
SCI9	ERI9	234	03A8h	Level	○	×	×	×	×	IER1D.IEN2	IPR234	—
	RXI9	235	03ACh	Edge	○	○	○	×	×	IER1D.IEN3		DTCER235
	TXI9	236	03B0h	Edge	○	○	○	×	×	IER1D.IEN4		DTCER236
	TEI9	237	03B4h	Level	○	×	×	×	×	IER1D.IEN5		—
SCI12	ERI12	238	03B8h	Level	○	×	×	×	×	IER1D.IEN6	IPR238	—
	RXI12	239	03BCh	Edge	○	○	○	×	×	IER1D.IEN7		DTCER239
	TXI12	240	03C0h	Edge	○	○	○	×	×	IER1E.IEN0		DTCER240
	TEI12	241	03C4h	Level	○	×	×	×	×	IER1E.IEN1		—
	SCIX0	242	03C8h	Level	○	×	×	×	×	IER1E.IEN2	IPR242	—
	SCIX1	243	03CCh	Level	○	×	×	×	×	IER1E.IEN3	IPR243	—
	SCIX2	244	03D0h	Level	○	×	×	×	×	IER1E.IEN4	IPR244	—
	SCIX3	245	03D4h	Level	○	×	×	×	×	IER1E.IEN5	IPR245	—
RIIC0	EEI0	246	03D8h	Level	○	×	×	×	×	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	○	○	○	×	×	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	○	○	○	×	×	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	○	×	×	×	×	IER1F.IEN1	IPR249	—
SCI11	ERI11	250	03E8h	Level	○	×	×	×	×	IER1F.IEN2	IPR250	—
	RXI11	251	03ECh	Edge	○	○	○	×	×	IER1F.IEN3		DTCER251
	TXI11	252	03F0h	Edge	○	○	○	×	×	IER1F.IEN4		DTCER252
	TEI11	253	03F4h	Level	○	×	×	×	×	IER1F.IEN5		—
—	Reserved	254	03F8h	—	×	×	×	×	×	—	—	—
—	Reserved	255	03FCh	—	×	×	×	×	×	—	—	—

Note 1. An interrupt source with a smaller vector number takes precedence.

14.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

14.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table is at FFFF FFF8h.

14.4 Interrupt Operation

The interrupt controller performs the following processing.

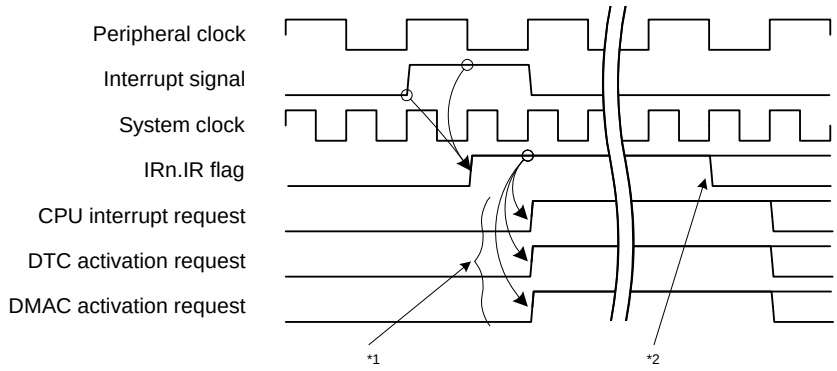
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation, or DMAC activation)
- Determining priority

14.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0 to 12) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCRi.
For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 14.3, Interrupt Vector Table.

14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 14.2 shows the operation of the IR flag in IRn in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IR flag in IRn is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DMAC or DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DMAC/DTC transfer settings and transfer count. For details, see Table 14.4, Operation at DMAC/DTC Activation.



Note 1. One of the following requests is issued: CPU interrupt request, DTC activation request, and DMAC activation request. For details of the setting, see section 14.4.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC activation or DMAC activation, see Table 14.4, Operation at DMAC/DTC Activation.

Figure 14.2 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 14.3 to Figure 14.6 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 76, “internal delay + 2 PCLKB cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 77 to 95, “2 PCLKB cycles” of delay is added.

If an interrupt signal is generated every clock cycle, the subsequent interrupts cannot be detected; secure two or more clock cycles of the system clock or peripheral clock, whichever is slower, between issuance of continuous interrupt requests.

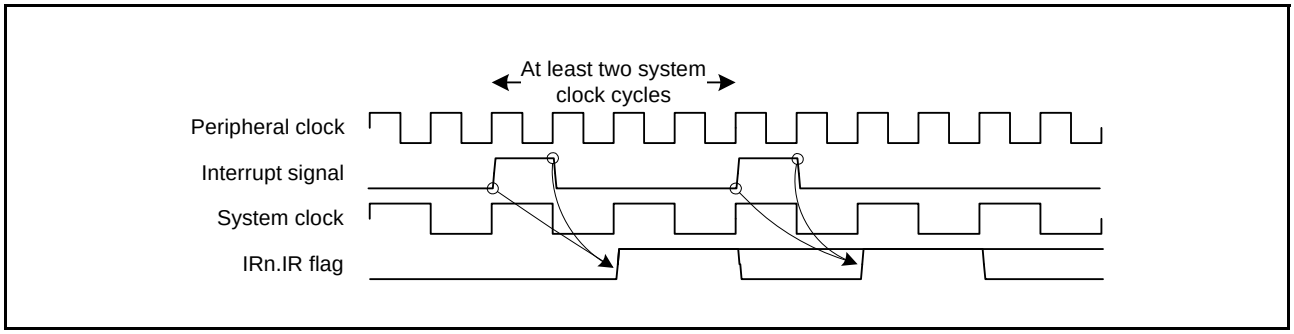


Figure 14.3 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Clock)

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1
Figure 14.4 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, or RIIC is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, see descriptions of the interrupts in section 30, Serial Communications Interface (SCIe, SCIf), section 33, I2C Bus Interface (RIIC), and section 34, Serial Peripheral Interface (RSPI).

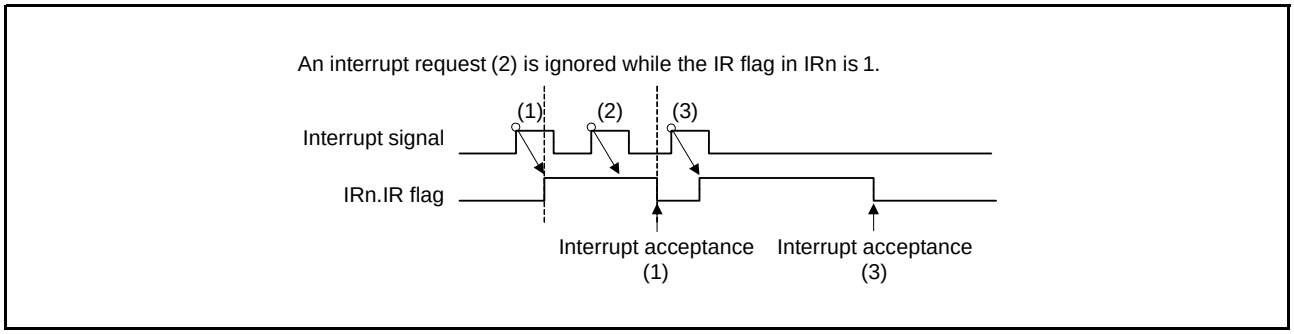


Figure 14.4 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 14.5 shows operation when the interrupt is disabled.

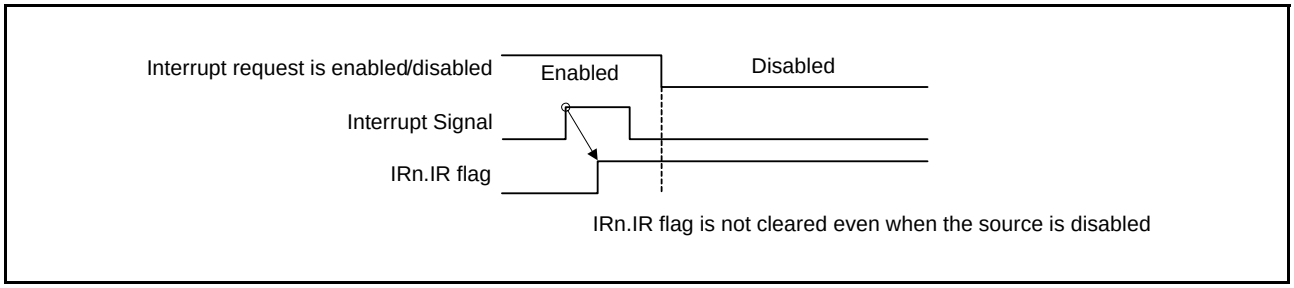


Figure 14.5 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 14.6 shows the operation of the interrupt status flag (IR flag) in IRn in the case of level detection of an interrupt from a peripheral module or an external pin.

The IR flag in IRn remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

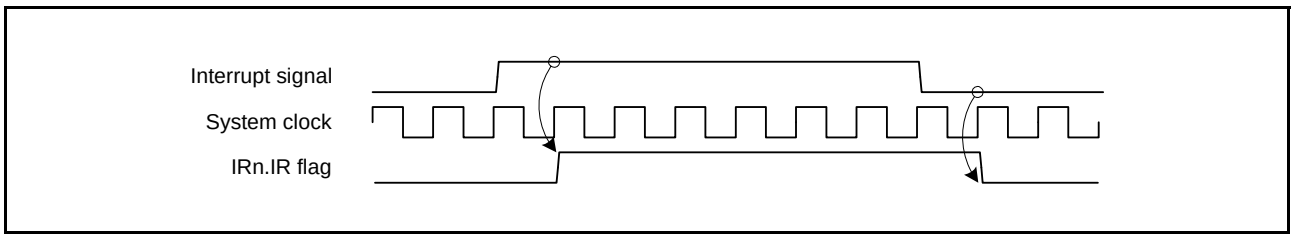


Figure 14.6 IRn.IR Flag Operation for Level Detection Interrupts

Figure 14.7 shows the procedure for handling level detection interrupts.

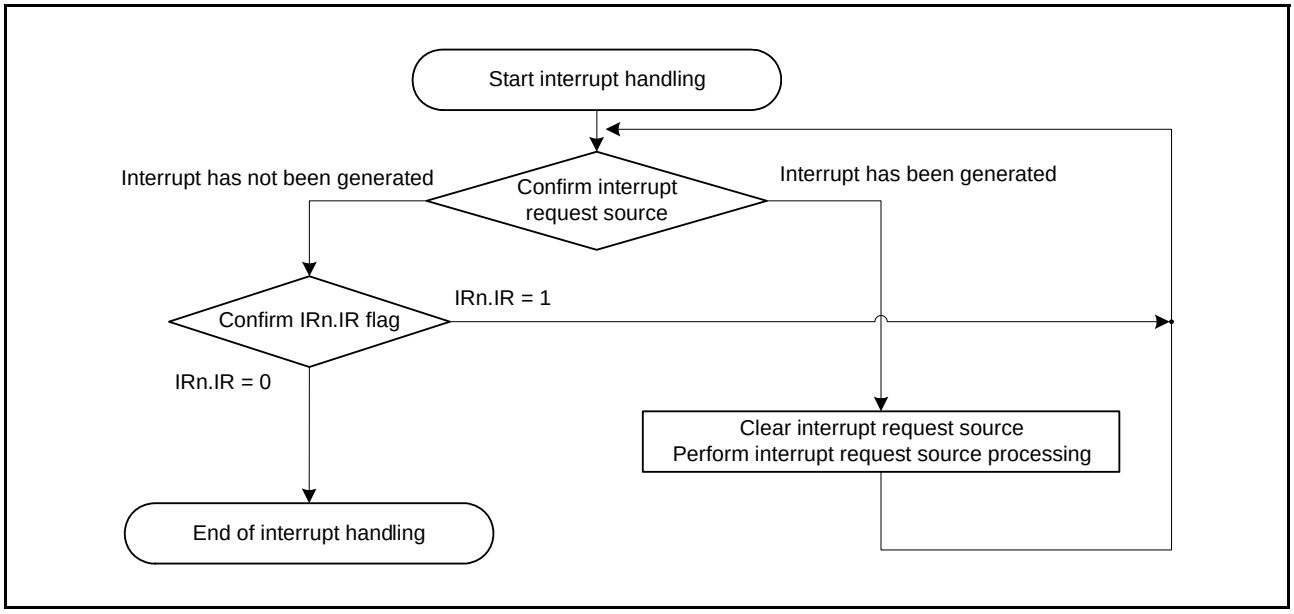


Figure 14.7 Procedure for Handling Level Detection Interrupts

14.4.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag is set to 1. Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, or RIIC from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, see descriptions of the interrupts in section 30, Serial Communications Interface (SCIe, SCIf), section 33, I^2C Bus Interface (RIIC), and section 34, Serial Peripheral Interface (RSPI).

14.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 14.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 14.3.
If the DMAC or DTC is selected as the destination for requests from an IRQ pin, be sure to set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

(1) DMAC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Specify the vector number of the desired interrupt in the DMAC activation request select register (DMRSRm) for the required channel of the DMAC.*1
2. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
3. Set the DMAC activation enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DMAC operation enable bit (DMAST.DMST) to 1. The order of making settings for each interrupt and enabling the DMAC operation enable bit does not matter.
For the DMAC setting procedure, refer to section 17.3.7, Activating the DMAC in section 17, DMA Controller (DMACA).

(2) DTC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Set the DTC activation enable bit in the DTC activation enable register (DTCERn.DTCE) for the pertinent source to 1.*1

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module start bit does not matter.
For the DTC setting procedure, refer to section 18.5, DTC Setting Procedure, in section 18, Data Transfer Controller (DTCa).

Note 1. Do not set a DTC activation enable bit (DTCERn.DTCE) and a DMAC activation request select register (DMRSRm) to select the same source. Do not select the same source in more than one DMRSRm register.

(3) CPU Interrupt Request

If the interrupt request destination is neither the DMAC nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while neither the DMAC activation settings nor the DTC activation settings described above are in place.

Table 14.4 shows operation when the DTC or the DMAC is the request destination.

Table 14.4 Operation at DMAC/DTC Activation

Interrupt Request Destination	DISEL	Remaining Number of Transfer Operations	Operation per Request	IR*1	Interrupt Request Destination after Transfer
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DMA transfer	Cleared at the start of DMAC transfer	DMAC
		= 0	DMA transfer*2	Cleared at the start of DMAC transfer*2	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCER.DTCE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer information	DTC
		= 0	DTC transfer → CPU interrupt *2	Cleared on interrupt acceptance by the CPU*2	The DTCER.DTCE bit is cleared and the CPU becomes the destination.

DISEL for the DMAC is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.
Note 1. When the IRn.IR flag is 1, an interrupt request (DTC or DMAC activation request) that is generated again will be ignored.
Note 2. When the DISEL bit is 0, operation with the remaining number of transfer operations being 0 differs according to whether the source is for DTC or DMAC.
Note 3. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 18.3, Chain Transfer Conditions in section 18, Data Transfer Controller (DTCa).

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DMA activation source is to be changed while a transfer is not complete (i.e. while the DMACm.DMCNT.DTE bit has not been cleared) after the settings described under (1) DMAC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in

IERm to 0.
2. Check the state of transfer by the DMAC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DMAC Activation.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit has not been cleared) after the settings described under (2) DTC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (2) DTC Activation.

14.4.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.
The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRn takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn have no effect. An interrupt source with a smaller vector number takes precedence.

(3) Determining Priority when the DMAC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn have no effect. Regarding the order of priority of DMAC channels, see section 17, DMA Controller (DMACA).

14.4.5 Multiple Interrupts

To enable multiple interrupts of the CPU, set the PSW.I bit to 1 (interrupt enabled) in the handling routine of accepted interrupts.
The PSW.IPL[3:0] bits immediately after processing branches to the interrupt handling routine are set to the same value as the interrupt priority level of the accepted interrupt request. If an interrupt request which has an interrupt level higher than that of the PSW.IPL[3:0] bits is generated at this time, this interrupt request (for multiple interrupts) is accepted.
If the interrupt priority level of the accepted interrupt request is 15 (fast interrupt or interrupt when IPR[3:0] are set to 1111b), multiple interrupts are not generated.

14.4.6 Fast Interrupt

The fast interrupt is an interrupt for executing a faster interrupt response by the CPU, so only one of the interrupt sources can be assigned.
The interrupt priority level of the fast interrupt is 15 (highest) regardless of the setting of the IPRn.IPR[3:0] bits. In addition, the fast interrupt is accepted with precedence over other interrupt sources with level 15. However, when the value of the PSW.IPL[3:0] bits are 1111b (priority level 15), even the fast interrupt cannot be accepted.
To assign an interrupt source to the fast interrupt, specify the vector number of the source in the FIR.FVCT[7:0] bits, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
For details on the fast interrupt, see section 2, CPU, CPU and section 13, Exception Handling.

14.4.7 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 12) and NMI pin interrupt. The digital filter samples input signals at the filter sampling clock (PCLKB) and removes the pulses of which length is less than three sampling cycles.

To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64) with the IRQFLTC0, 1.FCLKSELi[1:0] bits (i = 0 to 12) and set the IRQFLTE0, 1.FLTENi bit to 1 (digital filter enabled).

To use the digital filter for the NMI pin, set the sampling clock cycle (PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

Figure 14.8 shows an example of digital filter operation.

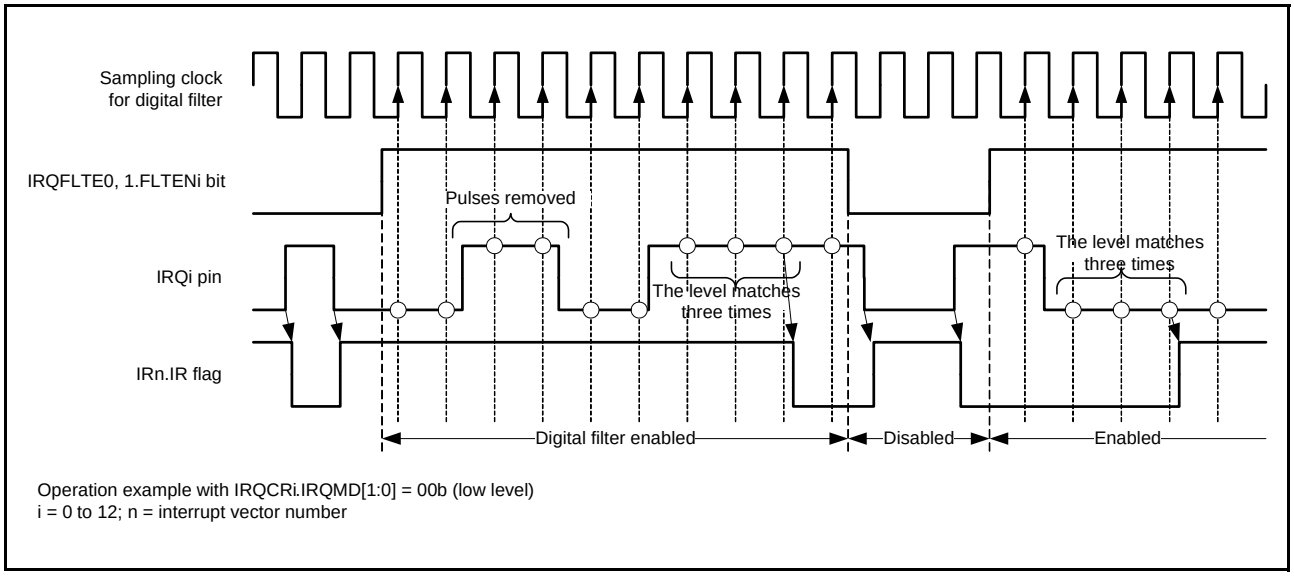


Figure 14.8 Digital Filter Operation Example

When entering software standby mode or deep software standby mode, set the IRQFLTE0, IRQFLTE1.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after returning from software standby mode or deep software standby mode, set the IRQFLTE0, IRQFLTE1.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

14.4.8 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit to 0 (interrupt request disabled).
2. Clear the IRQFLTE0, 1.FLTENi bit (i = 0 to 12) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTC0, 1.FCLKSELi[1:0] bits.
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag to 0 (if edge detection is in use).
7. Set the IRQFLTE0, 1.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DMAC activation, set the DMRSRp.DMRS[7:0] bits. If the interrupt is to be used for DTC activation, set the DTCERk.DTCE bit. The interrupt will be a CPU interrupt if neither of these settings is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

14.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error, IWDT underflow/refresh error, voltage monitoring 1 interrupt, and voltage monitoring 2 interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC or DMAC activation. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits in the NMISR have returned to 0 from within the handler for the non-maskable interrupt.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt processing, see section 13, Exception Handling.
Writing 1 to the NMICLR.NMICLR bit clears the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit clears the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.WDTCLR bit clears the WDT underflow/refresh error status flag (NMISR.WDTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit clears the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit clears the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.LVD2CLR bit clears the voltage monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.

14.6 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, all-module clock stop mode, or software standby mode are listed in Table 14.3, Interrupt Vector Table.
For details, refer to section 11, Low Power Consumption. The following describes how to use an interrupt to return operation from each low power consumption mode.

14.6.1 Return from Sleep Mode

If the interrupt controller is to return operation from sleep mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the CPU as the interrupt request destination.
2. Use the IENj bit in IERm to enable the given interrupt request.
3. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

14.6.2 Return from All-Module Clock Stop Mode

If the interrupt controller is to return operation from all-module clock stop mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the interrupt source that enables the return from the all-module clock stop mode.
2. Select the CPU as the interrupt request destination.
3. Use the IENj bit in IERm to enable the given interrupt request.
4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

14.6.3 Return from Software Standby Mode

The interrupt controller can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.
The conditions for the return are listed below.

- Interrupts

1. Select the interrupt source that enables the return from the software standby mode.
2. Select the CPU as the interrupt request destination.
3. Use the IENj bit in IERm to enable the given interrupt request.
4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
   (For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPRn) should be set above the level set by IPL in the PSW of the CPU.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

- Procedure to make a transition to/from software standby mode

1. Before software standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQFLTE0, 1.FLTENi = 0, NMIFLTE.NFLTEN = 0).
2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE0, 1.FLTENi = 1, NMIFLTE.NFLTEN = 1).

14.7 Usage Note

14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

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, ROM) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to ROM
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, DTC • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	• Connected to peripheral modules (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 and 3) • Operates in synchronization with the peripheral-module clock (PCLKB and PCLKD*1)
	Internal peripheral bus 3	• Connected to peripheral modules (CEC, RCR) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 6	• Connected to ROM (P/E) and E2 DataFlash memory • Operates in synchronization with the FlashIF clock (FCLK)
External bus	CS area	• Connected to the external devices • Operates in synchronization with the external-bus clock (BCLK)

Note 1. The peripheral module clock used as the operating clock is PCLKD for S12AD.
P/E: Programming/Erasure
BCLK (external-bus clock): 54 MHz (max.) The CSC (CS area controller) operate in synchronization with the BCLK.
BCLK pin output: The frequency is the same as the BCLK. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCKCR.BCLKDIV) in the external bus clock control register. For details, see section 9, Clock Generation Circuit.

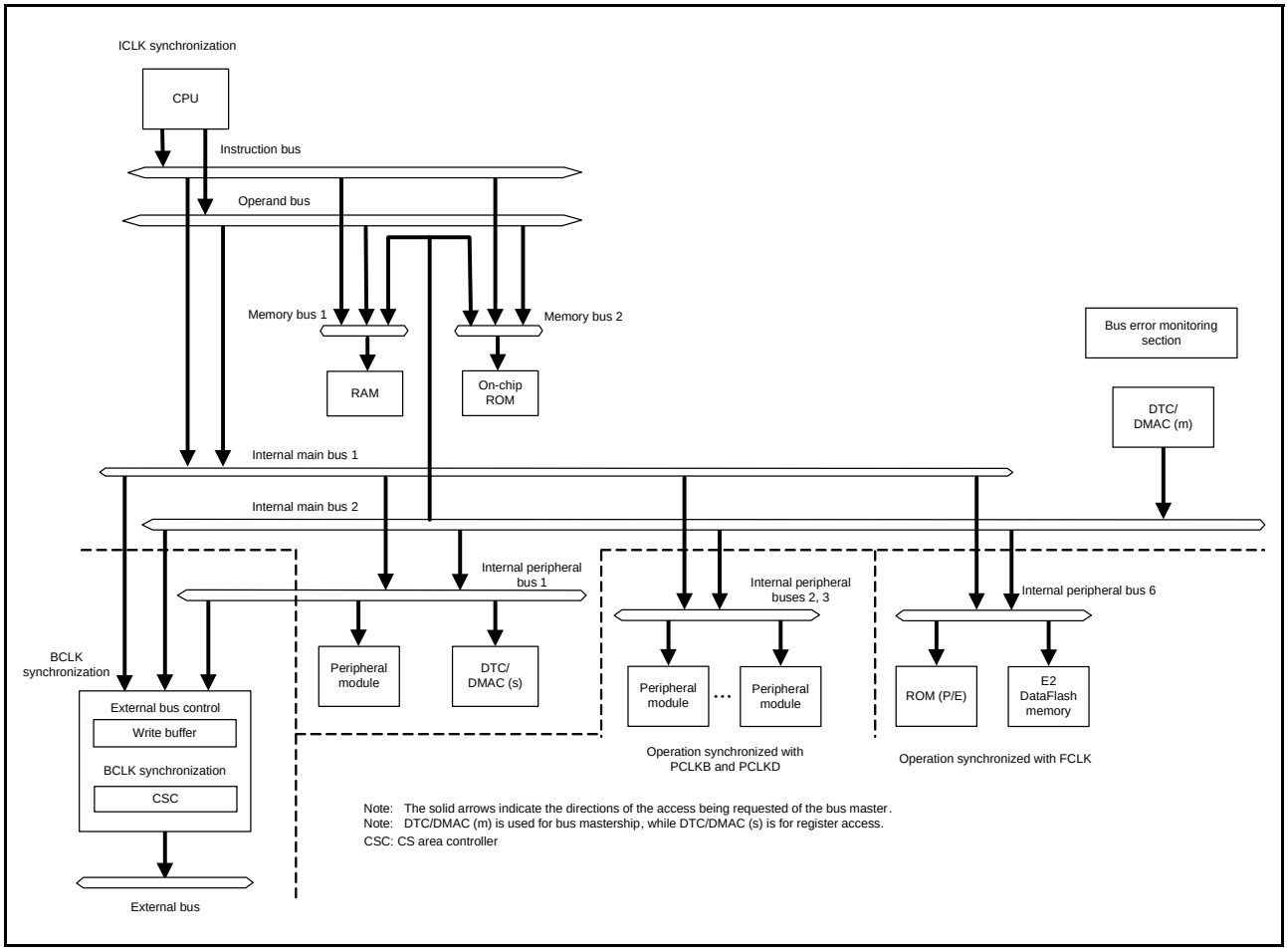


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

	Bus		Area	
Address	**On-chip ROM enabled**	**On-chip ROM disabled**	**On-chip ROM enabled**	**On-chip ROM disabled**
0000 0000h to 0001 FFFFh	Memory bus 1		RAM	
0002 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			
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved area	E2 DataFlash memory, and ROM (for programming/erasure)	Reserved area
0500 0000h to 07FF FFFFh	External bus		External address space (CS1 to CS3)	
0800 0000h to 0FFF FFFFh			Reserved area	
1000 0000h to 7FFF FFFFh	Reserved area		Reserved area	
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	ROM (for reading only)	Reserved area
FF00 0000h to FFFF FFFFh		External bus		External address space (CS0)

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. Connection of the instruction and operand buses to RAM and ROM 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 ROM 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 ROM and RAM or to ROM and external space is possible.

15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM is connected to memory bus 2. 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 CPU bus and internal main bus 2 can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) 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, 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, 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 activation request issues the bus mastership request. The priority order of activation requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 3, and 6, and external bus), 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 ↑ Low	DMAC
	DTC
	CPU

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	DTC, DMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1 and 3
Internal peripheral bus 3	CEC, RCR
Internal peripheral bus 6	ROM (P/E)/E2 DataFlash 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 3, and 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]), 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.
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 15.2).

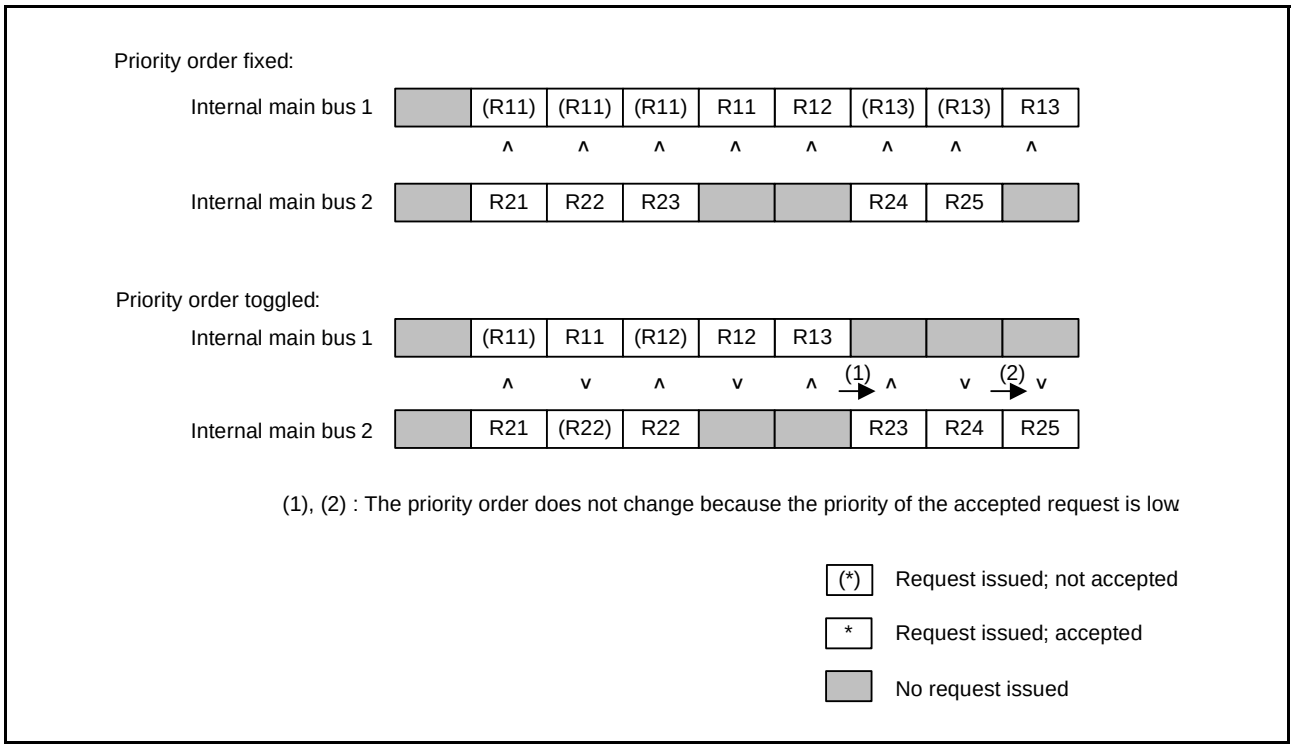


Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses

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 read 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).

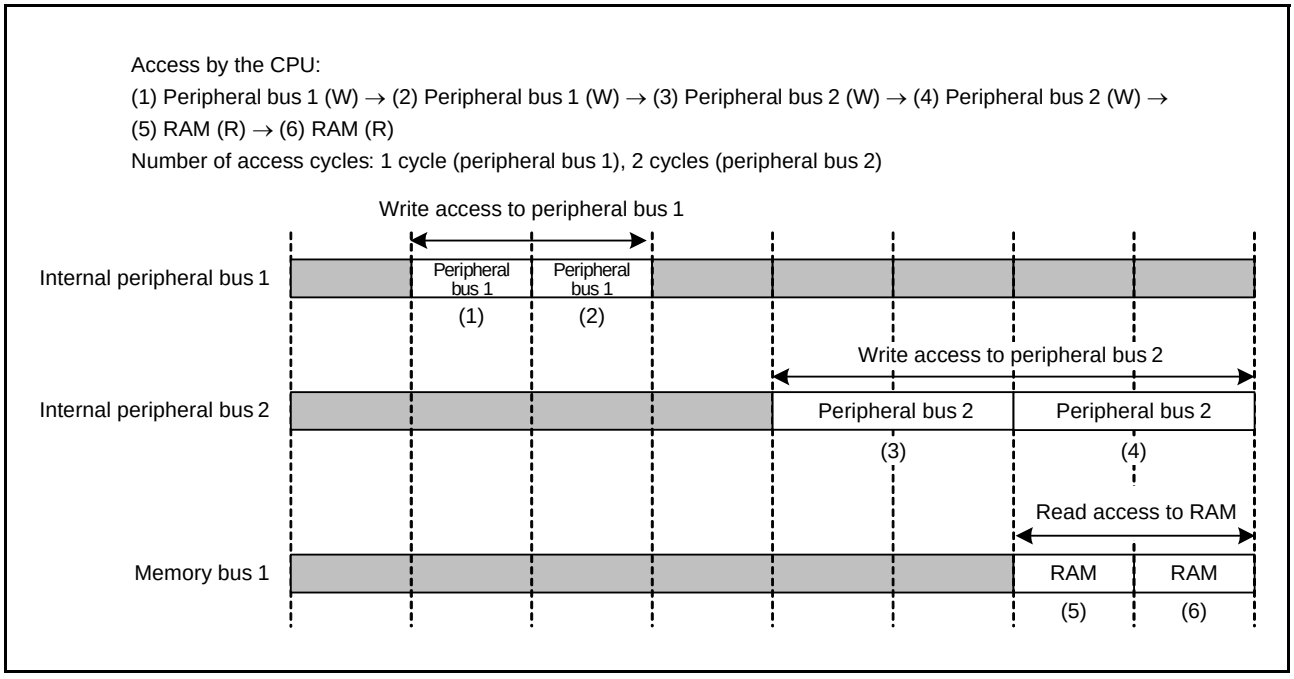


Figure 15.3 Write Buffer Function

15.2.6 External Bus

Table 15.5 lists the specifications of the external bus.
The external bus controller arbitrates requests for bus mastership from internal main bus 1, and internal main bus 2. The priority order of these two buses can be set using the external bus priority control bits (BPEB[1:0]) in the bus priority control register (BUSPRI). When the priority order is fixed, the order is internal main bus 2, and then internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and internal main bus 2.
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 15.2).

Table 15.5 Specifications of the External Bus

Item	Description
External address space	• An external address space is divided into four CS areas (CS0 to CS3) for management. • Chip select signals can be output for each area. • Bus width can be set for each area. Separate bus: An 8 or 16-bit bus space is selectable. Address/data multiplexed bus: An 8 or 16-bit bus space is selectable. • An endian mode can be specified for each area.
CS area controller	• Recovery cycles can be inserted. Read recovery: Up to 15 cycles Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles) • Wait control can be used to set up the following. Timing of assertion and negation for chip-select signals (CS0# to CS3#) The timing of assertion of the read signal (RD#) and write signals (WR#, WR0#, WR1#) The timing with which data output starts and ends • Write access mode: Single write strobe mode/byte strobe mode • Separate bus or address/data multiplexed bus can be set for each area.
Write buffer function	When write data from the bus master has been written to the write buffer, write access by the bus master is completed.
Frequency	• The CS area controller (CSC) operates in synchronization with the external-bus clock (BCLK).

Table 15.6 lists the input/output pins of the external bus.

Table 15.6 Pin Configuration of the External Bus

Pin Name	I/O	Description
A23 to A0*1	Output	Address output pins
D15 to D0	I/O	Data input/output pins D15 to D0 pins are enabled when the 16-bit bus space is specified. D7 to D0 pins are enabled when the 8-bit bus space is specified.
BC0#*1	Output	A strobe signal; (the BC0# signal being at the low level) during access to an external address space in single write strobe mode indicates that D7 to D0 are valid. When an 8-bit bus space is specified, this output pin is held low regardless of write access mode.
BC1#	Output	A strobe signal; (the BC1# signal being at the low level) during access to an external address space in single write strobe mode indicates that D15 to D8 are valid. This pin is not used when the 8-bit bus space is specified.
CS0#	Output	A chip select signal for area 0 (CS0)
CS1#	Output	A chip select signal for area 1 (CS1)
CS2#	Output	A chip select signal for area 2 (CS2)
CS3#	Output	A chip select signal for area 3 (CS3)
RD#	Output	A strobe signal indicating that reading from an external address space (CS0 to CS3) is in progress
WR0#/WR#*2	Output	WR0# signal is a strobe signal indicates that (the WR0# signal being at the low level) writing to an external address space is in progress in byte strobe mode, and D7 to D0 are valid. WR# signal is a strobe signal that indicates writing to an external address space is in progress in single write strobe mode. When an 8-bit bus space is specified, this output pin is held low during a write access regardless of write access mode.
WR1#	Output	A strobe signal; (the WR1# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D15 to D8 are valid. The BC1# signal is output in single write strobe mode. This pin is not used when the 8-bit bus space is specified.
ALE	Output	Address latch signal when address/data multiplexed bus is selected.
WAIT#	Input	A wait request signal when accessing the external address space (CS0 to CS3) (Low: Wait request)

Note 1. The A0 and BC0# pin functions share the same pin, and either becomes effective according to the area, with the function being A0 in byte strobe mode and BC0# in single write strobe mode. Note that setting the 8-bit external bus width is prohibited in single write strobe mode. For information on other multiplexed pin functions, see section 21, Multi-Function Pin Controller (MPC).

Note 2. The WR0# signal and WR# signal are identical. The WR0# signal is particularly referred to as WR# in single write strobe mode.

15.2.7 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 ROM and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and the external bus 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 ROM and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to RAM and ROM by the CPU.

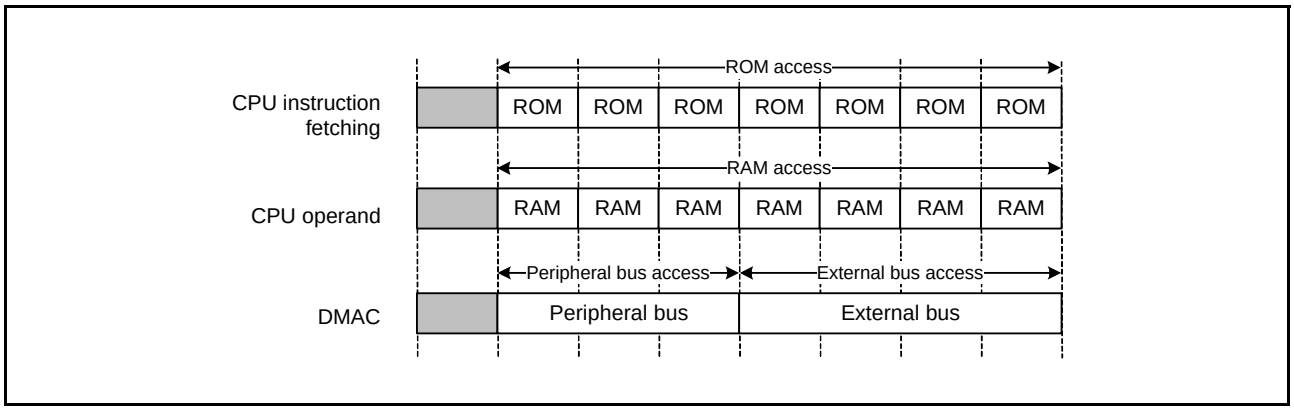


Figure 15.4 Example of Parallel Operations

15.2.8 Bus Settings

(1) Set the mode of the external bus in the CSn mode register (CSnMOD), CSn wait-control register 1 (CSnWCR1), CSn wait-control register 2 (CSnWCR2), CSn control register (CSnCR), CSn recovery-cycle setting register (CSnREC), CS recovery cycle insertion enable register (CSRECEN), bus error monitoring-enable register (BEREN), and bus-priority control register (BUSPRI).
(2) Make settings for pins in the CS output enable register (PFCSE), address output enable register 0 (PFAOE0), address output enable register 1 (PFAOE1), external-bus control register 0 (PFBCR0), and external-bus control register 1 (PFBCR1).
(3) Set up pins to be used as input port pins.
(4) Set the external-bus enable bit (EXBE) in the system control register 0 (SYSCR0) to 1 (enabling the external bus).

15.2.9 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

(a) 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 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.

(b) 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.

(3) Restriction on Endian

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

15.3 Register Descriptions

15.3.1 CSn Control Register (CSnCR) (n = 0 to 3)

Address(es): CS0CR 0008 3802h

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

Address(es): CS1CR 0008 3812h, CS2CR 0008 3822h, CS3CR 0008 3832h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after 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	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: Setting prohibited 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n is the same as the endian of operating mode. 1: Endian of area n is not the endian of operating mode. (n = 0 to 3)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n. (n = 0 to 3)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not write to the CSnCR register while access to the CSn area is in progress.

EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective areas.

After this LSI is reset, operation is enabled (EXENB = 1) only for area 0; operation in other areas is disabled (EXENB = 0).

An attempt at access to an area for which operation has been disabled does not lead to access via the external bus. However, if the illegal address access detection enable bit in the bus error monitoring enable register has been set to enable detection (BEREN.IGAEN = 1), such an attempt will lead to an illegal-access error.

BSIZE[1:0] Bits (External Bus Width Select)

These bits specify the data bus width of each area.
The data bus width of area 0 (CS0) after a reset depends on the setting of the bus width in operating mode.
When the address/data multiplexed I/O interface is selected with the MPXEN bit, the BSIZE[1:0] bits should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

EMODE Bit (Endian Mode)

This bit specifies the endian of each area.
When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

MPXEN Bit (Address/Data Multiplexed I/O Interface Select)

This bit specifies the bus interface for each area.

15.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 3)

Address(es): CS0REC 0008 380Ah, CS1REC 0008 381Ah, CS2REC 0008 382Ah, CS3REC 0008 383Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[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	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not write to the CSnREC register while access to the CSn area is in progress.

When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENj (j = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj) in CSRECEN.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

15.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): 0008 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Do not write to the CSRECEN register while access to the CSn area is in progress.

RCVEN0 Bit (Separate Bus Recovery Cycle Insertion Enable 0)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

RCVEN1 Bit (Separate Bus Recovery Cycle Insertion Enable 1)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in a different area.

RCVEN2 Bit (Separate Bus Recovery Cycle Insertion Enable 2)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the same area.

RCVEN3 Bit (Separate Bus Recovery Cycle Insertion Enable 3)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in a different area.

RCVEN4 Bit (Separate Bus Recovery Cycle Insertion Enable 4)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the same area.

RCVEN5 Bit (Separate Bus Recovery Cycle Insertion Enable 5)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in a different area.

RCVEN6 Bit (Separate Bus Recovery Cycle Insertion Enable 6)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the same area.

RCVEN7 Bit (Separate Bus Recovery Cycle Insertion Enable 7)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in a different area.

RCVENM0 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 0)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

RCVENM1 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 1)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in a different area.

RCVENM2 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 2)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the same area.

RCVENM3 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 3)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in a different area.

RCVENM4 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 4)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the same area.

RCVENM5 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 5)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in a different area.

RCVENM6 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 6)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the same area.

RCVENM7 Bit (Multiplexed Bus Recovery Cycle Insertion Enable 7)

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in a different area.

Table 15.7 Insertion of Recovery Cycles

Access Type	External Address Space	Insertion of Recovery Cycles	Corresponding Bits (Separate/Multiplexed)
Read access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN0/RCVENM0
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN1/RCVENM1
Read access after write access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN2/RCVENM2
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN3/RCVENM3
Write access after read access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN7/RCVENM7

15.3.4 CSn Mode Register (CSnMOD) (n = 0 to 3)

Address(es): CS0MOD 0008 3002h, CS1MOD 0008 3012h, CS2MOD 0008 3022h, CS3MOD 0008 3032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after 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	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.

Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0, 1) signal corresponding to the respective byte positions.

Writing 1 to this bit selects single write strobe mode where data write operation is controlled by the BCn# (n = 0, 1) signal and the WR# signal corresponding to respective byte positions. Note that setting the external bus width of 8 bits is prohibited in single write strobe mode.

Table 15.8 Control Signals for Write Access Mode

Mode	Pin Name			
Write Access Mode	**WR1#**	**WR0#/WR#**	**BC1#**	**BC0#**
Byte strobe mode	○	○	×	×
Single write strobe mode	×	○	○	○

○: Enabled, ×: Disabled

EWENB Bit (External Wait Enable)

This bit enables or disables external wait.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

Writing 0 to this bit disables the WAIT# signal.

PRENB Bit (Page Read Access Enable)

This bit enables or disables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB Bit (Page Write Access Enable)

This bit enables or disables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PRMOD Bit (Page Read Access Mode Select)

This bit selects a page read access operating mode.

Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.

Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

15.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 3)

Address(es): CS0WCR1 0008 3004h, CS1WCR1 0008 3014h, CS2WCR1 0008 3024h, CS3WCR1 0008 3034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
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
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select *1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select *2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles is inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles is inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles is inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles is inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles is inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles is inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles is inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles is inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles is inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles is inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles is inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles is inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles is inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles is inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles is inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles is inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles is inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles is inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles is inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles is inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles is inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles is inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles is inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles is inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles is inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles is inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles is inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles is inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles is inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles is inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles is inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles is inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles is inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles is inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles is inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles is inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles is inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles is inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles is inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles is inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles is inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles is inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles is inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles is inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles is inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles is inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles is inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles is inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles is inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles is inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.
Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

Do not write to the CSnWCR1 register while access to the CSn area is in progress.
Set each of these bits within a range of the restrictions described in section 15.6.1, Limitations on Using Separate Bus Interface or section 15.6.2, Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

CSPWWAIT[2:0] Bits (Page Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle.
This setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle.
This setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] Bits (Normal Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

15.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 3)

Address(es): CS0WCR2 0008 3008h, CS1WCR2 0008 3018h, CS2WCR2 0008 3028h, CS3WCR2 0008 3038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[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
	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles is inserted. 1 1: Wait with a length of 3 clock cycles is inserted.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	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
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not write to the CSnWCR2 register while access to the CSn area is in progress.
Set each of these bits within a range of the restrictions described in section 15.6.1, Limitations on Using Separate Bus Interface or section 15.6.2, Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

CSROFF[2:0] Bits (Read-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (RD# signal negated) until the CSn# signal (n = 0 to 3) is negated in read access mode.

CSWOFF[2:0] Bits (Write-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0, 1) negated) until the CSn# signal (n = 0 to 3) is negated in write access mode.

Note: Be sure to satisfy WDOFF[2:0] value ≤ CSWOFF[2:0] value.

WDOFF[2:0] Bits (Write Data Output Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0, 1) negated) until the write data output is completed in write access mode.

Note: Be sure to satisfy WDOFF[2:0] value ≤ CSWOFF[2:0] value.

AWAIT[1:0] Bits (Address Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value
For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] Bits (RD Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] Bits (WR Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the WRn# signal (n = 0, 1) is asserted.

Note: For normal write access, 1 ≤ satisfy CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, 1 ≤ satisfy CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] Bits (Write Data Output Wait Select)

These bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, 1 ≤ satisfy CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, 1 ≤ satisfy CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] Bits (CS Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 3) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

15.3.7 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.

15.3.8 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 cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not clear the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

15.3.9 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.

15.3.10 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

15.3.11 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
	—	—	BPEB[1:0]		BPFB[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 (ROM) 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	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
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
b13, b12	BPEB[1:0]	External Bus Priority Control	b13 b12 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, b14	—	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. When they are written to more than one time, the operation is not guaranteed.

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 the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPRO[1:0] Bits (Memory Bus 2 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

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.

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.

BPEB[1:0] Bits (External Bus Priority Control)

These bits specify the priority order for the external bus.
When the priority order is fixed, the order is internal main bus 2, and then internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and internal main bus 2.

15.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit or 16-bit bus space), data size, and endian format when accessing the external address space (the CS).

15.4.1 Data Alignment Control for CS Area

(1) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address bus A0 is disabled (output the low level).
When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# and WR1# pins are enabled. The BC0#, and BC1# pins are not used.
When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and outputs the low level during write access, regardless of the data size. Here, the WR1# pin is invalid (output the high level). The valid byte position is indicated by the BC0# and BC1# pins.
In 16-bit bus space, page access can occur in access to data in 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary and causes no change in BC0# and BC1# signals. The situations in which page access occurs are indicated by the letter (p) in Figure 15.5 and Figure 15.6.
In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n		15 8 7 0
	4n+1	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+2		15 8
	4n+2	One	First	16 bits	4n+2		15 8 7 0
	4n+3	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Two	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2	(p)	31 24 23 16
	4n+1	Three	First	8 bits	4n		7 0
			Second	16 bits	4n+2		23 16 15 8
			Third	8 bits	4n+4		31 24
	4n+2	Two	First	16 bits	4n+2		15 8 7 0
			Second	16 bits	4n+4		31 24 23 16
	4n+3	Three	First	8 bits	4n+2		7 0
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+6		31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.5 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1# / WR0#/BC0# / RD# Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n	7 0
	4n+1	One	First	8 bits	4n	7 0
	4n+2	One	First	8 bits	4n+2	7 0
	4n+3	One	First	8 bits	4n+2	7 0
16 bits	4n	One	First	16 bits	4n	15 8 7 0
	4n+1	Two	First	8 bits	4n	15 8
			Second	8 bits	4n+2	7 0
	4n+2	One	First	16 bits	4n+2	15 8 7 0
	4n+3	Two	First	8 bits	4n+2	15 8
			Second	8 bits	4n+4	7 0
32 bits	4n	Two	First	16 bits	4n	31 24 23 16
			Second	16 bits	4n+2 (p)	15 8 7 0
	4n+1	Three	First	8 bits	4n	31 24
			Second	16 bits	4n+2	23 16 15 8
			Third	8 bits	4n+4	7 0
	4n+2	Two	First	16 bits	4n+2	31 24 23 16
			Second	16 bits	4n+4	15 8 7 0
	4n+3	Three	First	8 bits	4n+2	31 24
			Second	16 bits	4n+4	23 16 15 8
			Third	8 bits	4n+6	7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.6 Data Alignment (Big Endian) in 16-Bit Bus Space

(2) 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and outputs the low level during write access. The WR1#, BC0#, and BC1# pins are not used.

Page access can occur in access to data in 16- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 15.7 and Figure 15.8.

In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n+1		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+3		7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
	4n+1	Two	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
	4n+3	Two	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
			Third	8 bits	4n+2	(p)	23 16
			Fourth	8 bits	4n+3	(p)	31 24
	4n+1	Four	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
			Third	8 bits	4n+3	(p)	23 16
			Fourth	8 bits	4n+4		31 24
	4n+2	Four	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
			Third	8 bits	4n+4		23 16
			Fourth	8 bits	4n+5	(p)	31 24
	4n+3	Four	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
			Third	8 bits	4n+5	(p)	23 16
			Fourth	8 bits	4n+6	(p)	31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.7 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# Data Bus D15–D8	D7–D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+1	Two	First	8 bits	4n+1			15 8
			Second	8 bits	4n+2	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
	4n+3	Two	First	8 bits	4n+3			15 8
			Second	8 bits	4n+4			7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0
	4n+1	Four	First	8 bits	4n+1			31 24
			Second	8 bits	4n+2	(p)		23 16
			Third	8 bits	4n+3	(p)		15 8
			Fourth	8 bits	4n+4			7 0
	4n+2	Four	First	8 bits	4n+2			31 24
			Second	8 bits	4n+3	(p)		23 16
			Third	8 bits	4n+4			15 8
			Fourth	8 bits	4n+5	(p)		7 0
	4n+3	Four	First	8 bits	4n+3			31 24
			Second	8 bits	4n+4			23 16
			Third	8 bits	4n+5	(p)		15 8
			Fourth	8 bits	4n+6	(p)		7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.8 Data Alignment (Big Endian) in 8-Bit Bus Space

15.5 Operation of CS Area Controller

15.5.1 Separate Bus

The various periods in the timing charts are described below.
The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and BCLK pin output are the same, unless otherwise noted.
The external bus access is started at the rising edge of the BCLK pin output pulse. However, when the external bus clock (BCLK) frequency and BCLK pin output frequency are different, if two or more rounds of external bus access are generated in response to a single transfer request from the bus master, the second and subsequent rounds of the external bus access may be started at the falling edge of the BCLK pin output pulse according to the wait settings (see Figure 15.14 to Figure 15.18).

(a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles is selectable within the range from zero to 31. Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON), the RD assert wait select bits (RDON), the WR assert wait select bits (WRON), and the write data output wait select bits (WDON) in CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

(b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.
If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. (Tend) indicates the cycle where sampling of the wait signal starts.
After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point (e) below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point (d) below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF) and the write-access CS extension cycle select bits (CSWOFF) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.
In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.
For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

(d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).
In the case of normal access, this is inserted within the period of clock cycles of CS extension (point (c) above).
In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point (c) above). Valid address and data output are extended over this period, and the WRn# signal is negated.

(e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.
CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, see section 15.5.4, Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page-read and page-write access, respectively, all bus accesses will take the form of normal read and write operations.
Even when the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, bus access other than page access will take the form of normal read and write operations.
Figure 15.9 to Figure 15.11 show the normal access operations.

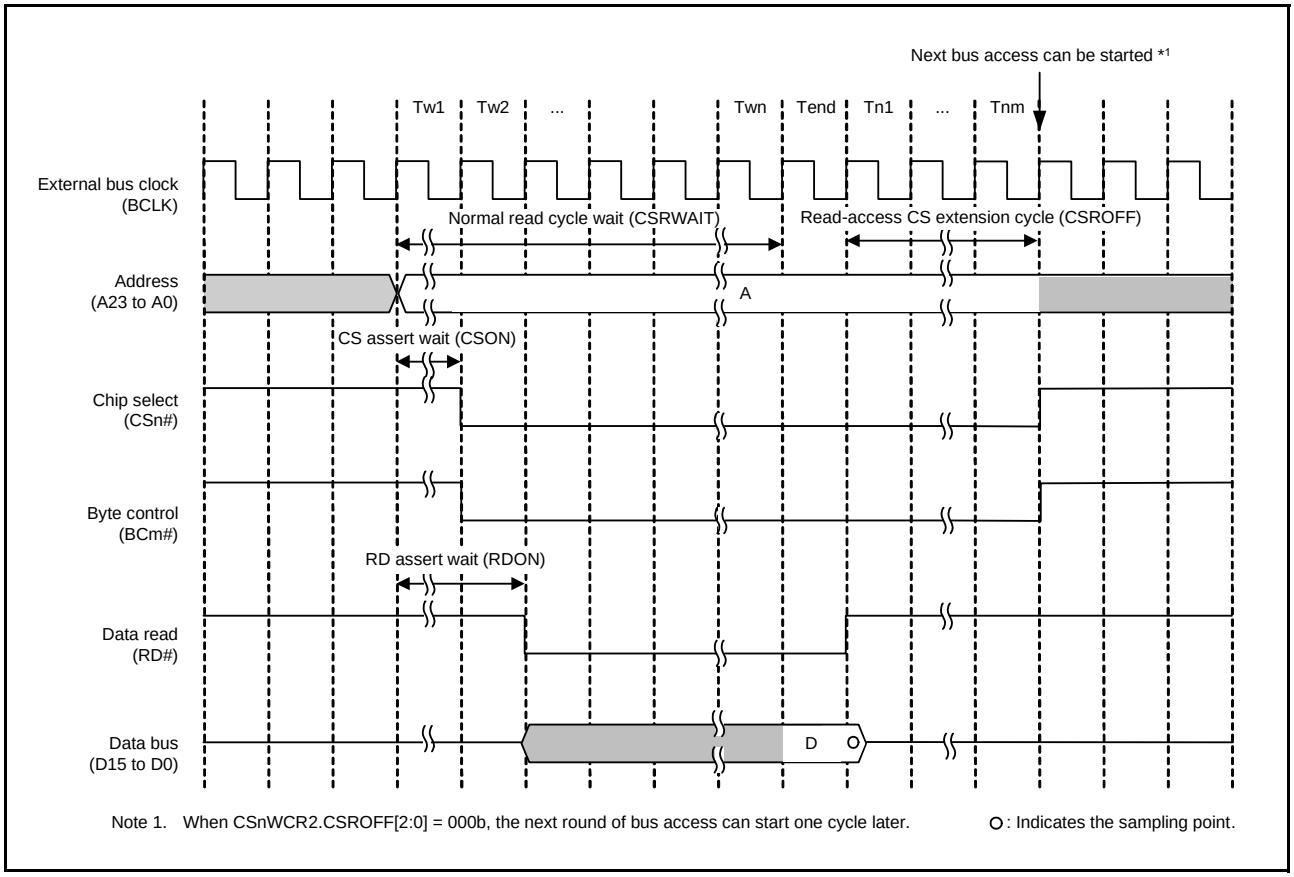


Figure 15.9 Bus Timing (Normal-Read Operation) (n = 0 to 3, m = 0, 1)

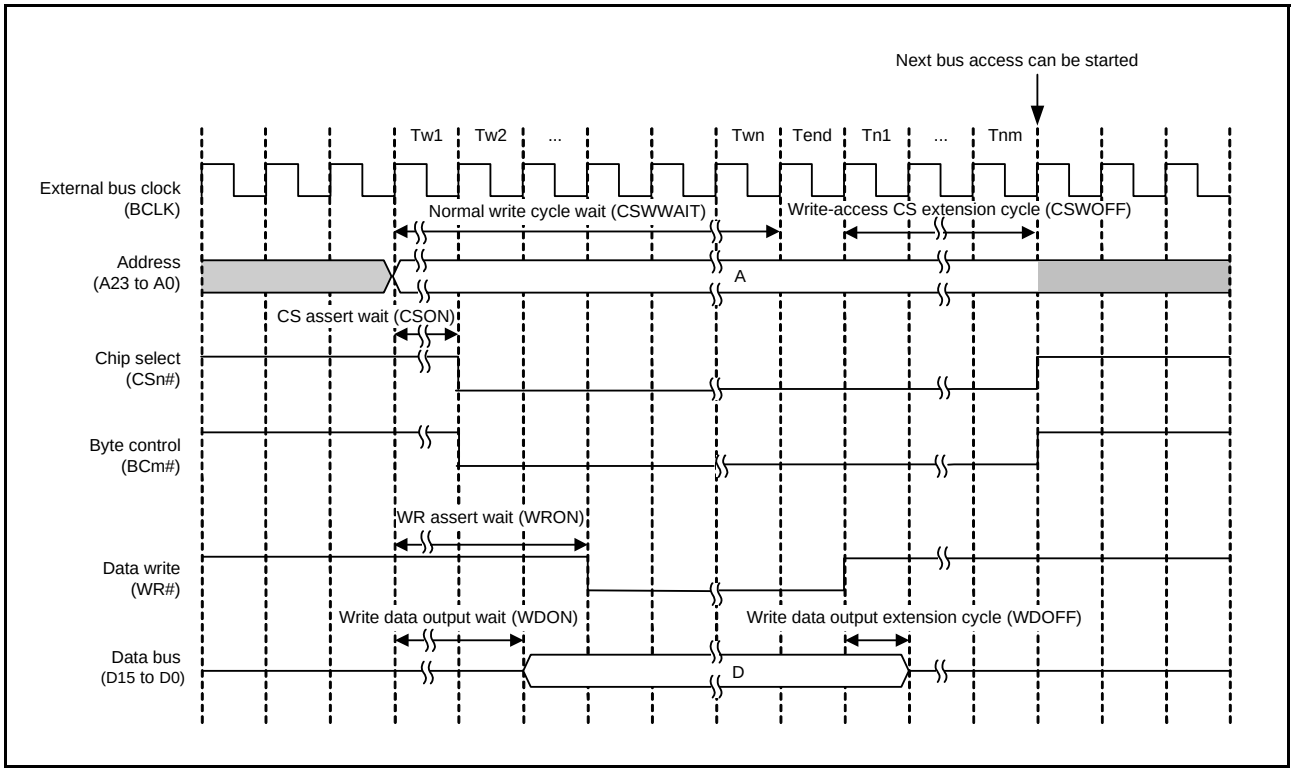


Figure 15.10 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 3, m = 0, 1)

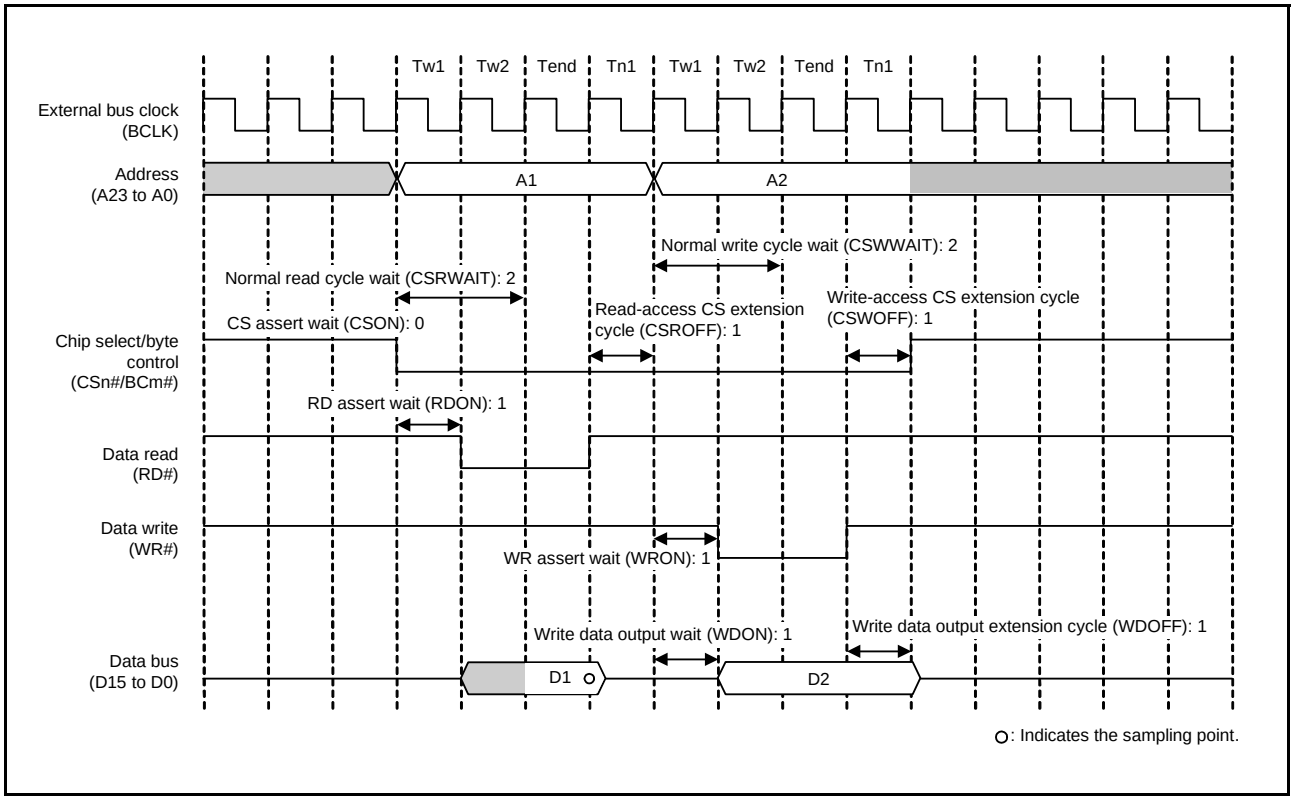


Figure 15.11 Example of Normal Access Operation (Read/Write) (n = 0 to 3, m = 0, 1)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations (steps (a) to (d) above) are repeated. Figure 15.12 and Figure 15.13 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. If the recovery cycle insertion condition is satisfied, recovery cycles (step (f) above) are also inserted in the second and subsequent external bus accesses (see Figure 15.30).

The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

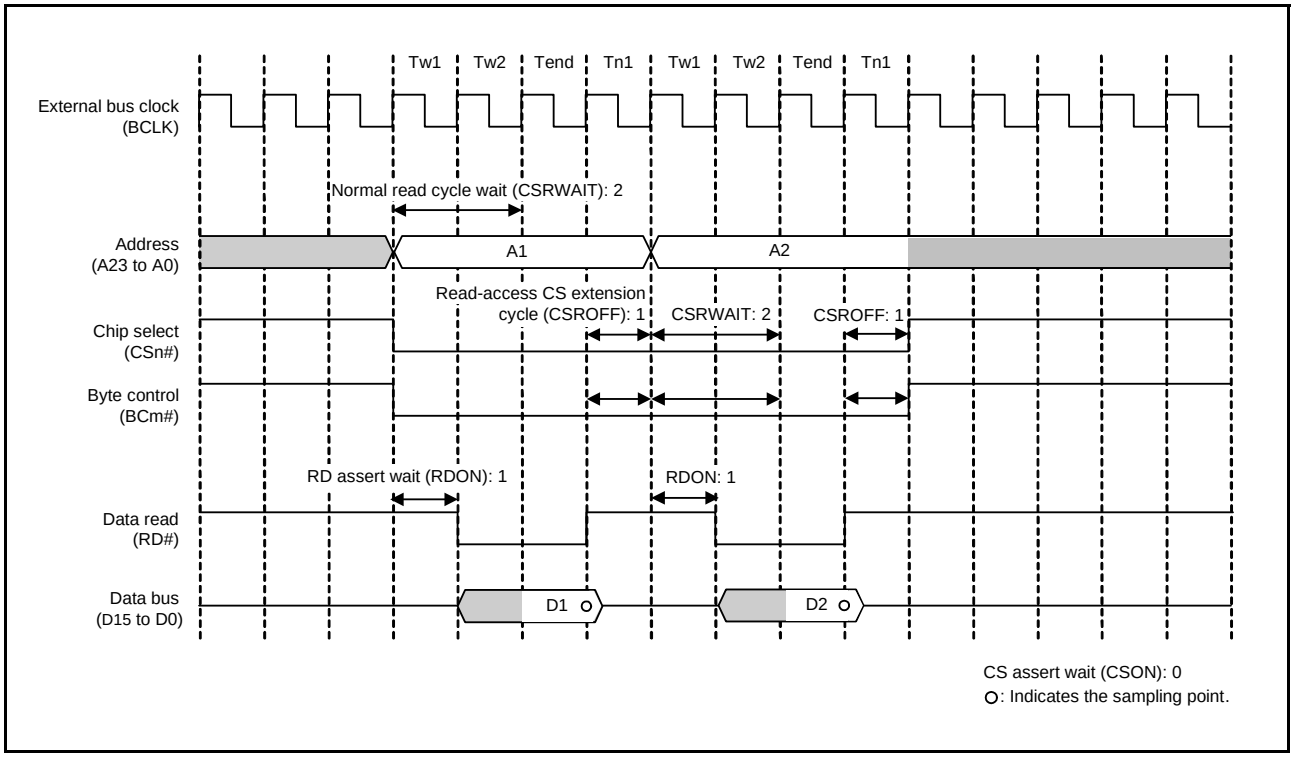


Figure 15.12 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

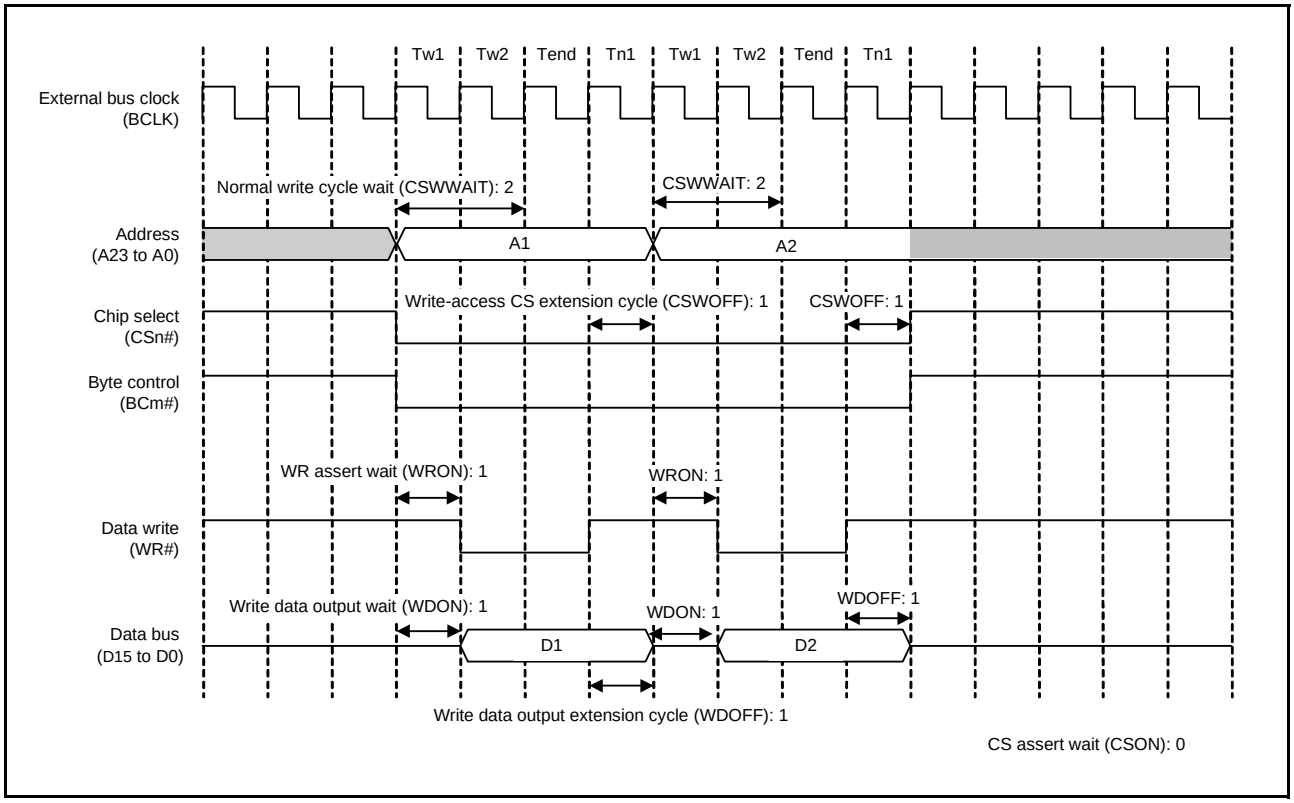


Figure 15.13 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 3, m = 0, 1)

Figure 15.14 to Figure 15.18 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.

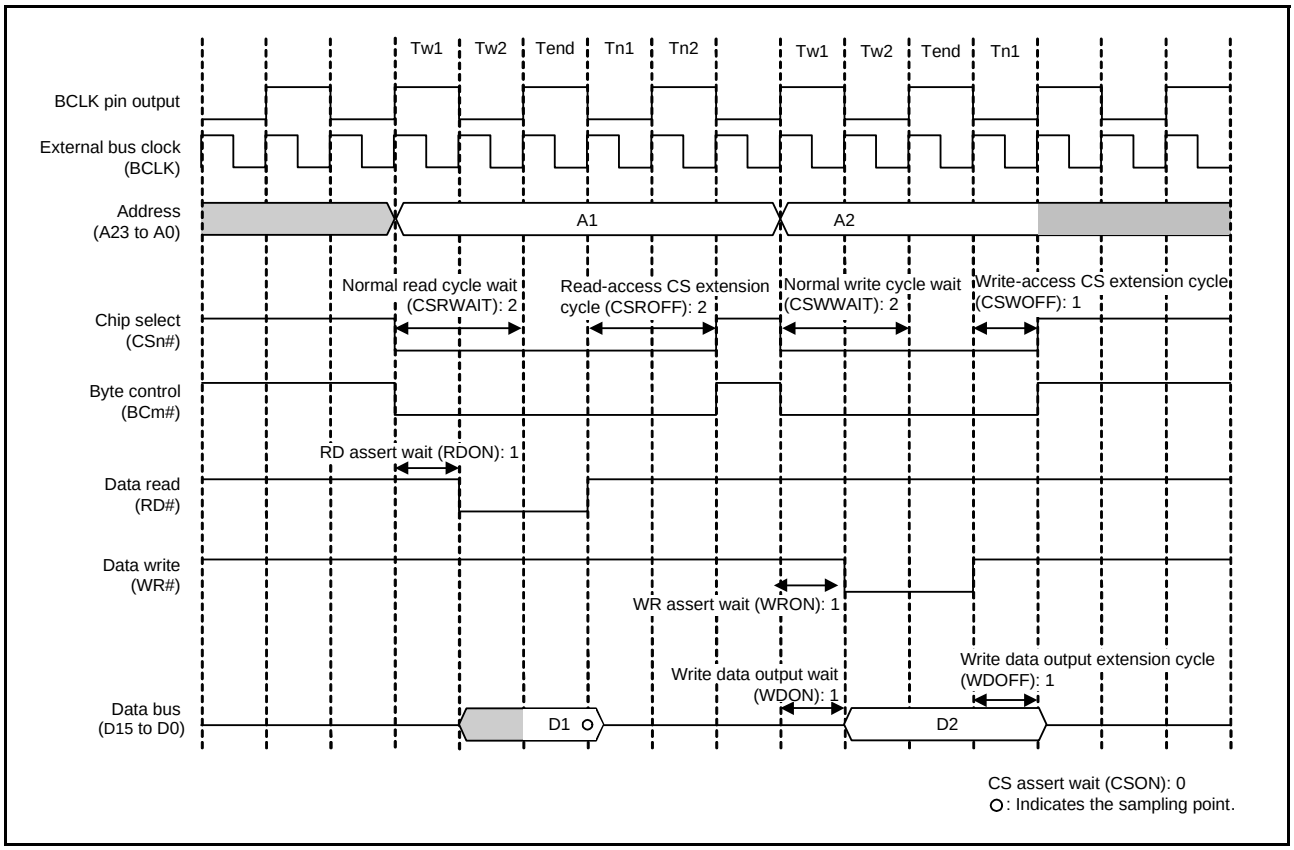


Figure 15.14 Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

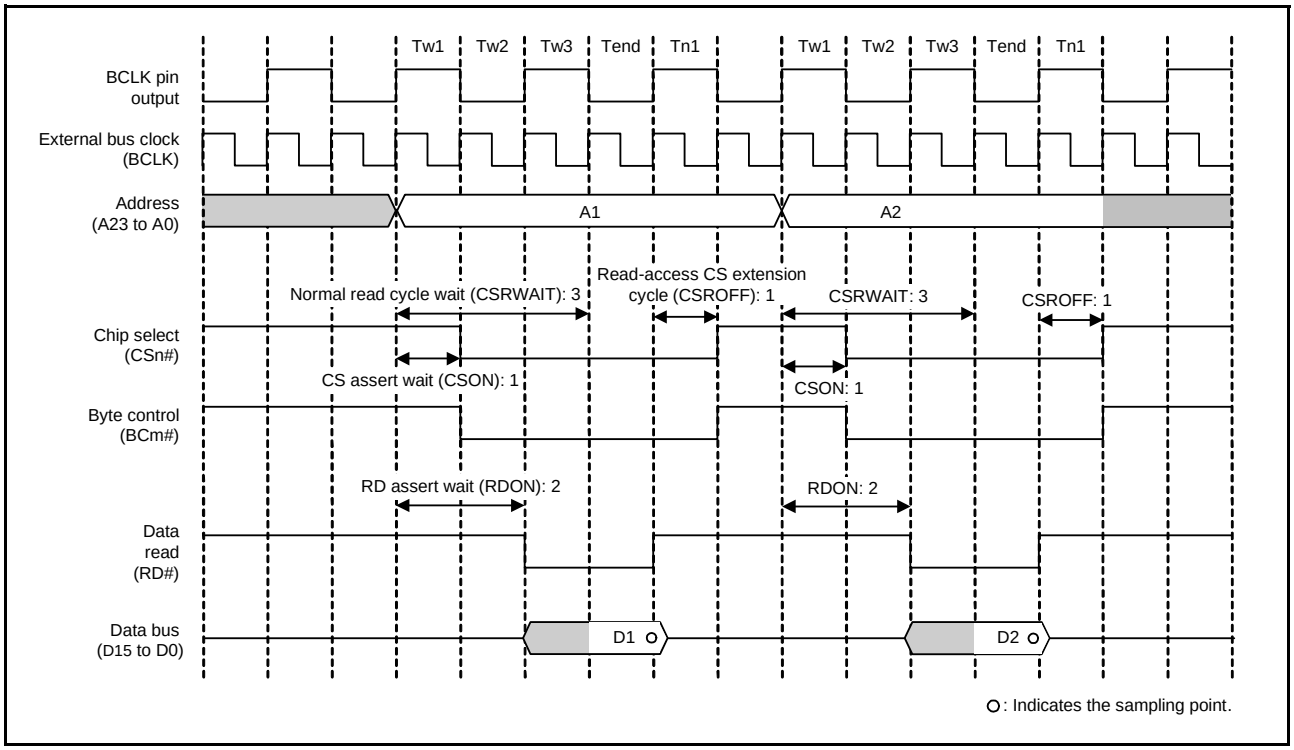


Figure 15.15 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

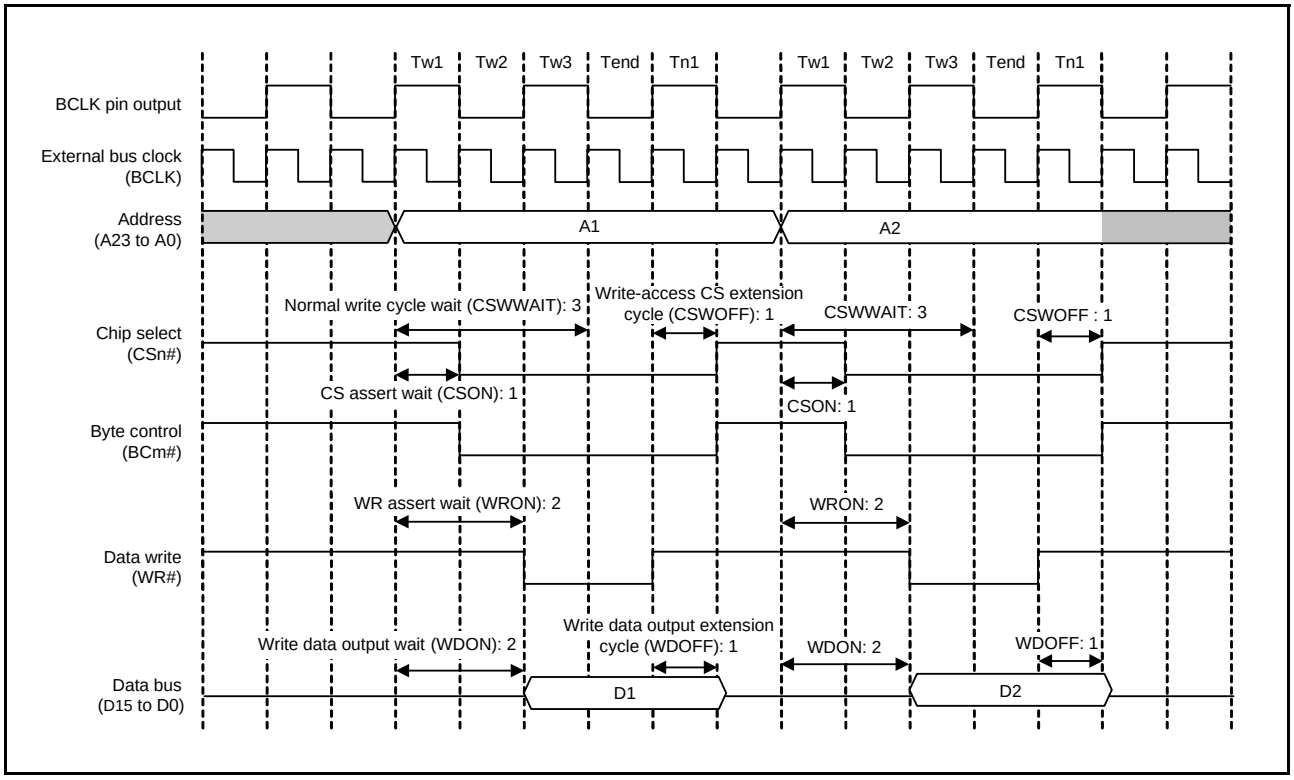


Figure 15.16 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 3, m = 0, 1)

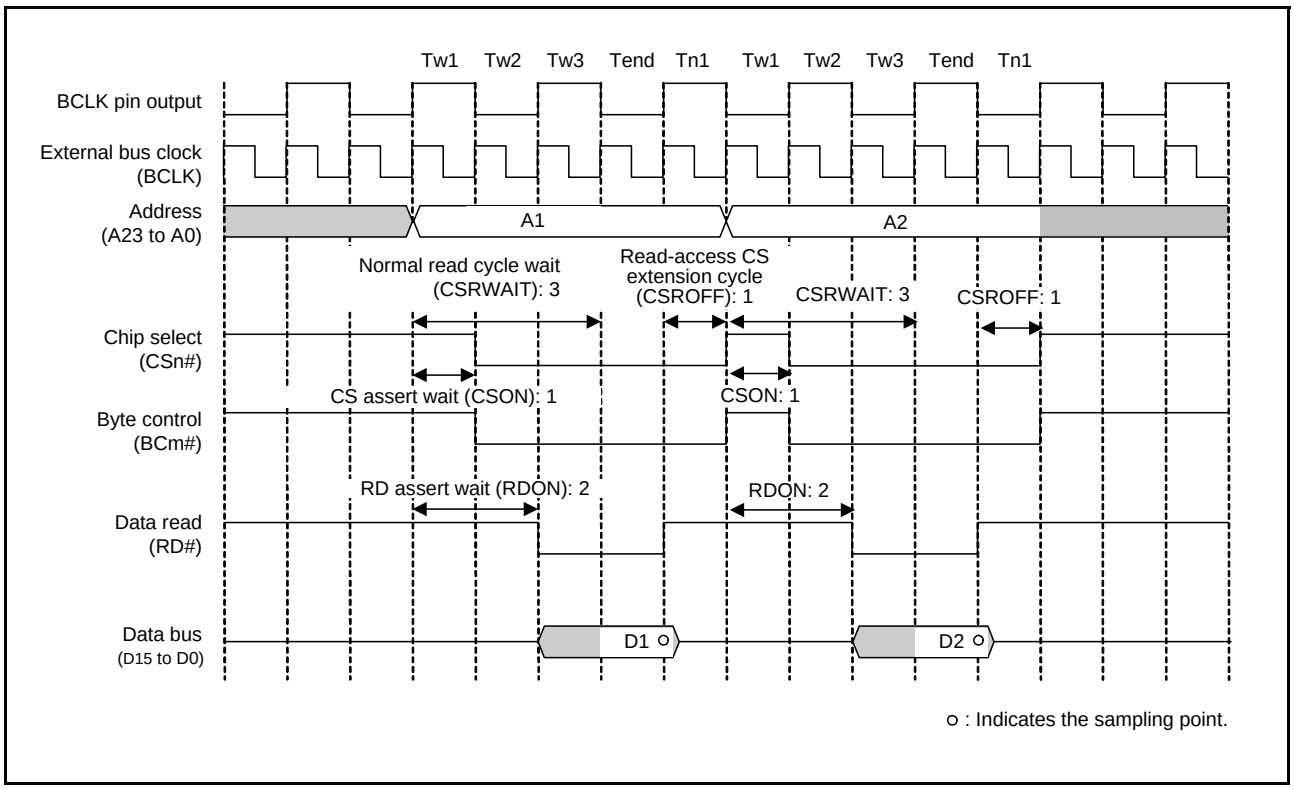


Figure 15.17 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

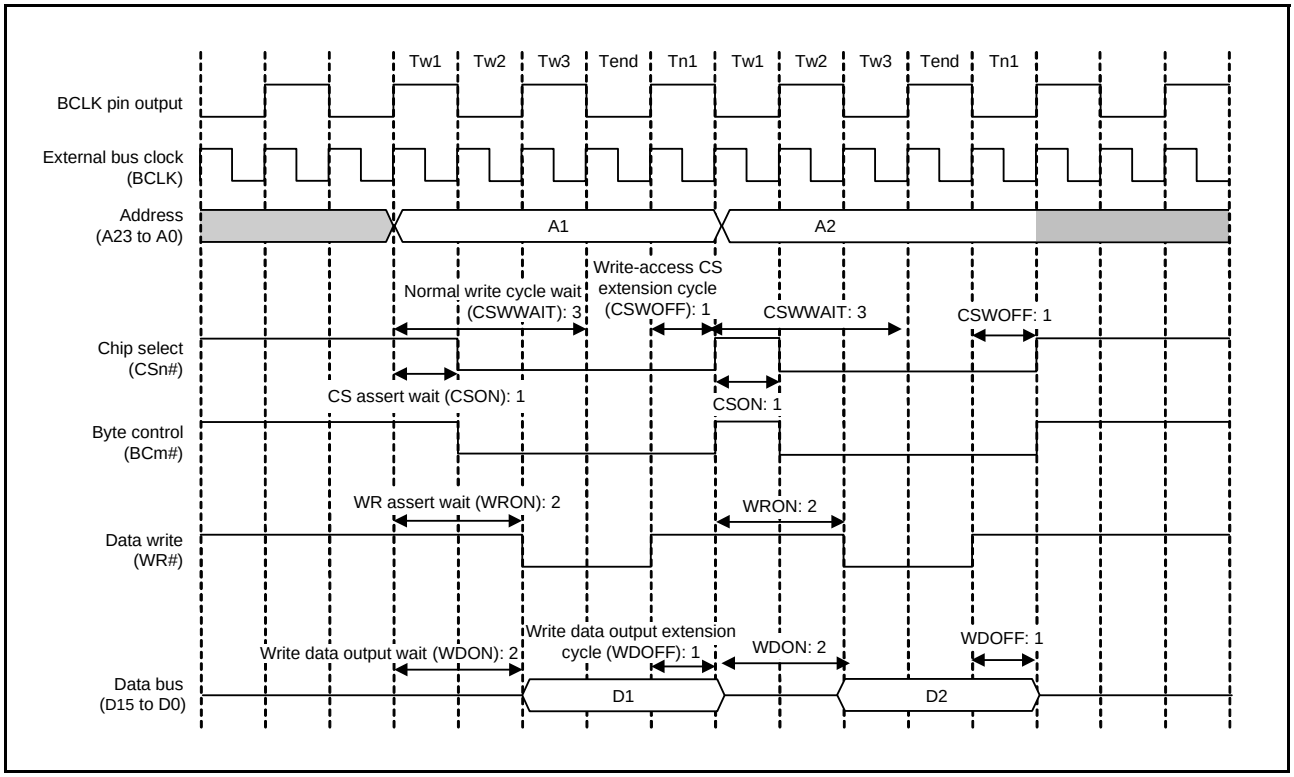


Figure 15.18 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3, m = 0, 1)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, the bus access for page access operations becomes page reading and writing. Specifically, page access can occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when the split accesses are not aligned or the access spreads over the 32-bit boundary. See Figure 15.5 to Figure 15.8 for the conditions under which page access occurs.

Figure 15.19 and Figure 15.20 show examples of page access operations.

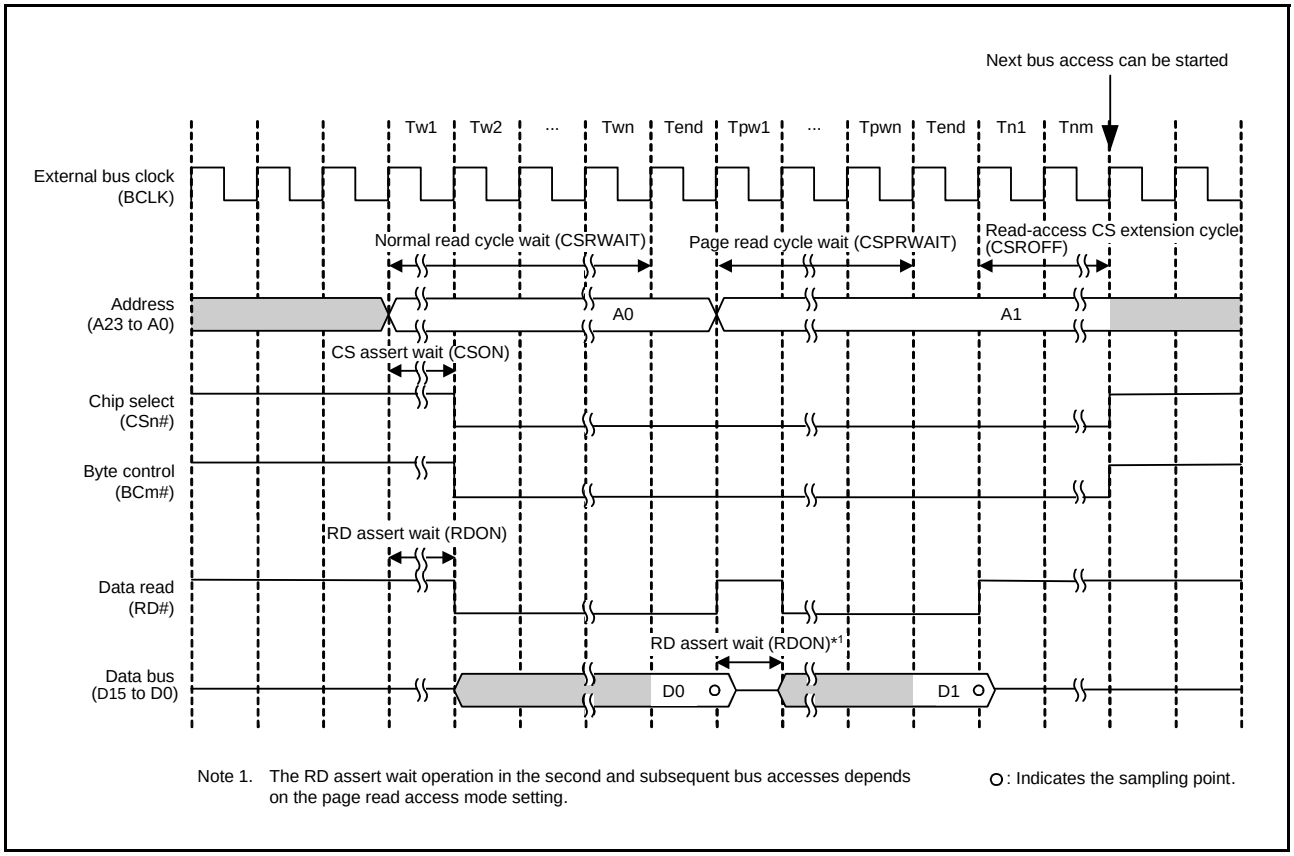


Figure 15.19 Page-Read Access Timing (n = 0 to 3, m = 0, 1)

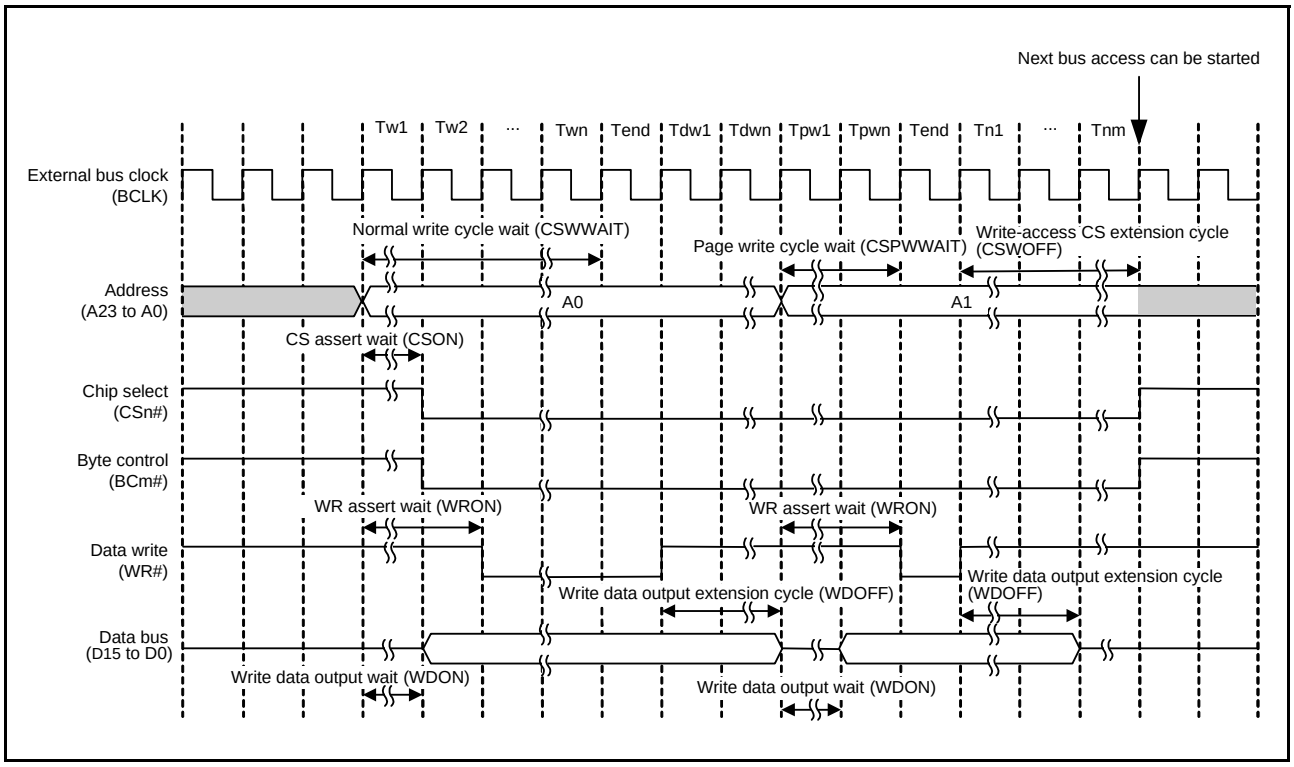


Figure 15.20 Page-Write Access Timing (n = 0 to 3, m = 0, 1)

Figure 15.21 and Figure 15.22 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.

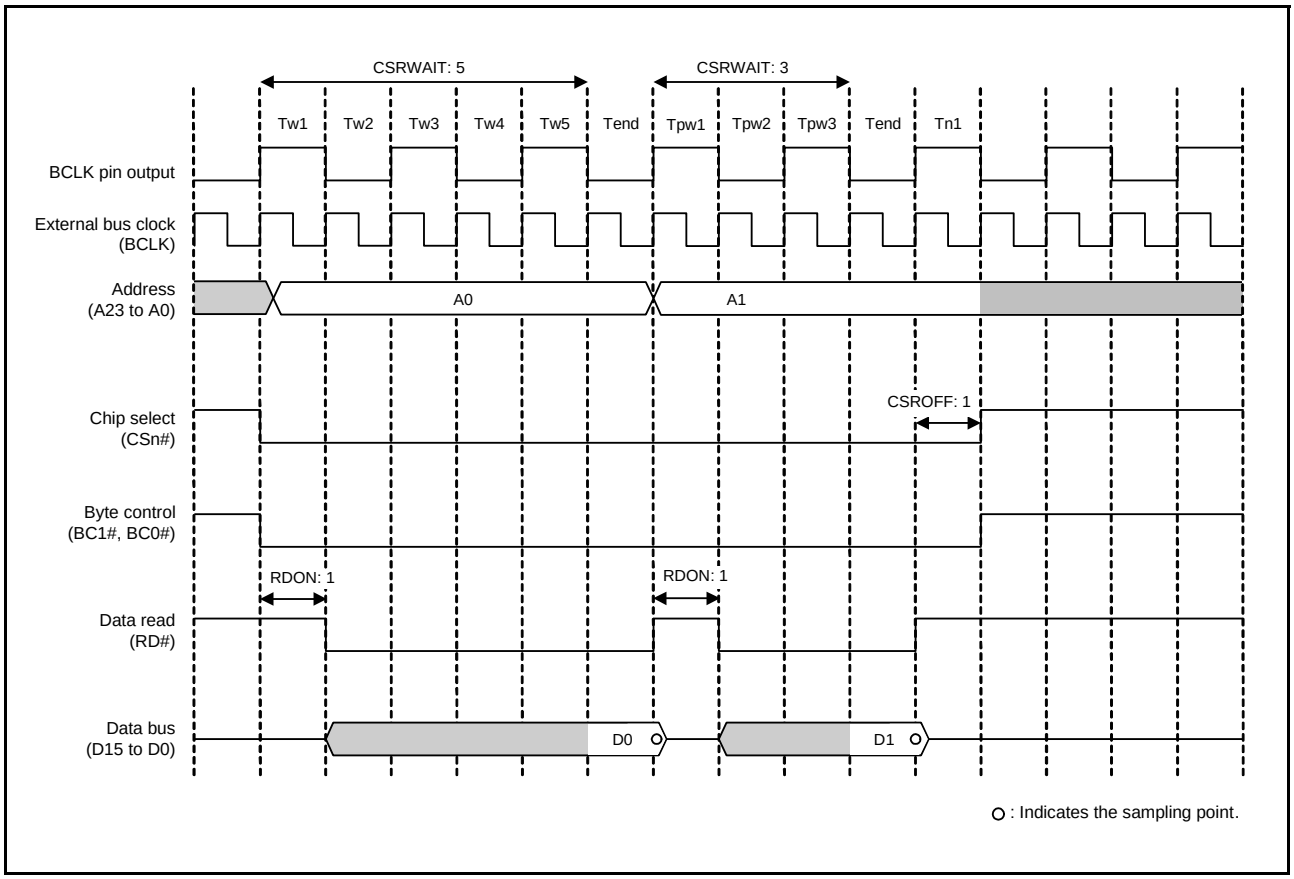


Figure 15.21 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 3)

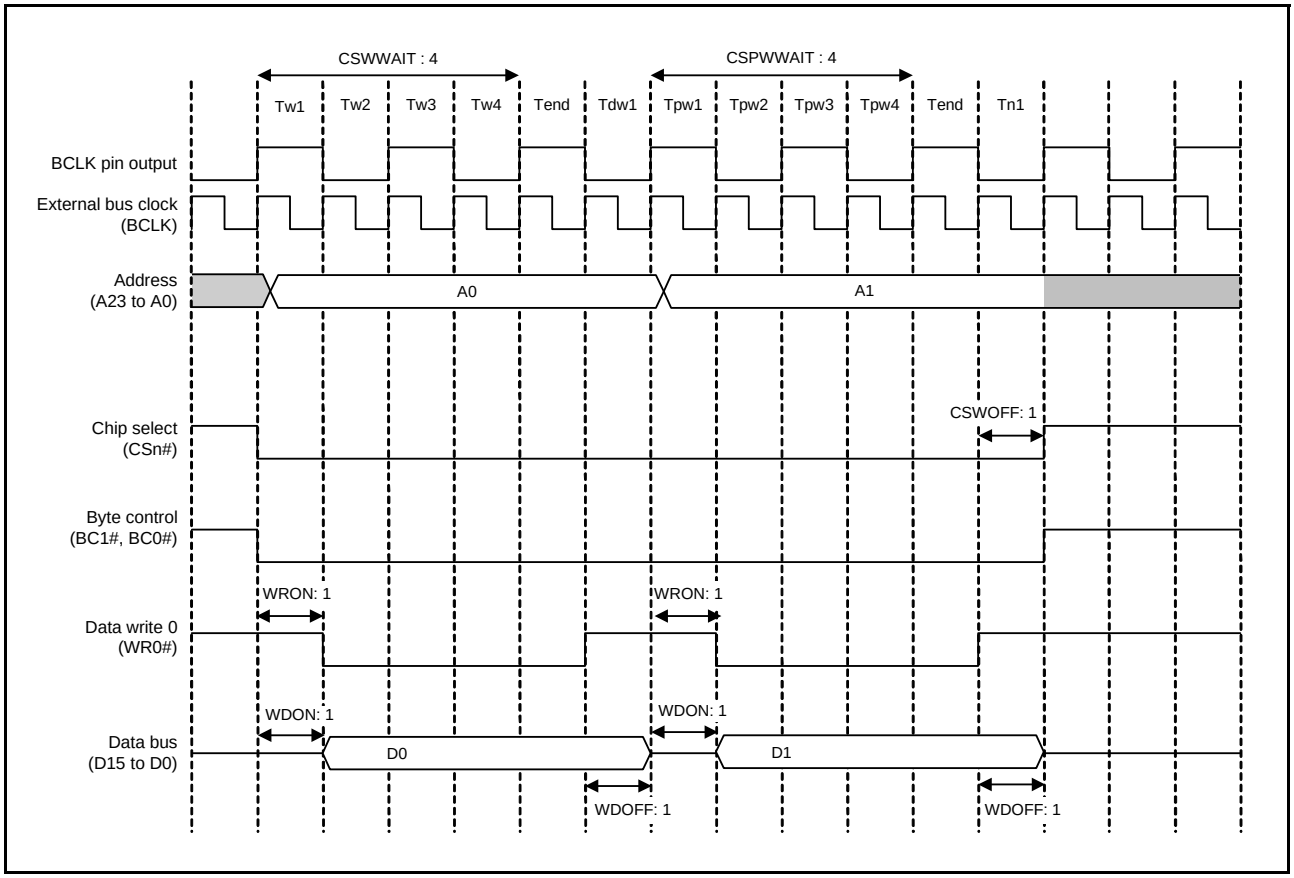


Figure 15.22 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 3)

15.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexingly input/output to/from the D15 to D0 pins in the corresponding area. Using this function enables direct connection of this LSI to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D0 are multiplexed with A7 to A0. When 16-bit width is selected, D15 to D0 are multiplexed with A15 to A0. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD#, WRn#, and BCn# signals.
Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn# signals within the address cycle, the byte-control signal is output for the data being read or written.
During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

- Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and one cycle before the address latch (ALE) signal is negated. The number of cycles is selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS# assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 15.23 to Figure 15.25 show examples of operations with the address/data multiplexed I/O interface.

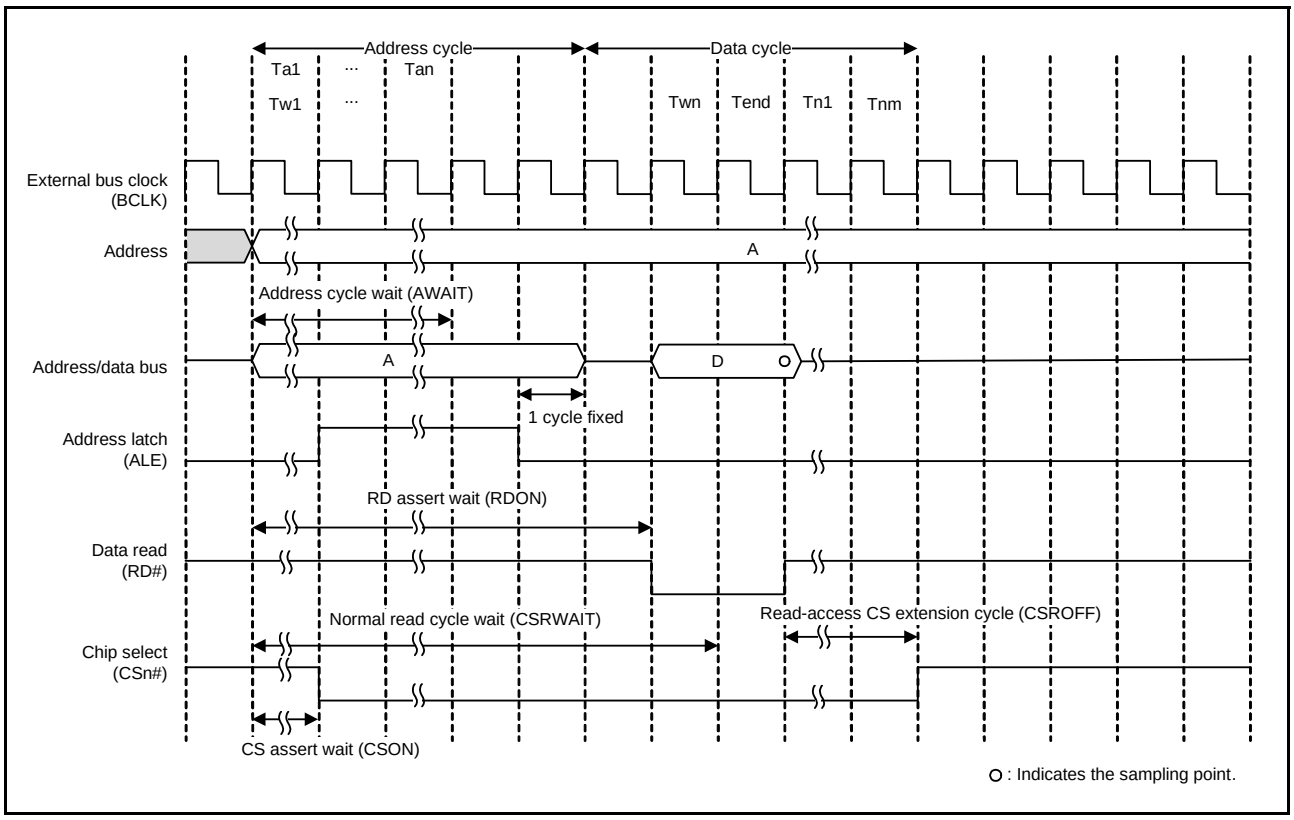


Figure 15.23 Example of Read Access Operation (with Address/Data Multiplexed I/O Interface) (n = 0 to 3)

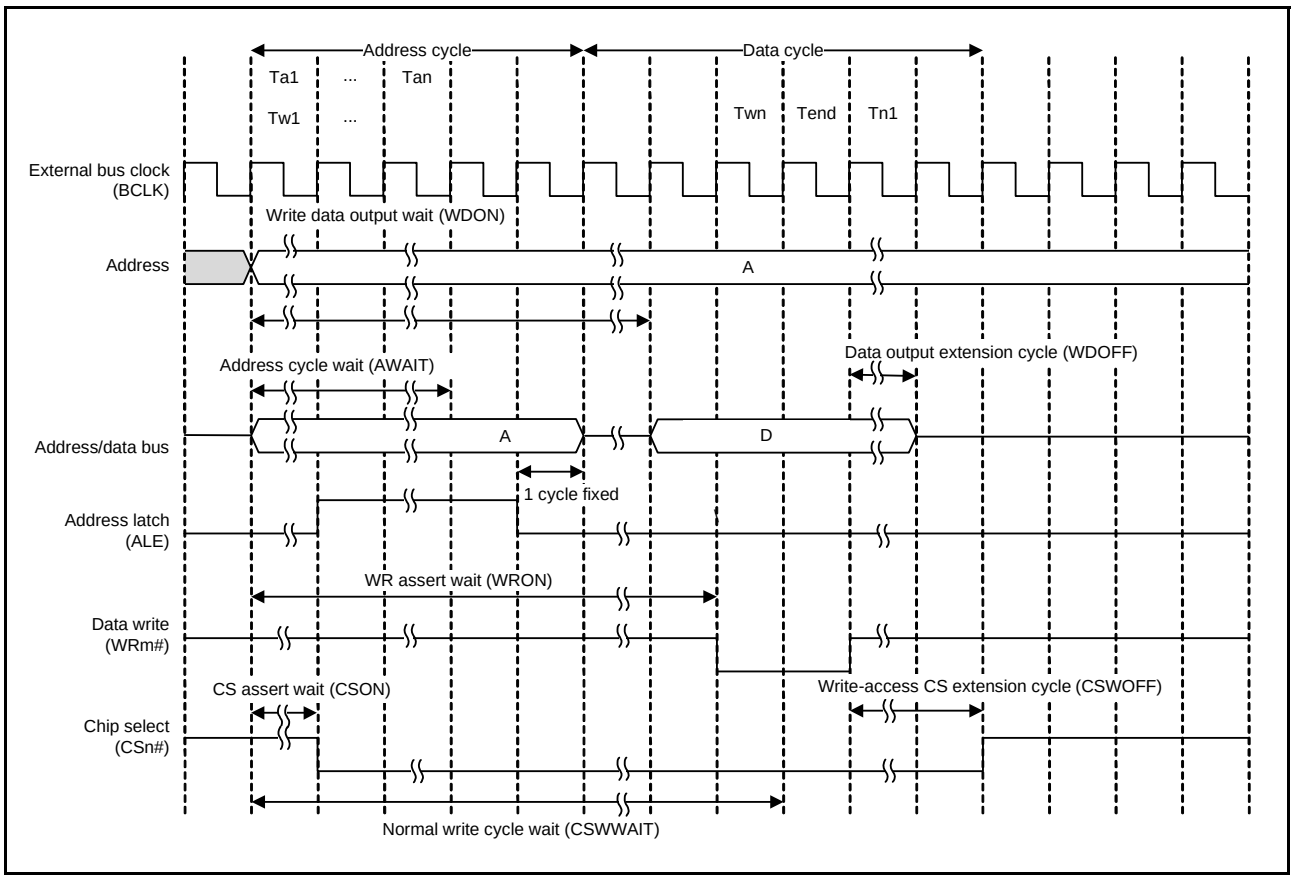


Figure 15.24 Example of Write Access Operation (with Address/Data Multiplexed I/O Interface; n = 0 to 3, m = 0, 1)

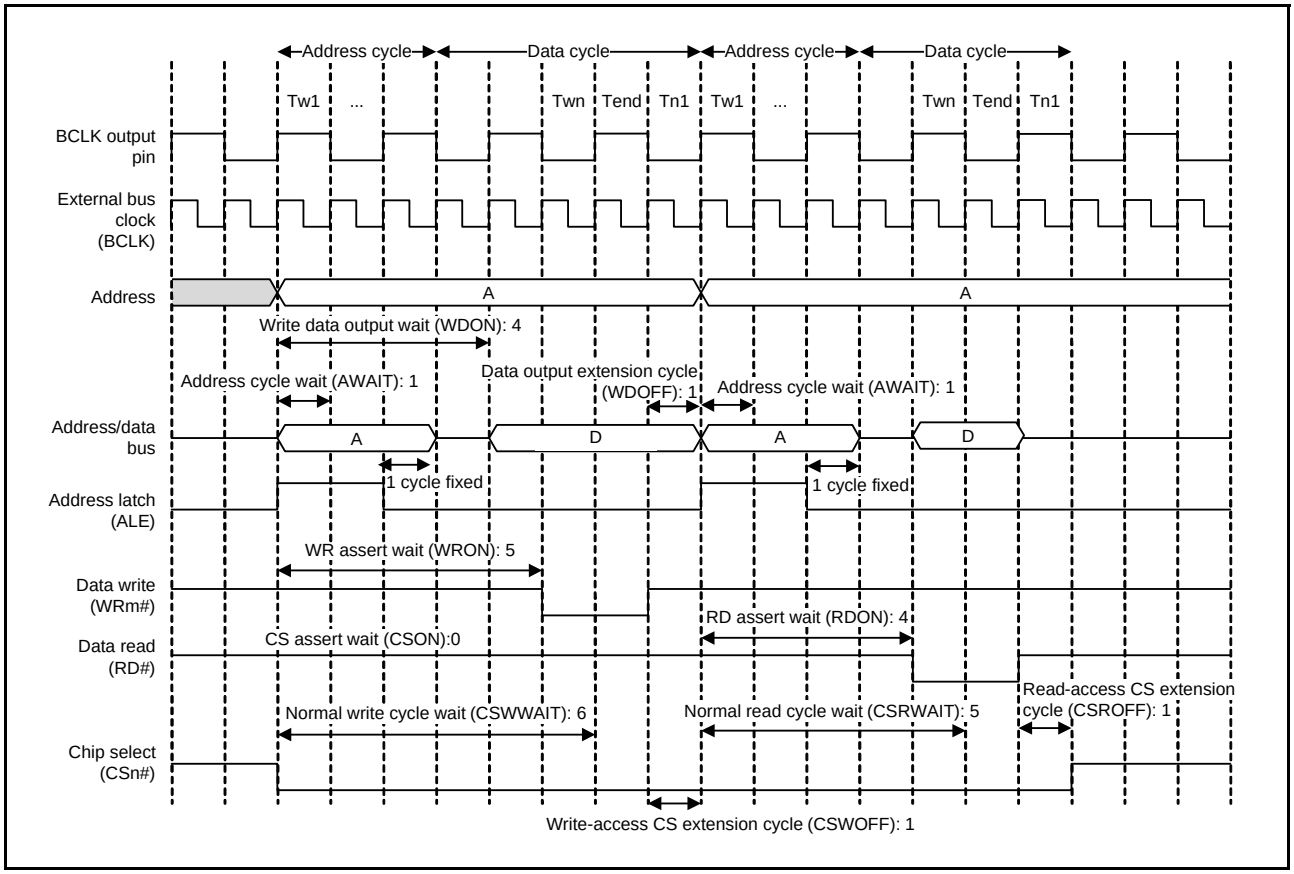


Figure 15.25 Example of Bus Timing (with Address/Data Multiplexed I/O Interface; n = 0 to 3, m = 0, 1)

15.5.3 External Wait Function

Wait cycles can be extended by the WAIT# signal over the length of normal access cycle wait (specified by the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1) and page access cycle wait (specified by the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1).
When external wait is enabled (the EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT# signal is held low. When external wait is disabled (the EWENB bit = 0 in CSnMOD), the WAIT# signal has no effect.
All wait cycles specified in CSnWCR1 are inserted independently of the WAIT# signal.

(1) Normal Access

Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.
With respect to the second and subsequent accesses, sampling of the WAIT# signal begins upon completion of the wait cycle of the page access (Tend). The wait cycle of the page access is extended while the WAIT# signal is held low, and ends (Tend) at the next cycle after the WAIT# signal becomes high.
Figure 15.26 and Figure 15.27 show examples of external wait insertion timing with the separate bus interface.

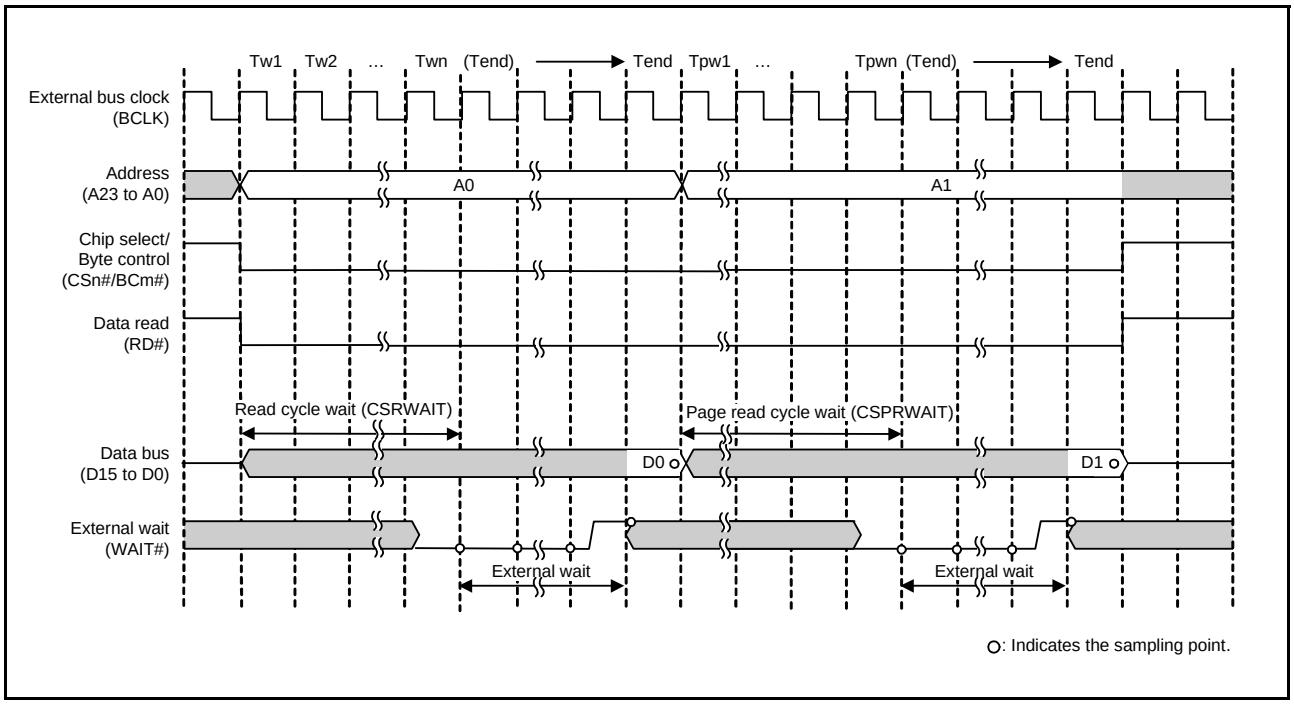


Figure 15.26 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 3, m = 0, 1)

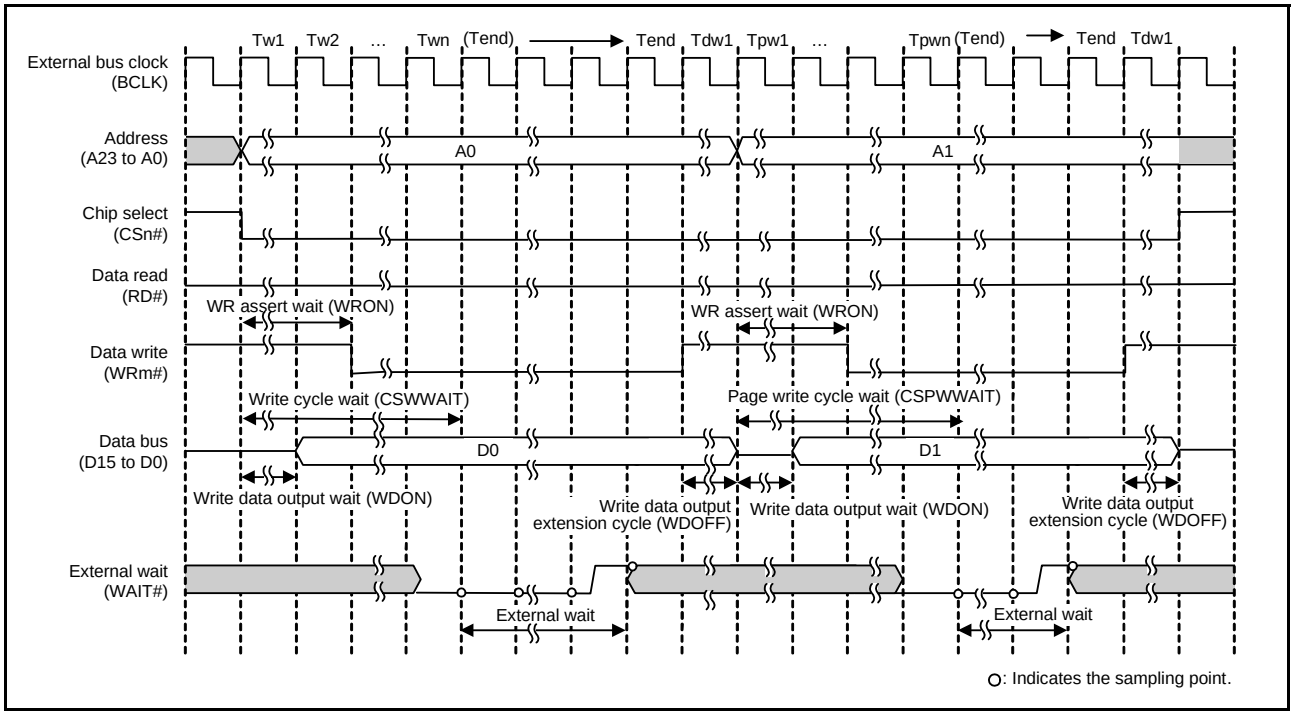


Figure 15.27 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 3, m = 0, 1)

(3) Address/Data Multiplexed I/O Interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 15.28 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

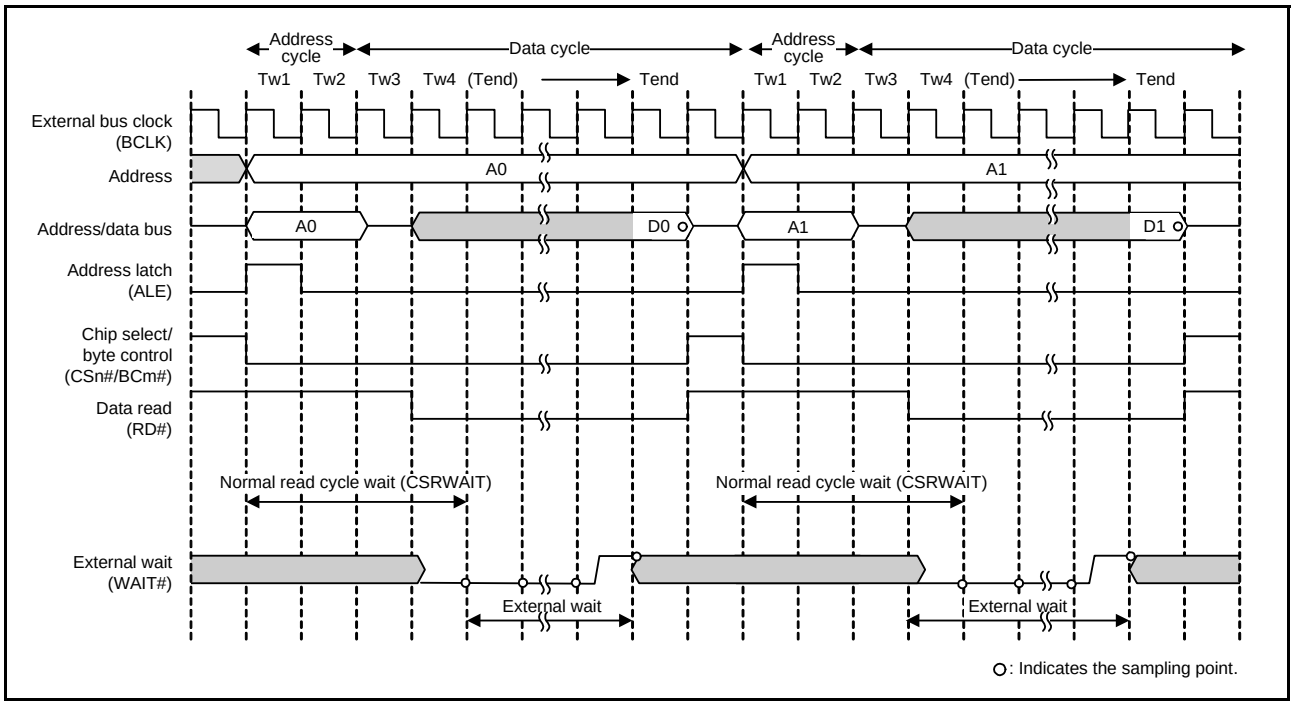


Figure 15.28 Example of External Wait Insertion Timing (with Address/Data Multiplexed I/O Interface; n = 0 to 3, m = 0, 1)

15.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the recovery cycle insertion enable bit in CSRECEN to 1.
The number of recovery cycles to be inserted after read cycles and write cycles can be separately set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles should be set with the WRCV[3:0] bits for the area. When the preceding bus cycle is a read access, the number of read recovery cycles should be set with the RRCV[3:0] bits for the area. For example, when CS1 read access occurs after CS0 read access, the number of recovery cycles to be inserted between them is set by the RRCV[3:0] bits in CS0REC.
Recovery cycles can be inserted on any of the following eight conditions. The recovery cycle insertion can be enabled or disabled with the RCVENj (j = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, i.e. when the CSn# signal (n = 0 to 3) is negated. A high-level period of the CSn# signal is inserted for the specified recovery cycle period starting from this point.
The CSn# signal for the next round of bus access is asserted immediately after the end of recovery cycles in the fastest case. Even if the next request for access to an external address space is generated during the recovery period, the next round of access over the external bus will start immediately after the end of recovery cycles.
When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer even if the recovery cycle insertion condition is satisfied (Figure 15.31). Similarly, during normal accesses with page access enabled, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer. Similarly, during normal accesses with page access enabled, with the separate bus interface, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer. With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.
Figure 15.29 to Figure 15.31 show examples of recovery cycle insertion with the separate bus interface.

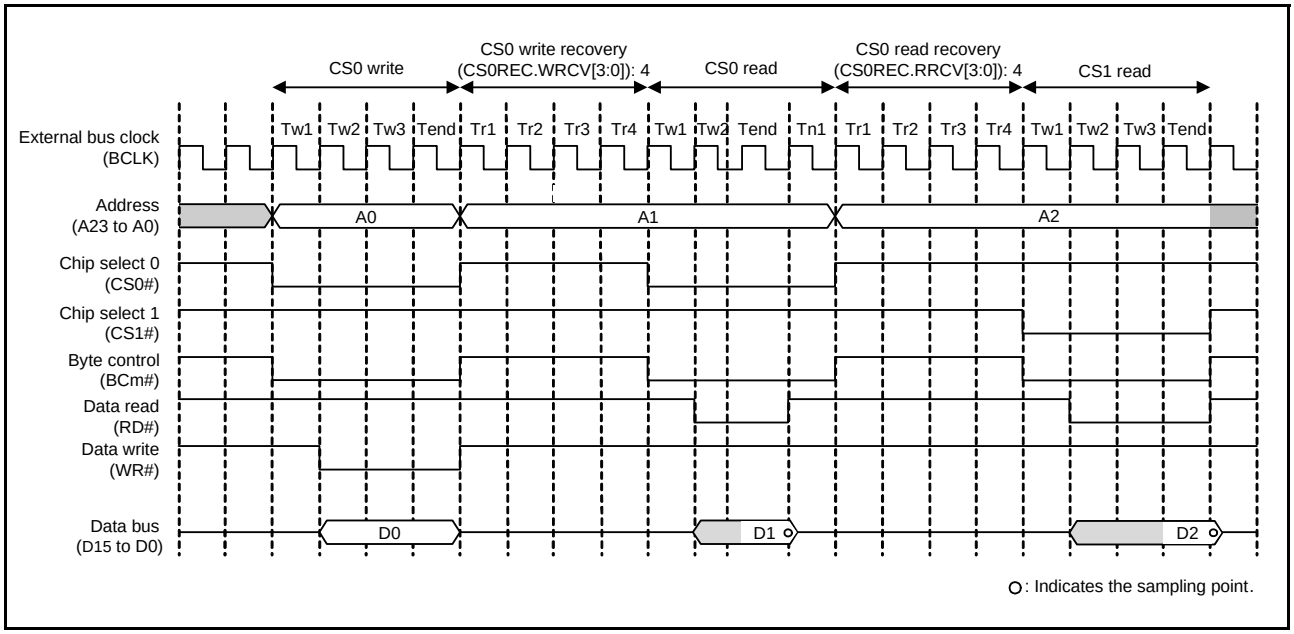


Figure 15.29 Example of Recovery Cycle Insertion (with Separate Bus Interface) (m = 0, 1)

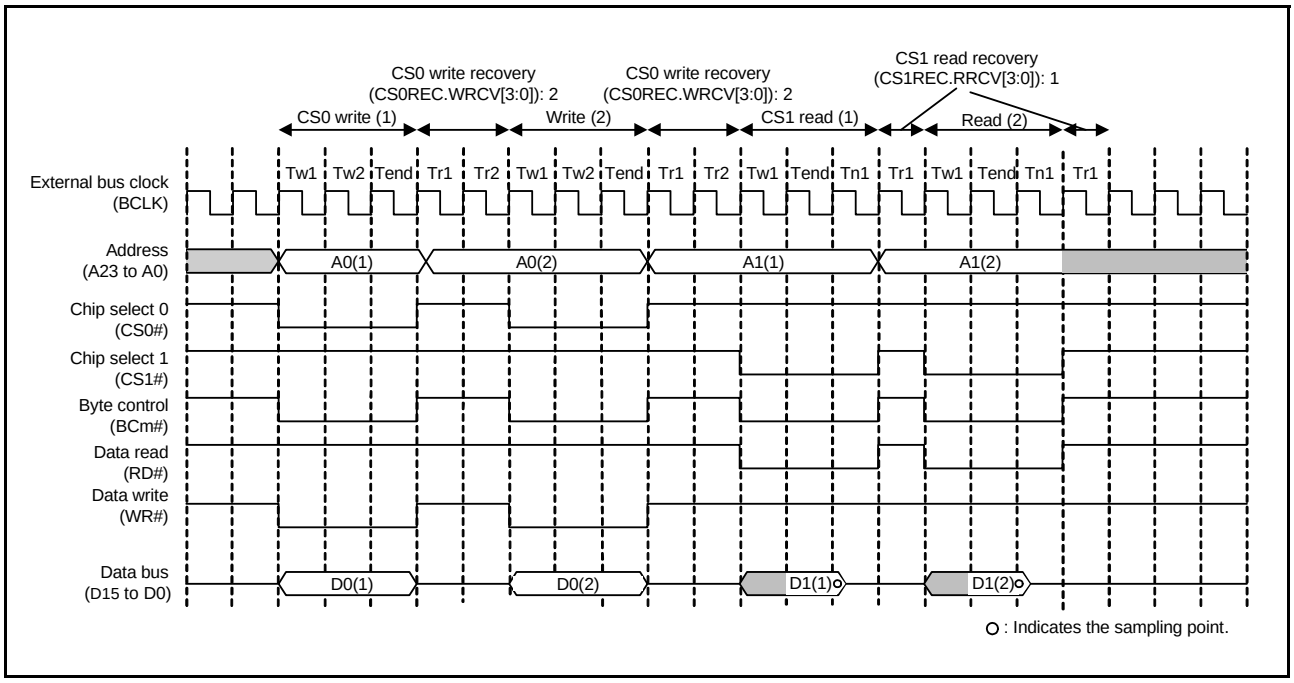


Figure 15.30 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0, 1)

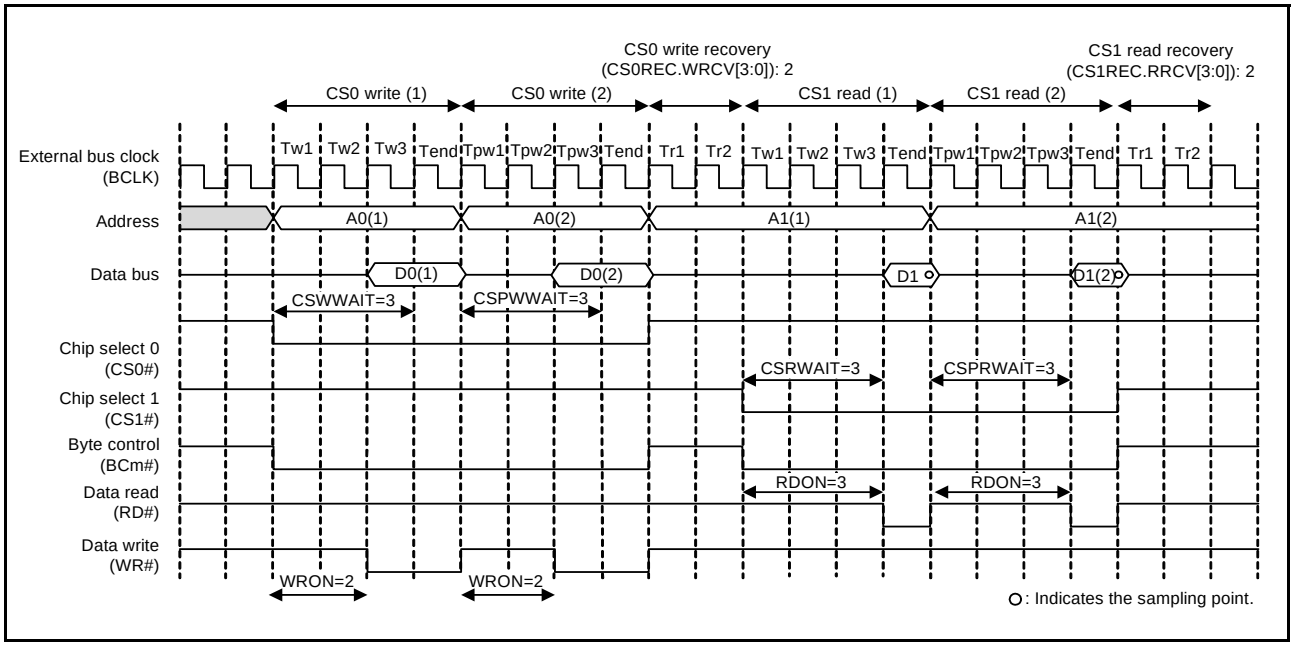


Figure 15.31 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0, 1)

Figure 15.32 shows examples of operations when the BCLK pin output select bits are set for frequency-division of BCLK by 2.

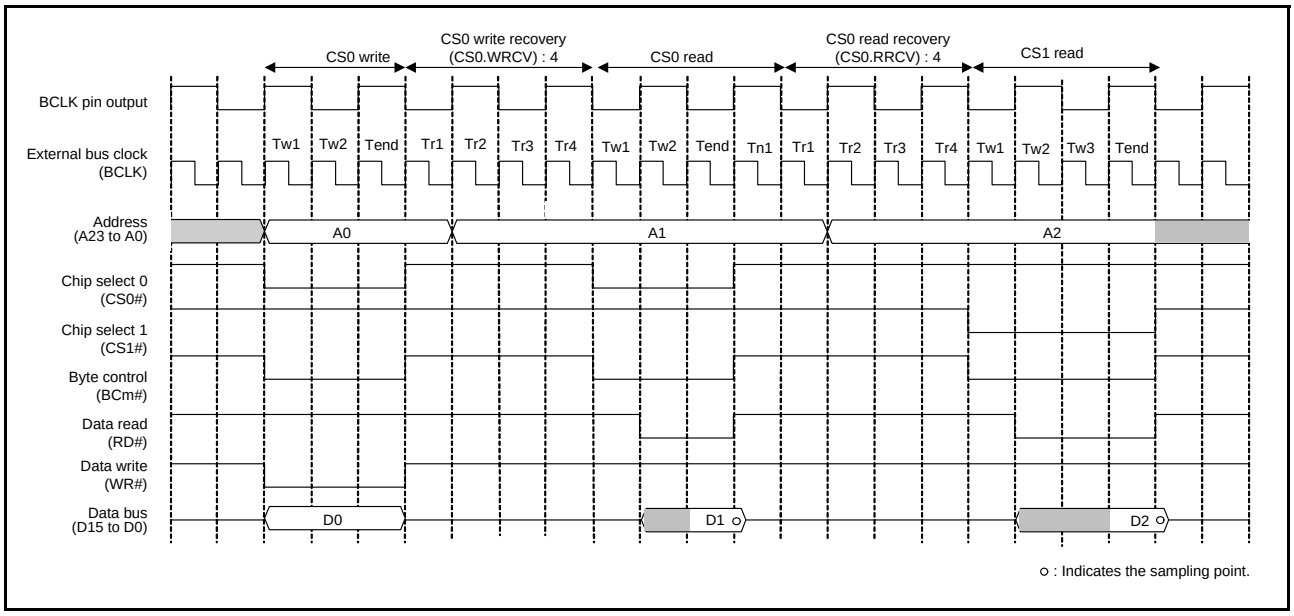


Figure 15.32 Example of Operation for Recovery Cycles when the BCLK Pin Output Select Bits Are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access Through a Separate Bus Interface; m = 0, 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 15.33 and Figure 15.34 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

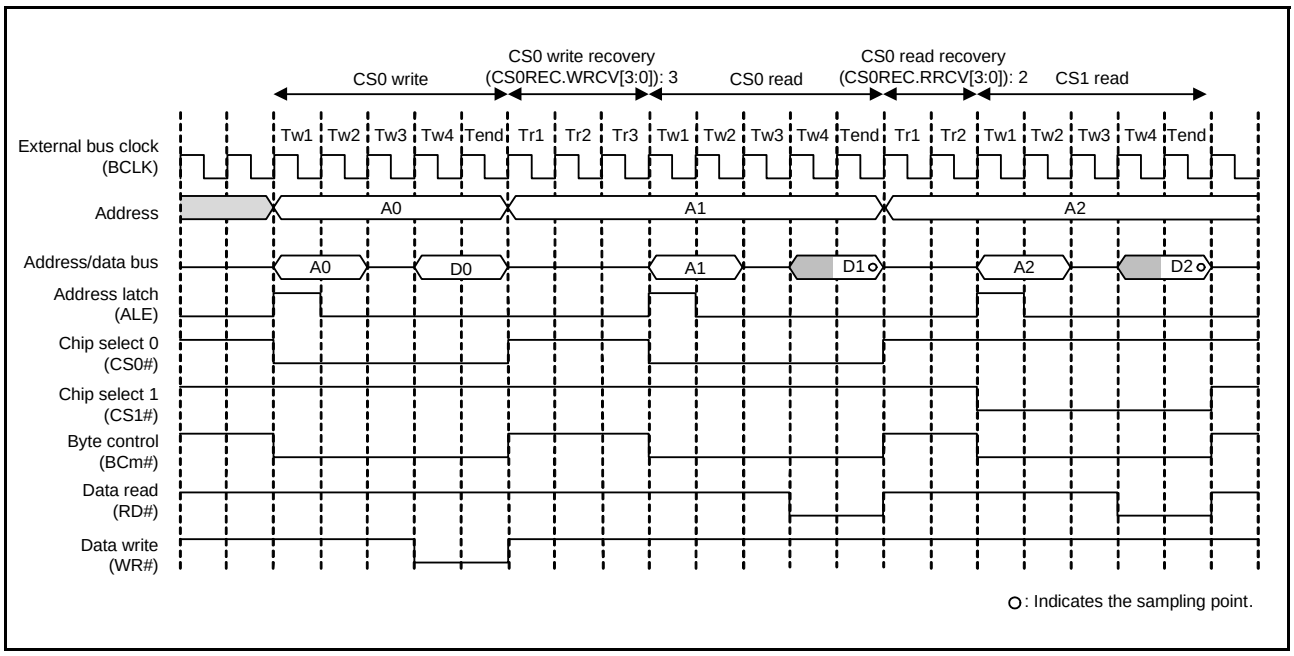


Figure 15.33 Example of Recovery Cycle Insertion (with Address/Data Multiplexed I/O Interface; m = 0, 1)

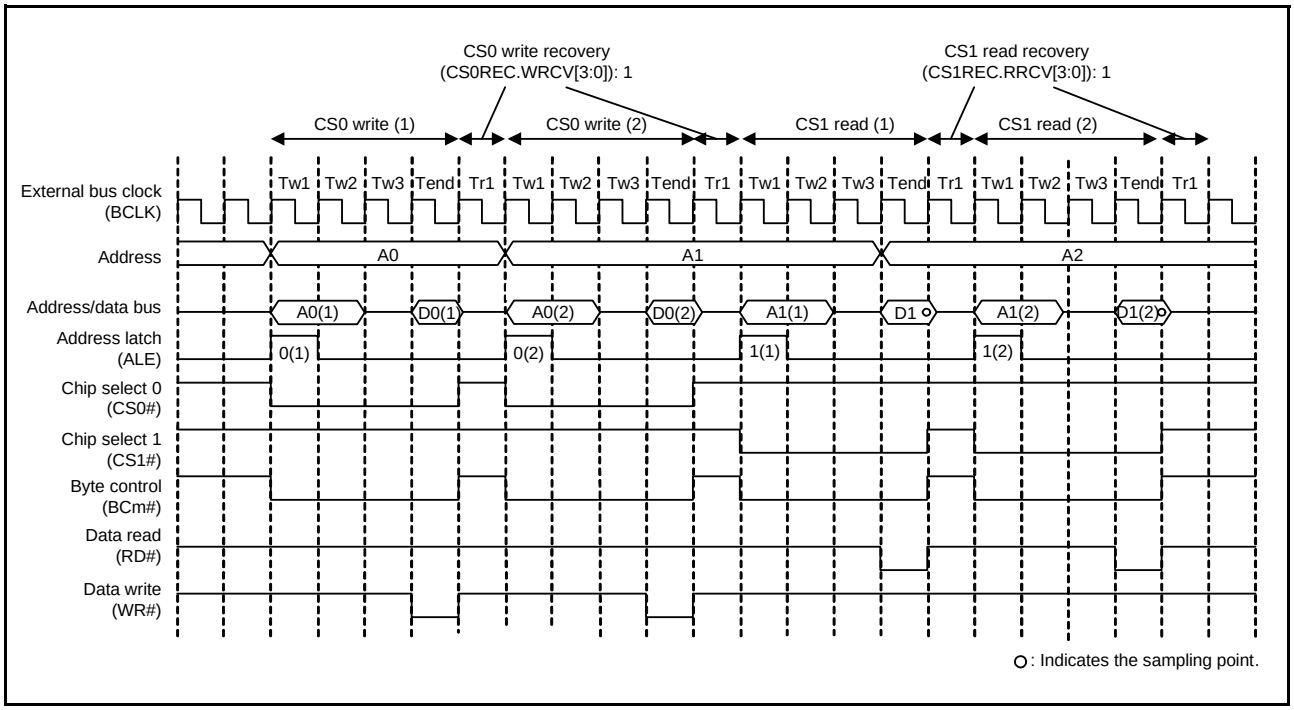


Figure 15.34 Example of Recovery Cycle Insertion When a Bus Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1)

15.5.5 No Access State

Table 15.9 lists the states of external bus signals when no external address space is accessed.

Table 15.9 States of External Bus Signals during No Access

Signal Line Name	Signal State	
	Separate Bus	Address/Data Multiplex Bus
CSn#, BCn#, WRn#, RD#	High	
ALE	Low	
A23 to A16	Undefined	
A15 to A0	Undefined	Hi-Z
D15 to D0	Hi-Z	

15.5.6 Write Buffer Function (External Bus)

The internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed. Figure 15.35 shows an example of operation when the write-buffer function is in use. When this function is in use, if the next operation after an external write is internal access, the internal access (access to on-chip memory or a peripheral module) is executed in parallel with the external write, i.e. without waiting for completion of the latter operation.

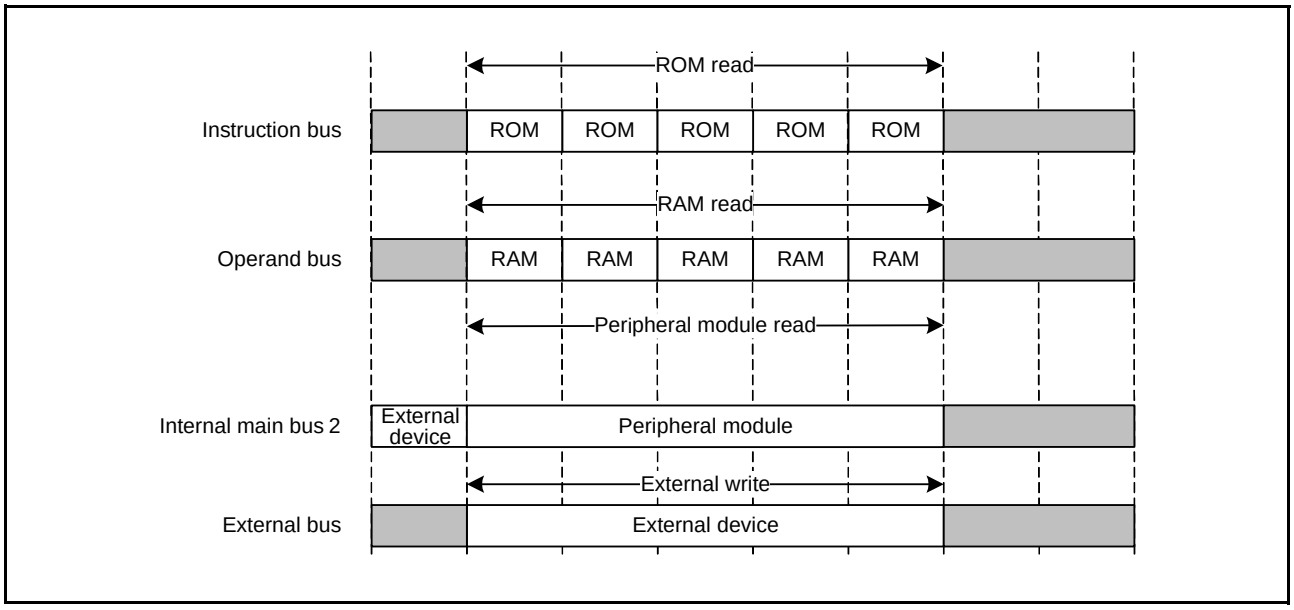


Figure 15.35 Example of Operation when the Write-Buffer Function is in Use

15.6 Limitations

15.6.1 Limitations on Using Separate Bus Interface

(1) Limitations that apply to various bits of CSn wait control register 1 (CSnWCR1) and CSn wait control register 2 (CSnWCR2) at the times of normal and page accesses are listed in Table 15.10.
Even if the setting of the page-read access enable bit in the CSn mode register or the page-write access enable bit in the CSn mode register selects permission (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first round of access for page access and the access that does not fall within the scope of page access are normal access operation, and thus limitations on normal access must be satisfied.

Table 15.10 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

(2) When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.

15.6.2 Limitations on Using Address/Data Multiplexed Bus Interface

(1) In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed.

Table 15.11 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access	
Reading	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON AWAIT[1:0]+2≤RDON CSON[2:0]≤AWAIT	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON AWAIT[1:0]+2≤WRON AWAIT[1:0]+2≤WDON CSON[2:0]≤AWAIT

15.6.3 Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, setting the single write strobe mode is prohibited in the 8-bit bus space; otherwise the operation is not guaranteed.

15.6.4 Limitations when 1/2 BCLK is Selected with BCLK Pin Output Select Bit

When 1/2 BCLK is selected through the BCLK pin output select bit, the external bus access cycle starts at the rising edge of the BCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle may start at the falling edge of the BCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

15.6.5 Prohibition of Access that Spans Areas of Address Space

Single access that spans several 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.

15.6.6 Restrictions on RMPA and String-Manipulation Instructions

(1) 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 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) 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.

15.6.7 Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

15.7 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.

15.7.1 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.

15.7.1.1 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 of the following types leads to illegal address access errors.

- Access to areas of external space for which operation has been disabled (CSnCR.EXENB = 0)
- With respect to areas other than those described above, access to illegal address ranges
  The address ranges where access will lead to illegal address access errors are indicated in Table 15.12.

15.7.1.2 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.

- CS areas (CS0 to CS3): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.

15.7.2 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.

15.7.3 Conditions Leading to Bus Errors

Table 15.12 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, 2) is cleared), the detected error is reflected in the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected in 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 15.12 Types of Bus Errors

Address	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	On-chip ROM enabled	On-chip ROM disabled	On-chip ROM enabled	On-chip ROM disabled	On-chip ROM enabled	On-chip ROM disabled
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	Reserved area		Δ		—	
000E 0000h to 000F FFFFh	Reserved area		Δ		—	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved area	Δ	○	—	—
0500 0000h to 07FF FFFFh	External bus (CS1 to CS3)		[IA]		[TO]	
0800 0000h to 0FFF FFFFh	Reserved area		—		—	—
1000 0000h to 7FFF FFFFh	Reserved area		○		—	—
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	—	○	—	—
FF00 0000h to FF7F FFFFh		External bus (CS0)	—	[IA]	—	[TO]
FF80 0000h to FFFF FFFFh			—		—	

—: A bus error does not result.

Δ: A bus error may or may not result.

○: A bus error results.

[IA]: Access to this area leads to detection of a bus error if operation for this area is disabled (CSnCR.EXENB = 0; n = 0 to 3).

[TO]: Bus access not being completed within 768 cycles leads to detection of a bus error.

Note: The capacity of the RAM, E2 DataFlash, and ROM differs depending on the product. For details, see section 39, RAM and section 40, Flash Memory.

16. Memory-Protection Unit (MPU)

16.1 Overview

The RX 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

Specifictaions	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 start-ing 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.

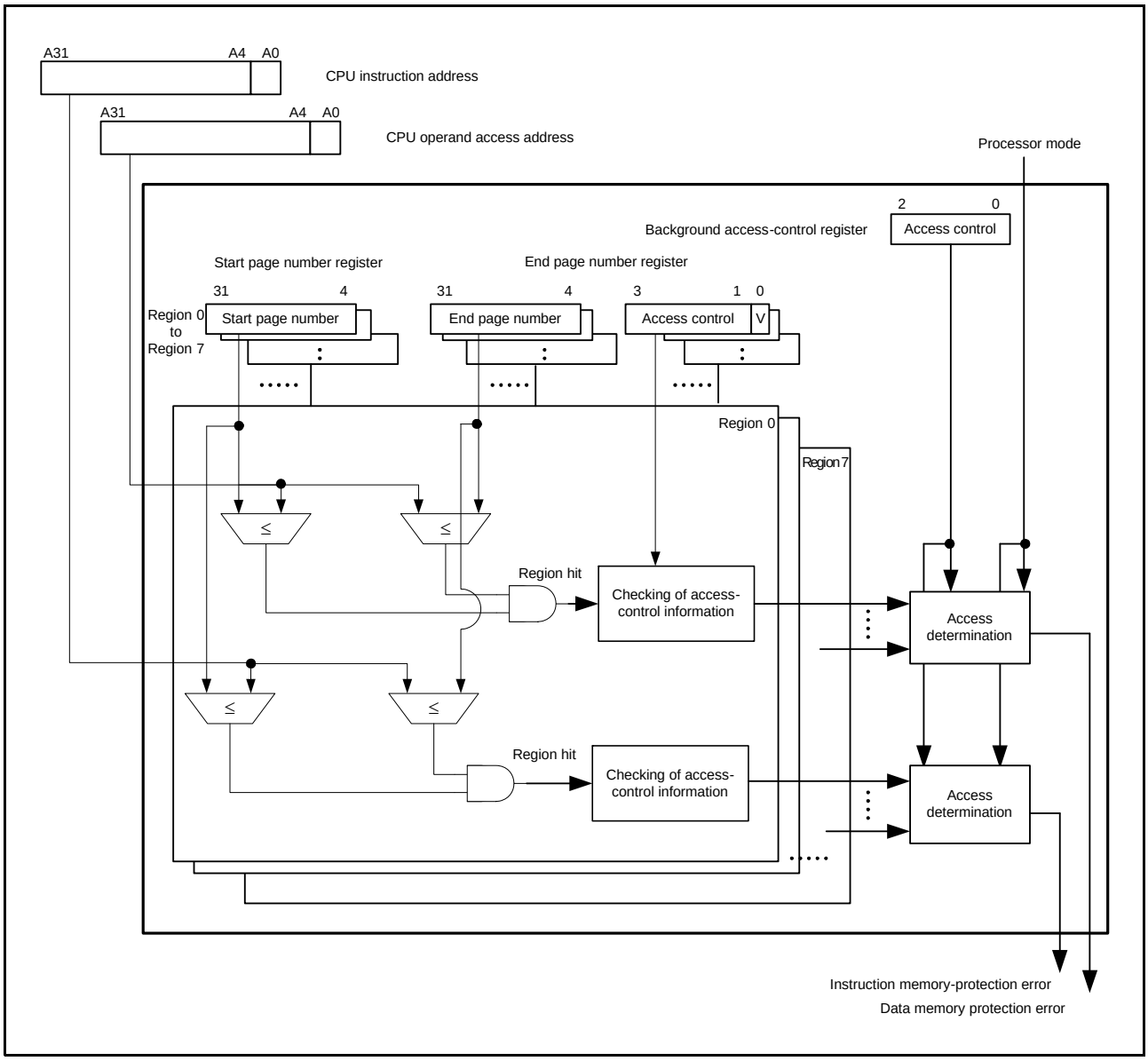


Figure 16.1 Block Diagram of the Memory-Protection Unit

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)

Addresses: 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	Function	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)

Addresses: 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	Function	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 cleared 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: 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	Function	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled. 0: The memory protection is disabled.	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 and RTFI) that shifts operation to the user mode.

16.2.4 Background Access Control Register (MPBAC)

Address: 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	Function	R/W
b0	—	Reserved	These bits are 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: 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	Function	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 the MPESTS are cleared 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 generated bit (DMPER), and the instruction memory-protection error generated bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

16.2.6 Memory-Protection Error Status Register (MPESTS)

Address: 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	Function	R/W
b0	IMPER	Instruction Memory-Protection Error Generated Bit	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DMPER	Data Memory-Protection Error Generated Bit	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write Bit	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 Generated Bit)

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 Generated Bit)

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 Bit)

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 data memory-protection error generated bit (DMPER) 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: 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	Function	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: 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	Function	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: 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	Function	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: 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	Function	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 cleared to 0, all settings other than background access-control settings are invalid.

16.2.11 Instruction-Hit Region Register (MHITI)

Address: 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	Function	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: 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	Function	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

Bit	Symbol	Bit Name	Function	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 cleared 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

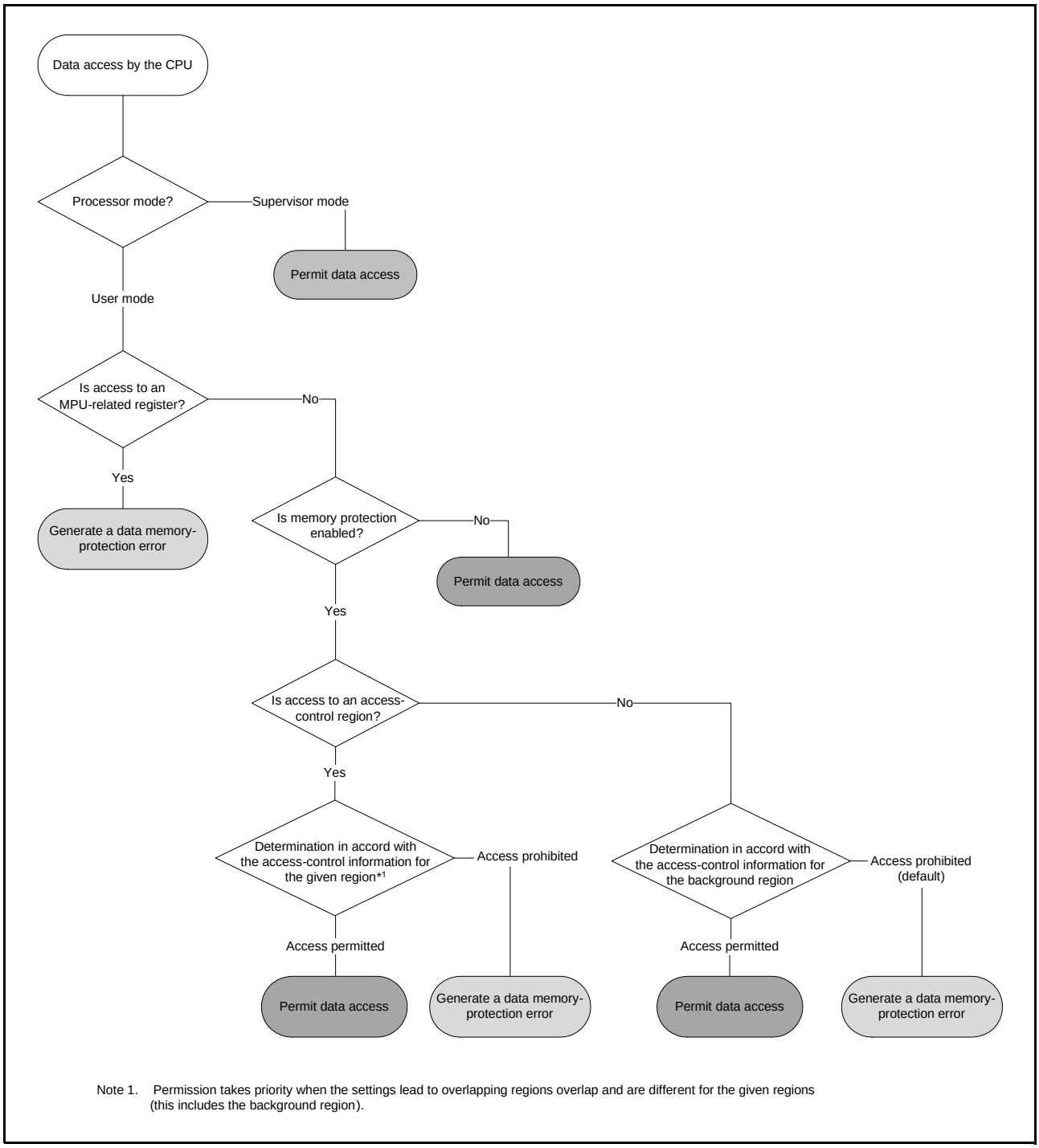
Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOP) register is set to 1, the address specified 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. 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.

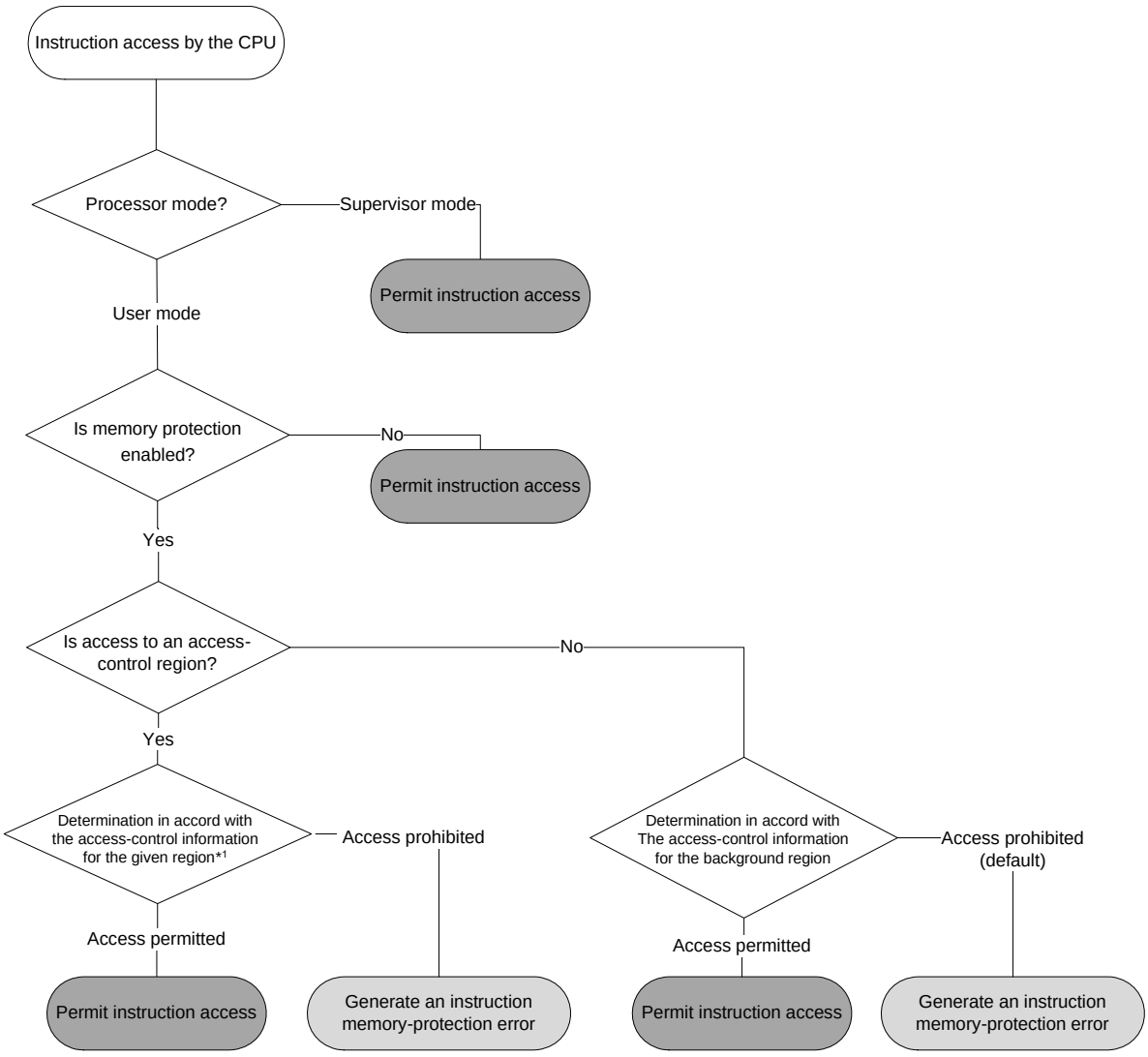
16.3.4 Flow for Determination of Access by the Memory-Protection Function

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.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are different for the given regions (this includes the background region).

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, be sure to read the registers for which writing was performed and check that the settings have been made as the final step 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 generated (IMPER) and data memory-protection error generated (DMPER) bits in the memory-protection error status (MPESTS) register from within the exception-processing routine.
After confirming the type of error, clear the memory-protection error status (MPESTS) register by writing 1 to the status clearing (MPE) bit in the memory-protection error status clearing (MPECLR) register.

(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 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 (MHITD.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 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 (DMACA)

This MCU incorporates a 4-channel direct memory access controller (DMAC).
The DMAC module performs data transfers 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		4 (DMACm (m = 0 to 3))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1M data (Maximum number of transfers in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Activation 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 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 on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Power consumption reduction function		Module stop state can be set.
Event link function		Event link request is generated after one data transfer (for block, after one block transfer).

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

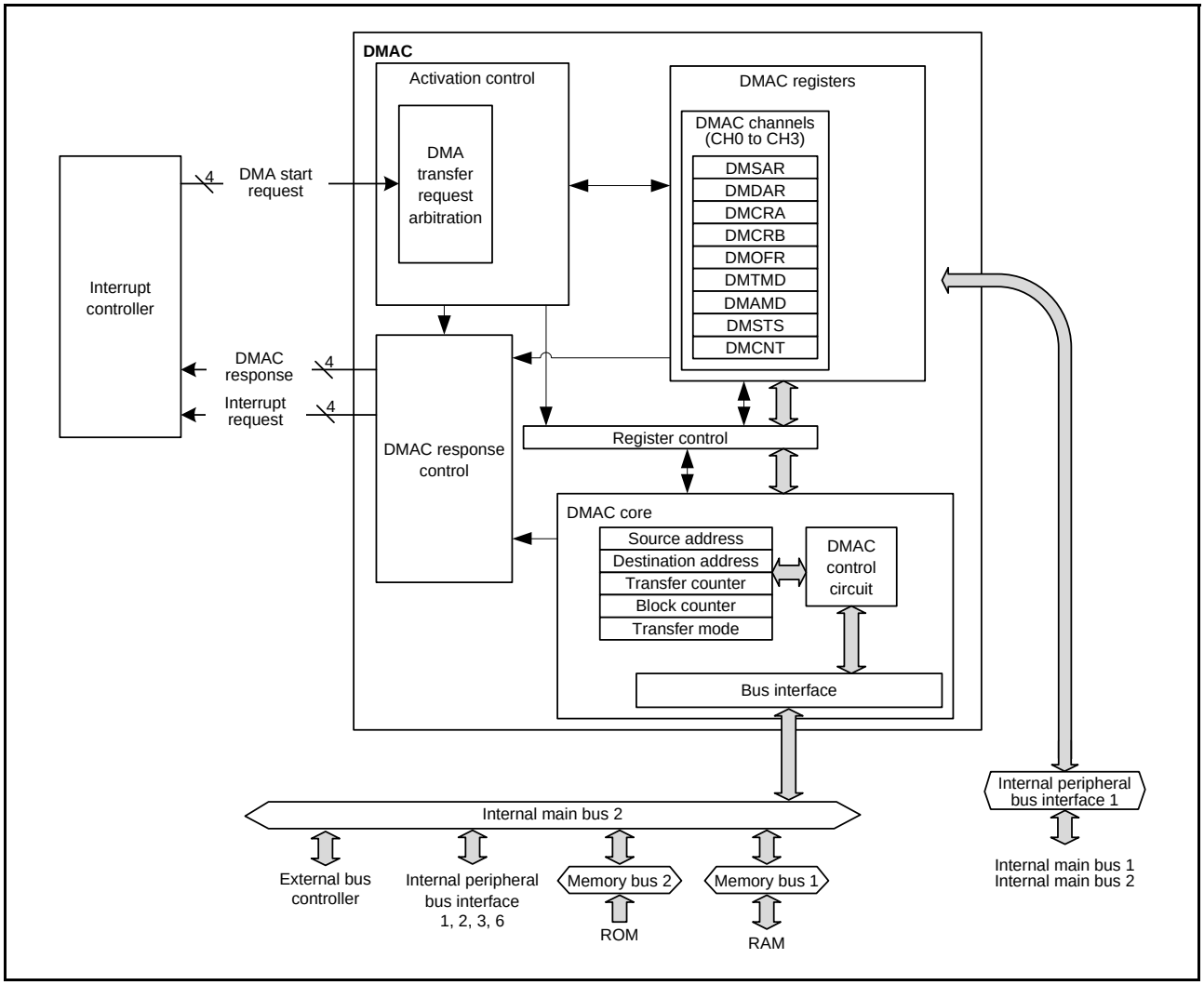


Figure 17.1 Block Diagram of DMAC

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

	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 activation is disabled (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

	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

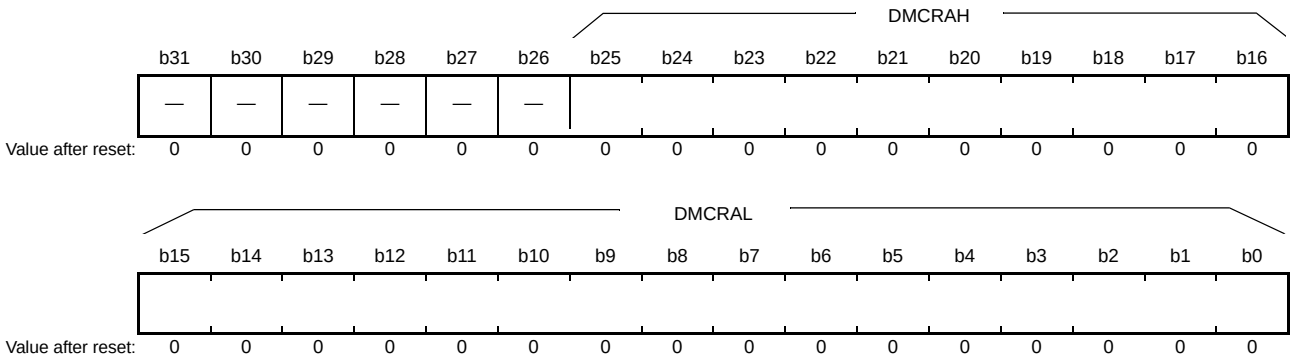
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

DMDAR specifies the start address of the transfer destination. Set DMDAR while DMAC activation is disabled (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.

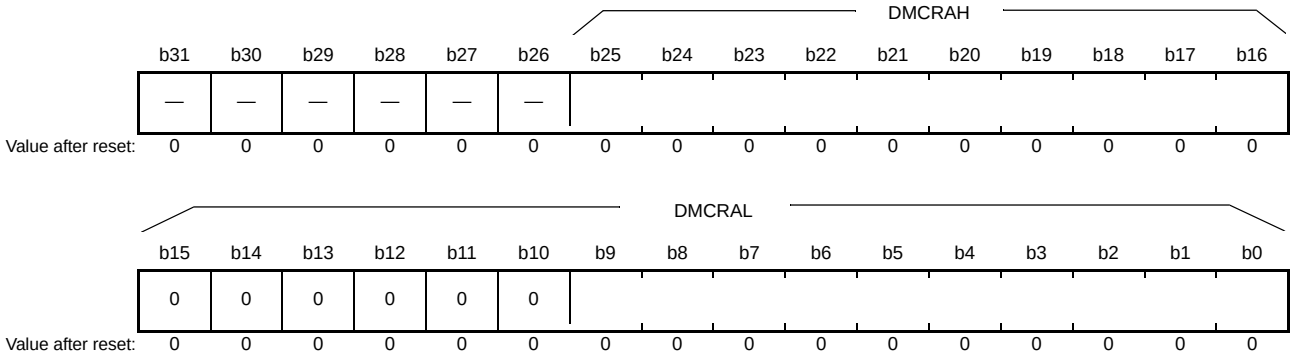
17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



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 65535 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

	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
b9 to b0	Specifies the number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

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, 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

	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]	DMA 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. DMAC activation source is selected using the DMRSRm registers of the ICU. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

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

	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 cleared 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 cleared 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 cleared 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 cleared 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 clearing 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 clearing 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

	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, or 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 21 (2 bytes) and 217 (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, or 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] Bit (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

SARA4 to SARA0 or DARA4 to DARA0	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

	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 activation is enabled) 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

	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 cleared to 0 if the CLRS bit is set to 0. This bit is not cleared 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 activation 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 cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared 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

	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 Activation Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh

	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 flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation 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 flag of the activation source of the DMAC is cleared to 0 or issues an interrupt to the CPU, at the beginning of transfer.

When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.

17.2.13 DMA Module Activation 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 Operation Enable	0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Operation Enable)

When this bit is set to 1, DMAC activation is enabled 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 activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is 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

17.3 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 count	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.

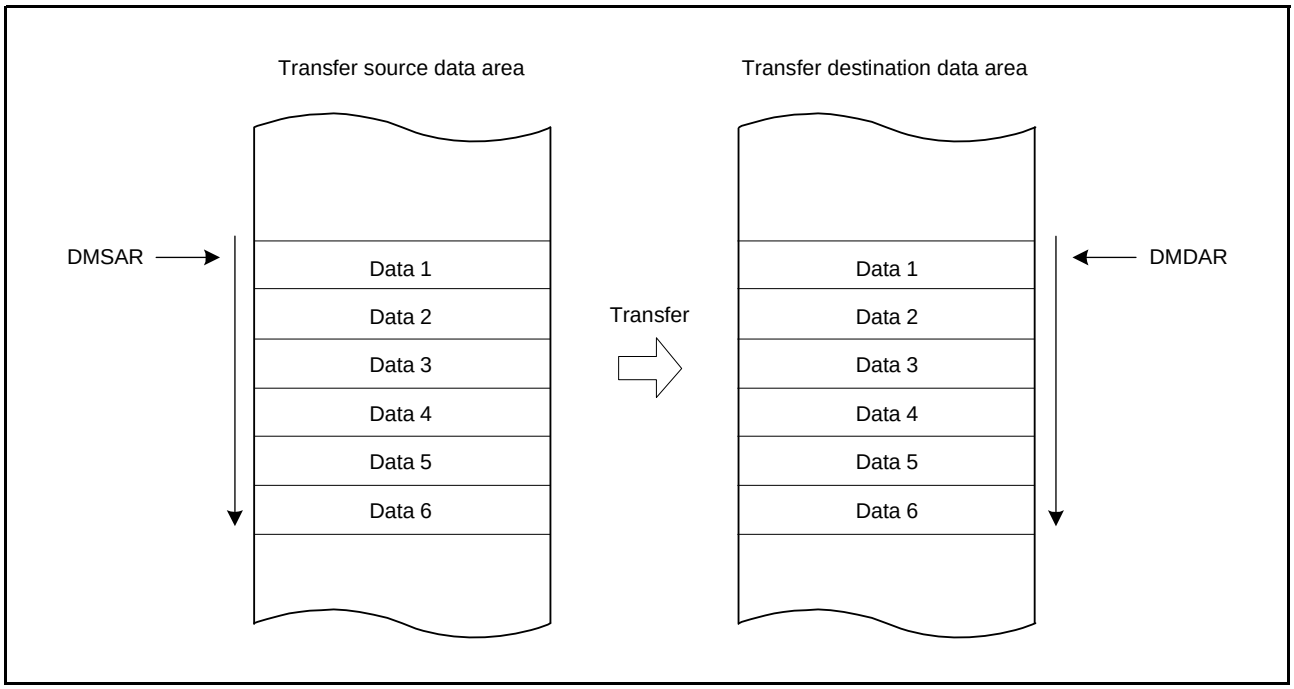


Figure 17.2 Operation in Normal Transfer Mode

(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 1K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count 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 count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

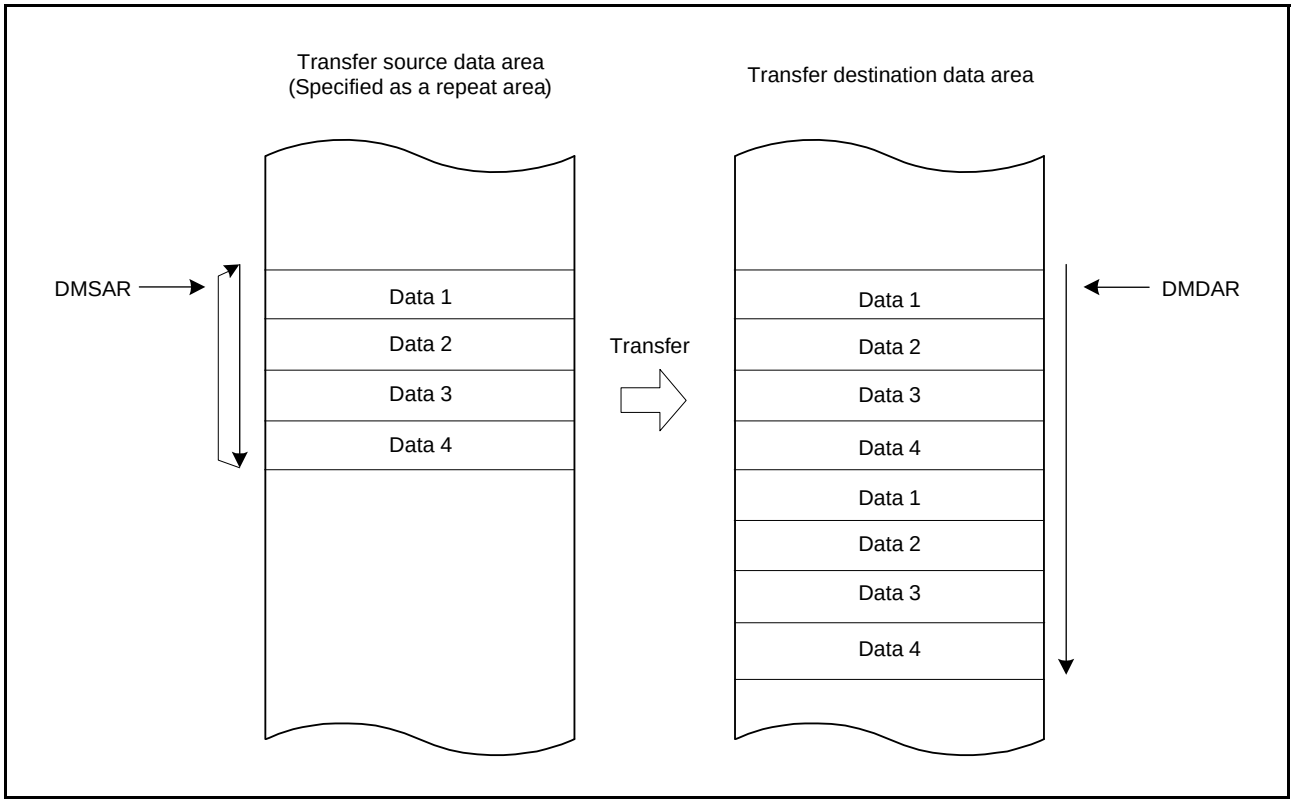


Figure 17.3 Operation in Repeat Transfer Mode

(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 1K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count 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	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

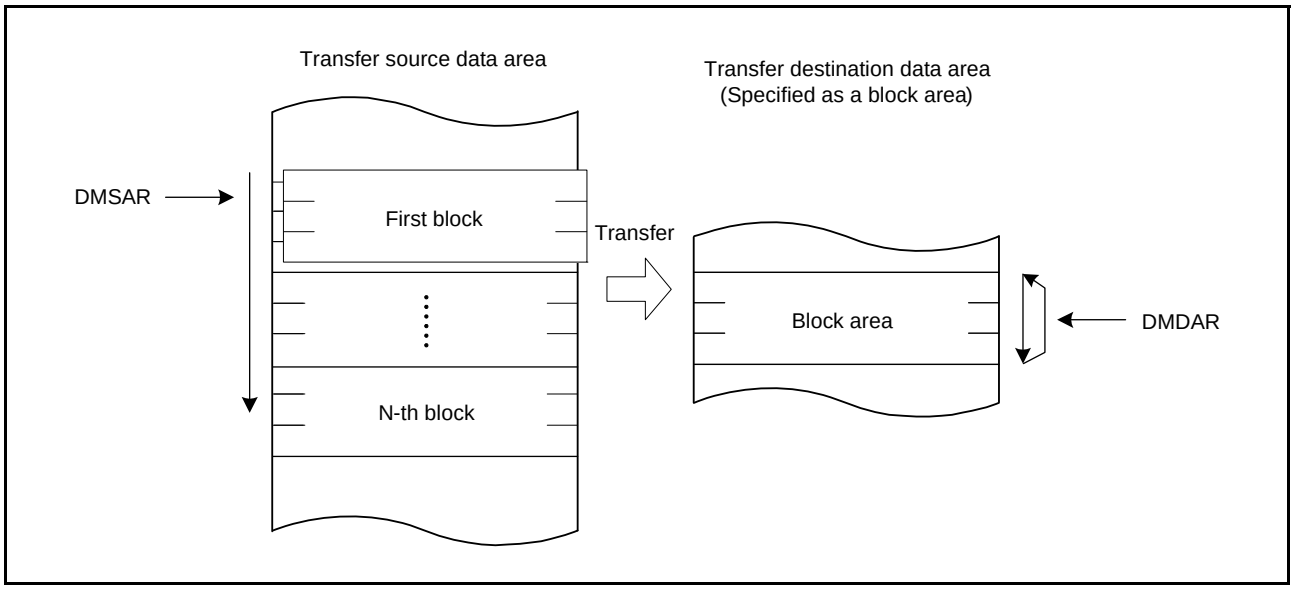


Figure 17.4 Operation in Block Transfer Mode

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 cleared 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.

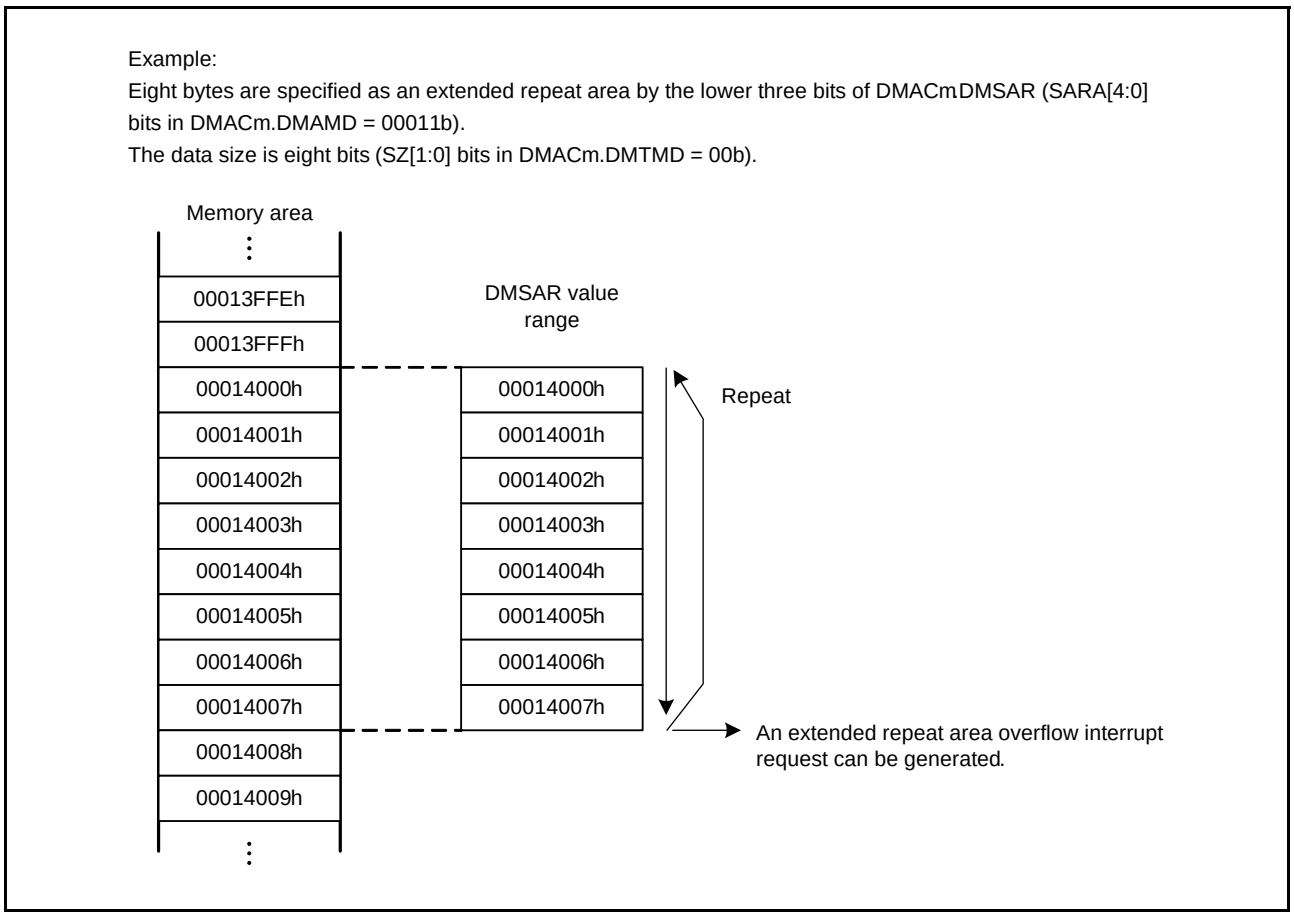


Figure 17.5 Example of Extended Repeat Area Operation

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.

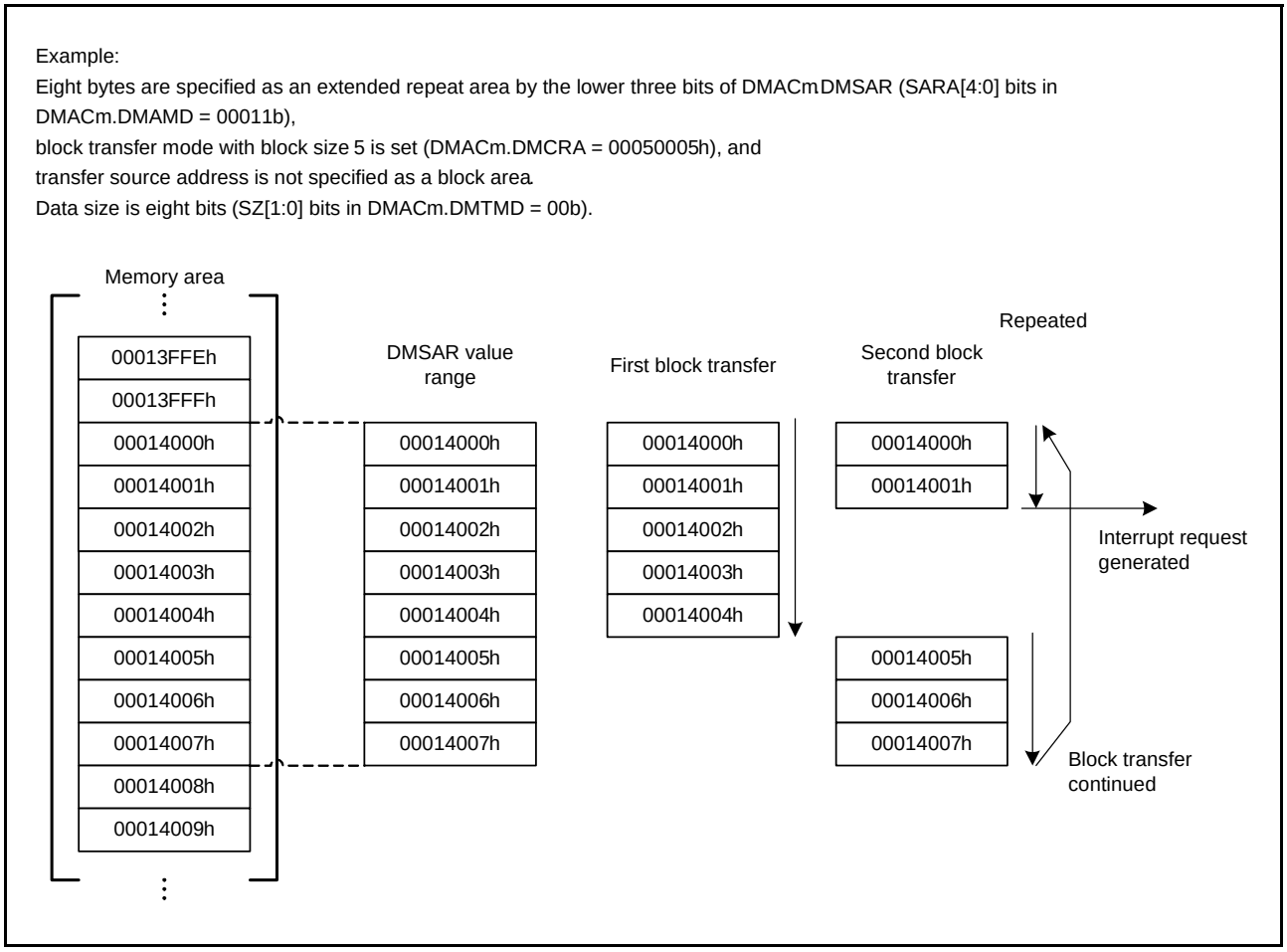


Figure 17.6 Example of Extended Repeat Area Function in Block Transfer Mode

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 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.

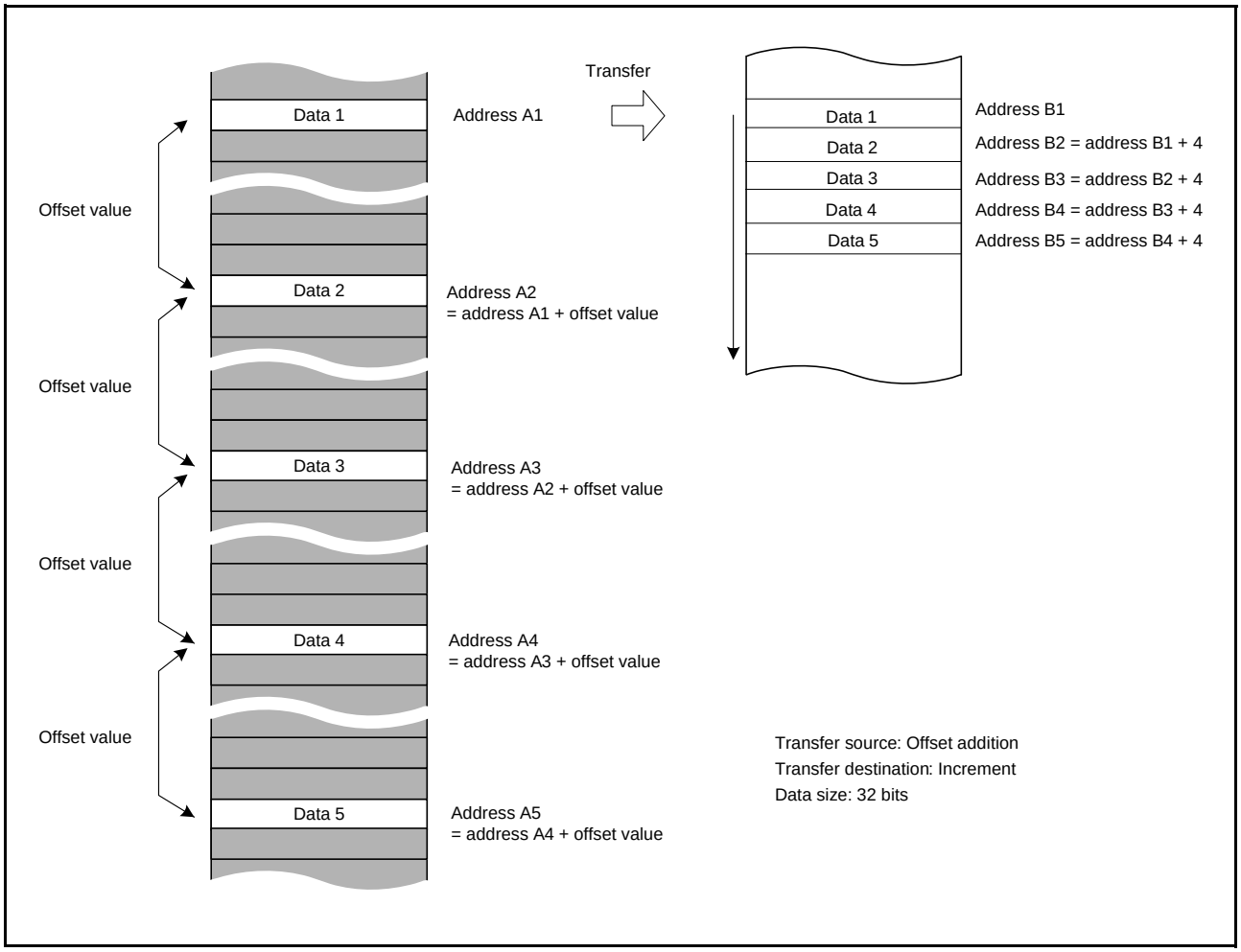


Figure 17.7 Example of Address Updating by Offset Addition

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.

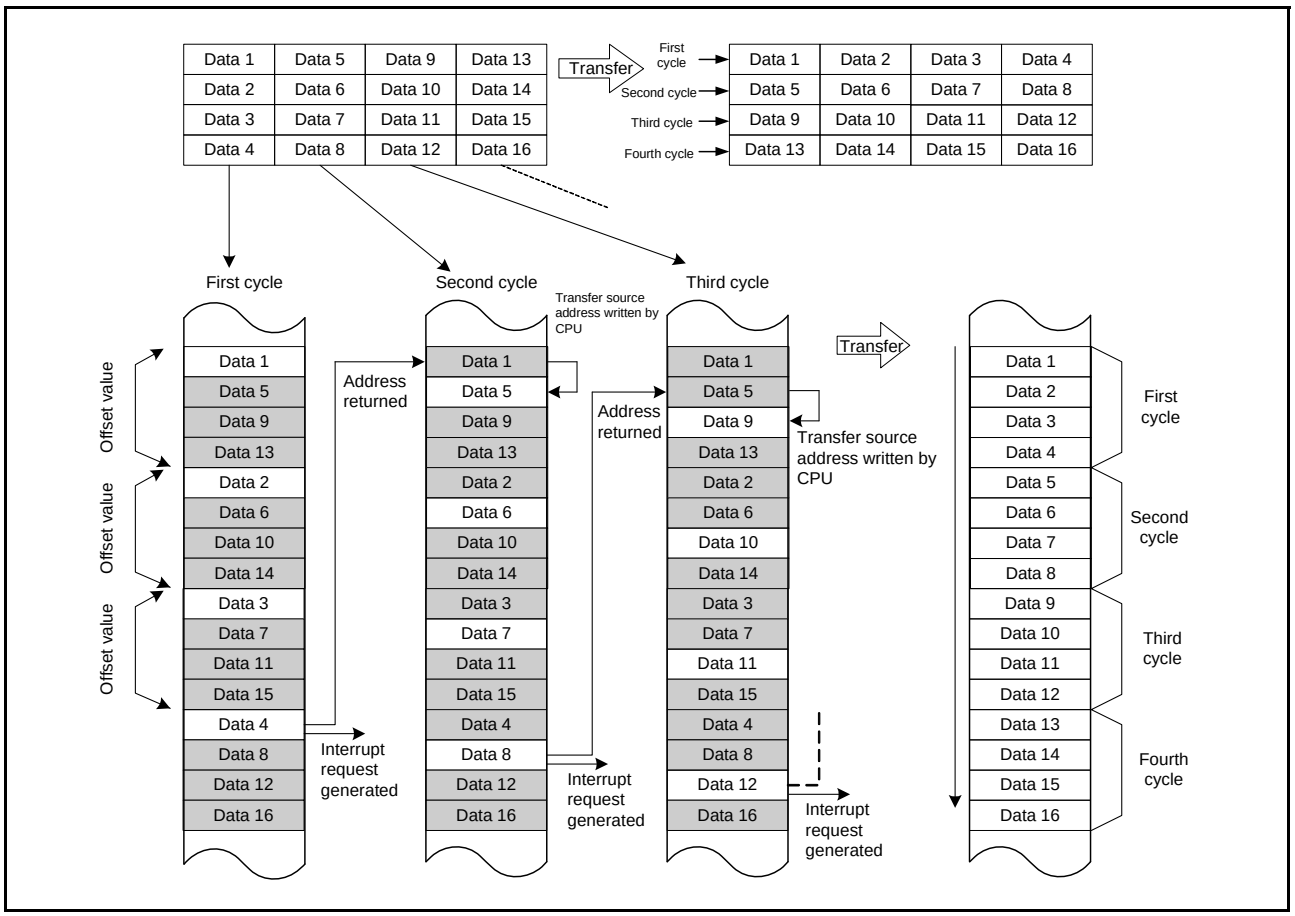


Figure 17.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

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.

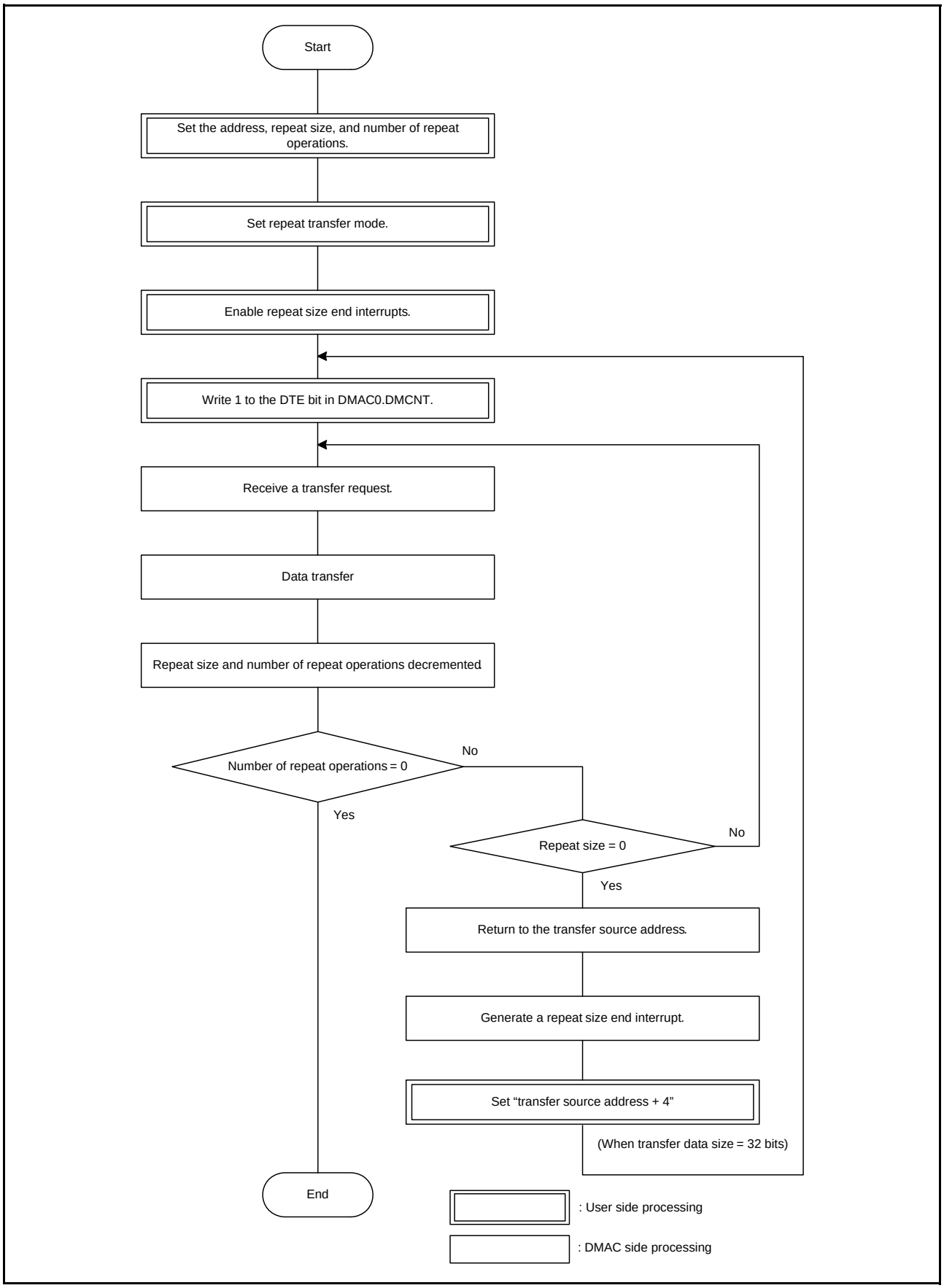


Figure 17.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

17.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the activation source.

(1) DMAC Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the DMAC activation 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 activation enabled).
When the DMAC is activated by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is cleared to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) DMAC Activation 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 DMAC activation sources. The activation source can be selected separately for each channel using the DMRSRm registers (m = 0 to 3) of the ICU.
The DMAC is activated 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 is 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 activation is enabled).

For interrupt requests specified as DMAC activation sources, see Table 14.3, Interrupt Vector Table, in section 14, Interrupt Controller (ICUb).

17.3.5 Operation Timing

Figure 17.10 and Figure 17.11 show DMAC operation timing examples.

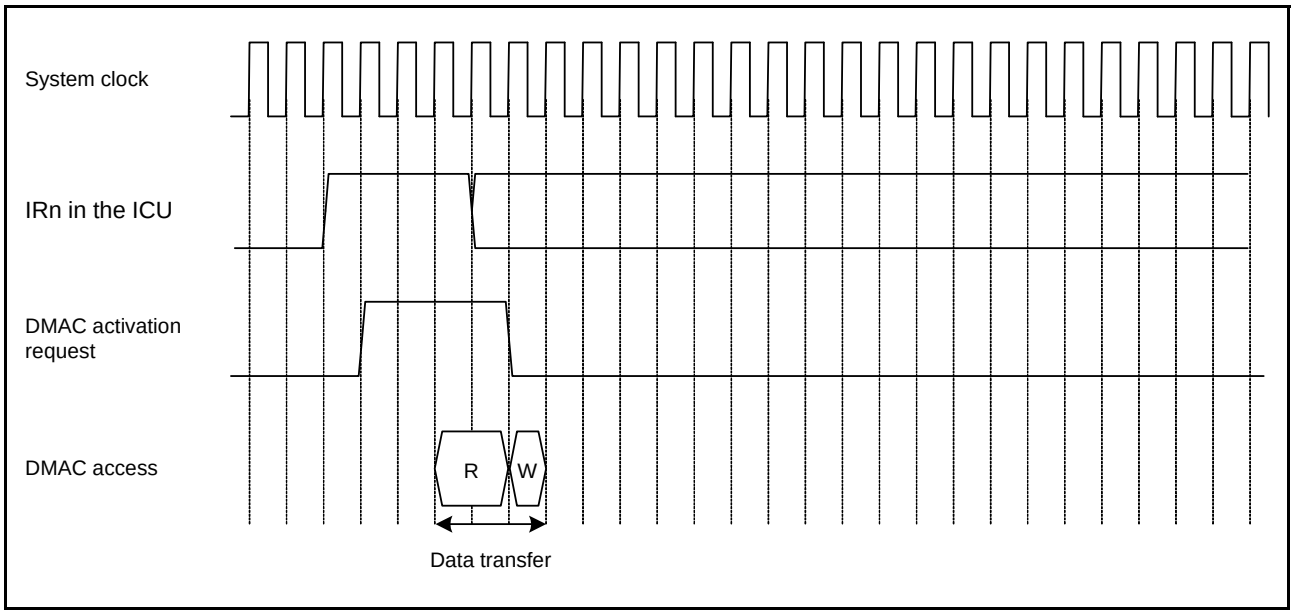


Figure 17.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

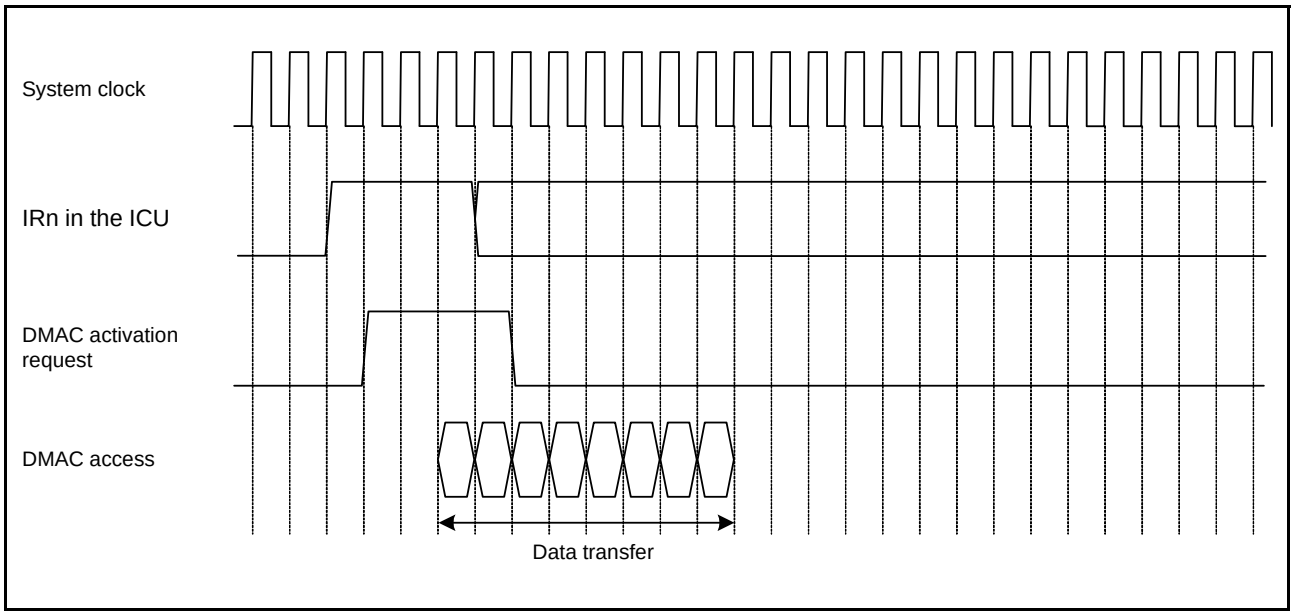


Figure 17.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

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 39, RAM, section 40, Flash Memory, section 5, I/O Registers, and section 15.2.6, External Bus.
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.

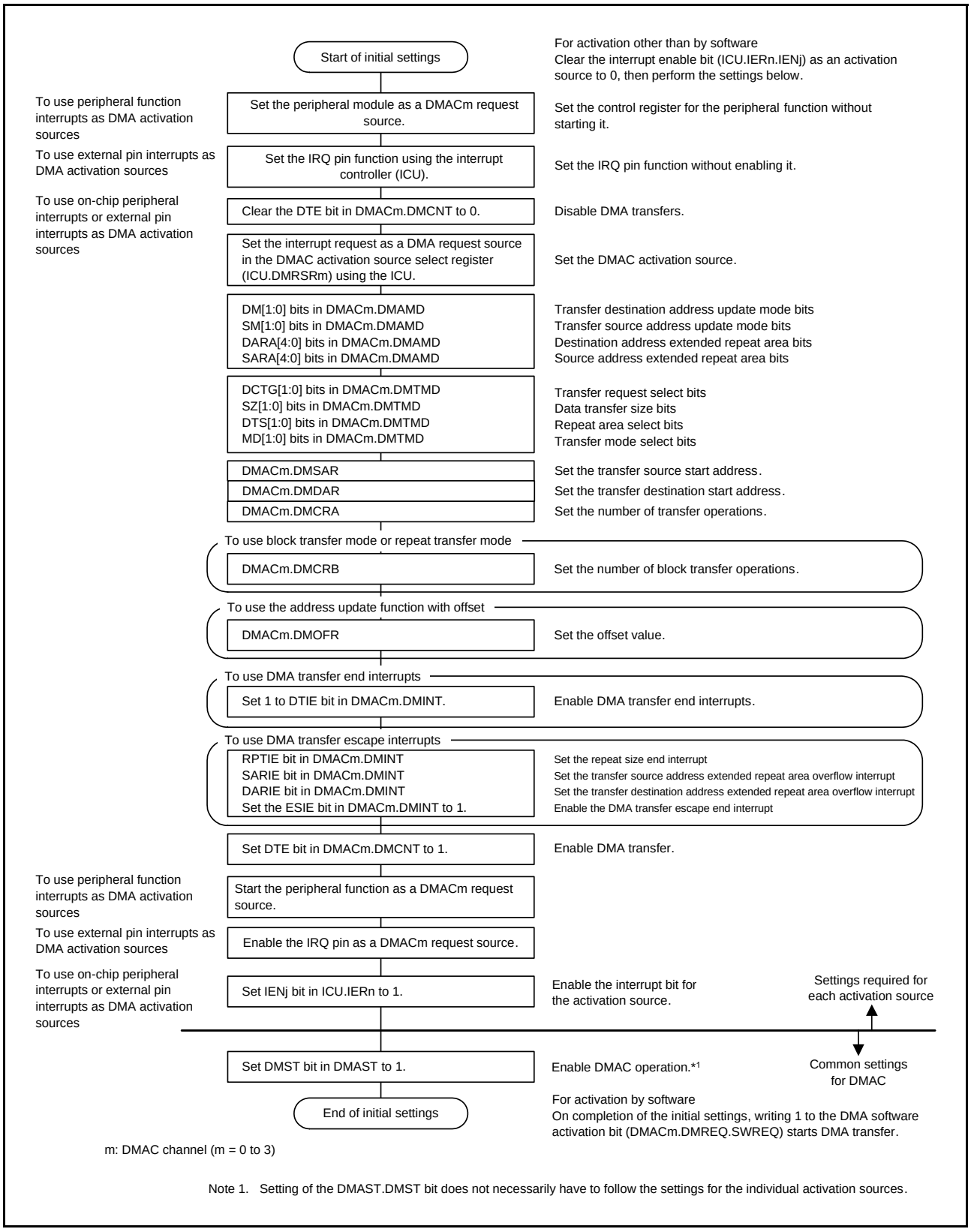


Figure 17.12 Register Setting Procedure

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 start enabled) enable DMA transfer of channel m (m = 0 to 3).
Another activation 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 cleared 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 cleared 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 cleared 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 cleared to 0 indicating the DMA transfer end.
This flag is automatically cleared 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 (ICUb).

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 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 cleared 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 cleared 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 cleared 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 (ICUb).

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 cleared 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 (ICUb).

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 cleared 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 (ICUb).

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 external bus or 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. Figure 17.14 shows the DMAC interrupt handling routine to resume or 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		

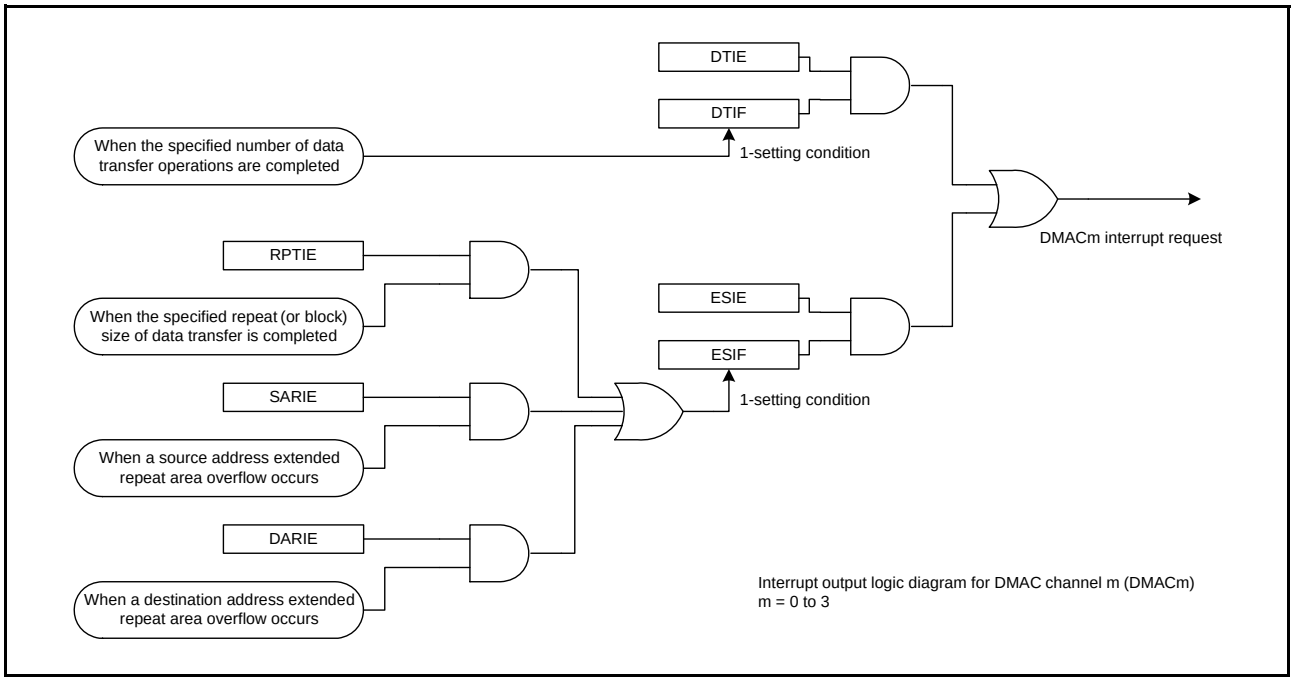


Figure 17.13 Schematic Logic Diagram of Interrupt Outputs

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 cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

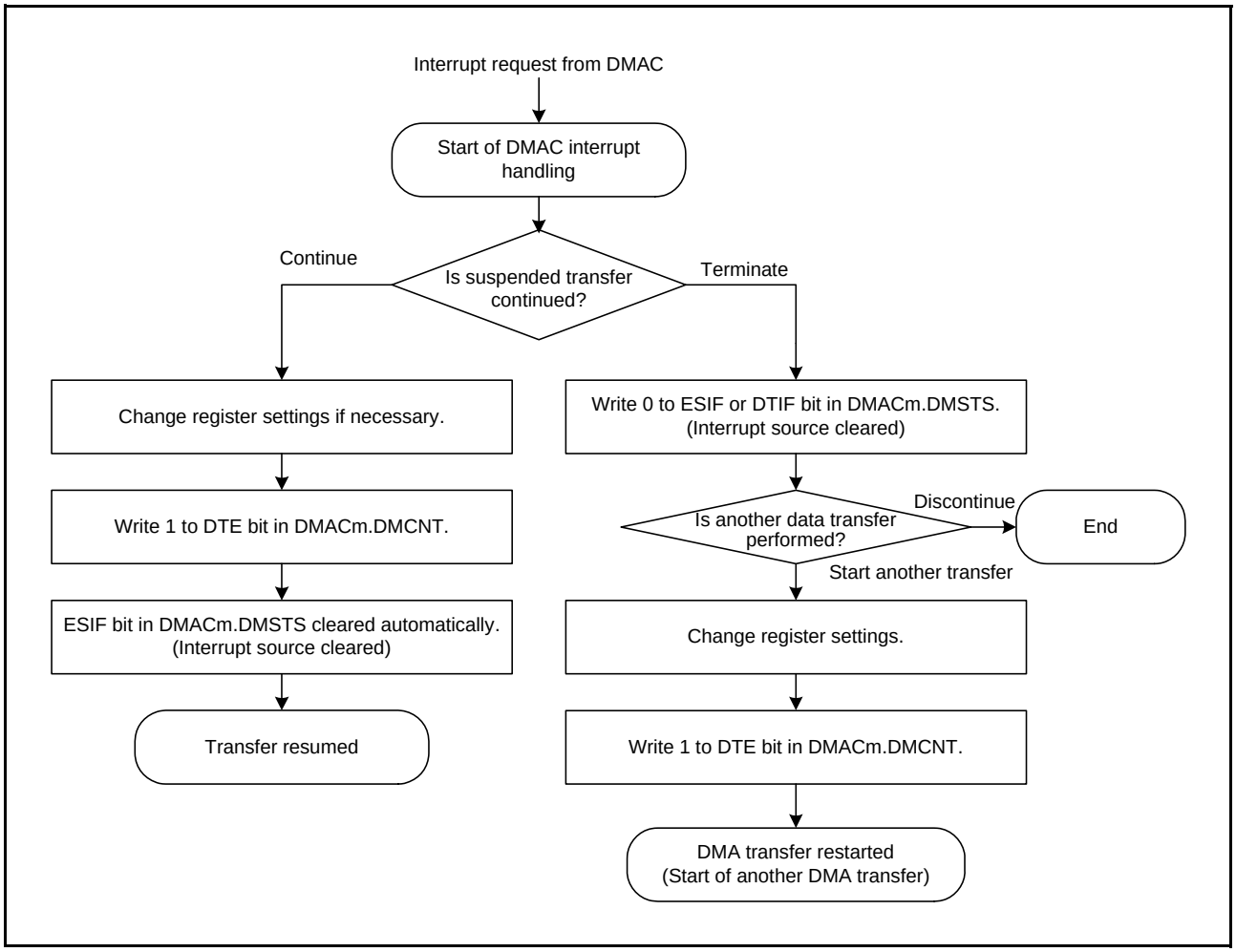


Figure 17.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6 Event Link Function

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 external bus or internal peripheral bus, an event link request signal is generated when the write to the write buffer is accepted.

17.7 Low Power Consumption Function

Before transition to the module stop state, all-module clock stop mode, software standby mode, or deep software standby mode, clear 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 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 11.6.2.1, 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 and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.3.1, Transition to Software Standby Mode or section 11.6.4.1, Transition to Deep 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 or deep 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 11.7.6, Timing of WAIT Instructions 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 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

17.8 Usage Notes

17.8.1 DMA Transfer to External Devices

In DMA transfer to an external device, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

17.8.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

17.8.3 Access to the Registers during DMA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm must not be accessed 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).

17.8.4 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.

17.8.5 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated 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 flag of the DMAC activation source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt flag is not cleared to 0 at the end of DMAC transfer, it should be cleared 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 (ICUb). For the DMACm.DMCSL.DISEL bit setting, see section 17.2.12, DMA Activation Source Flag Control Register (DMCSL).

17.8.6 Setting of DMAC Activation Source Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC activation source select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation 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 (ICUb).

17.8.7 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the interrupt enable bit for the activation 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 (DTCa)

This MCU incorporates a data transfer controller (DTC).
The DTC is activated by an interrupt request to perform data transfers.

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
Transfer modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation 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 activation leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer corresponding to the interrupt source is possible (transferred by the DTC activation request from the ICU). • Multiple data can be transferred on a single activation source (chain transfer). • Either “executed when the counter is 0” or “always executed” can be selected for chain transfer.
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 DTC activation interrupt. • 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	Transfer information read skip can be executed.
Write-back skip	When “fixed” is selected for transfer source address or transfer destination address, write-back skip is executed.
Low power consumption function	Module stop state can be set.

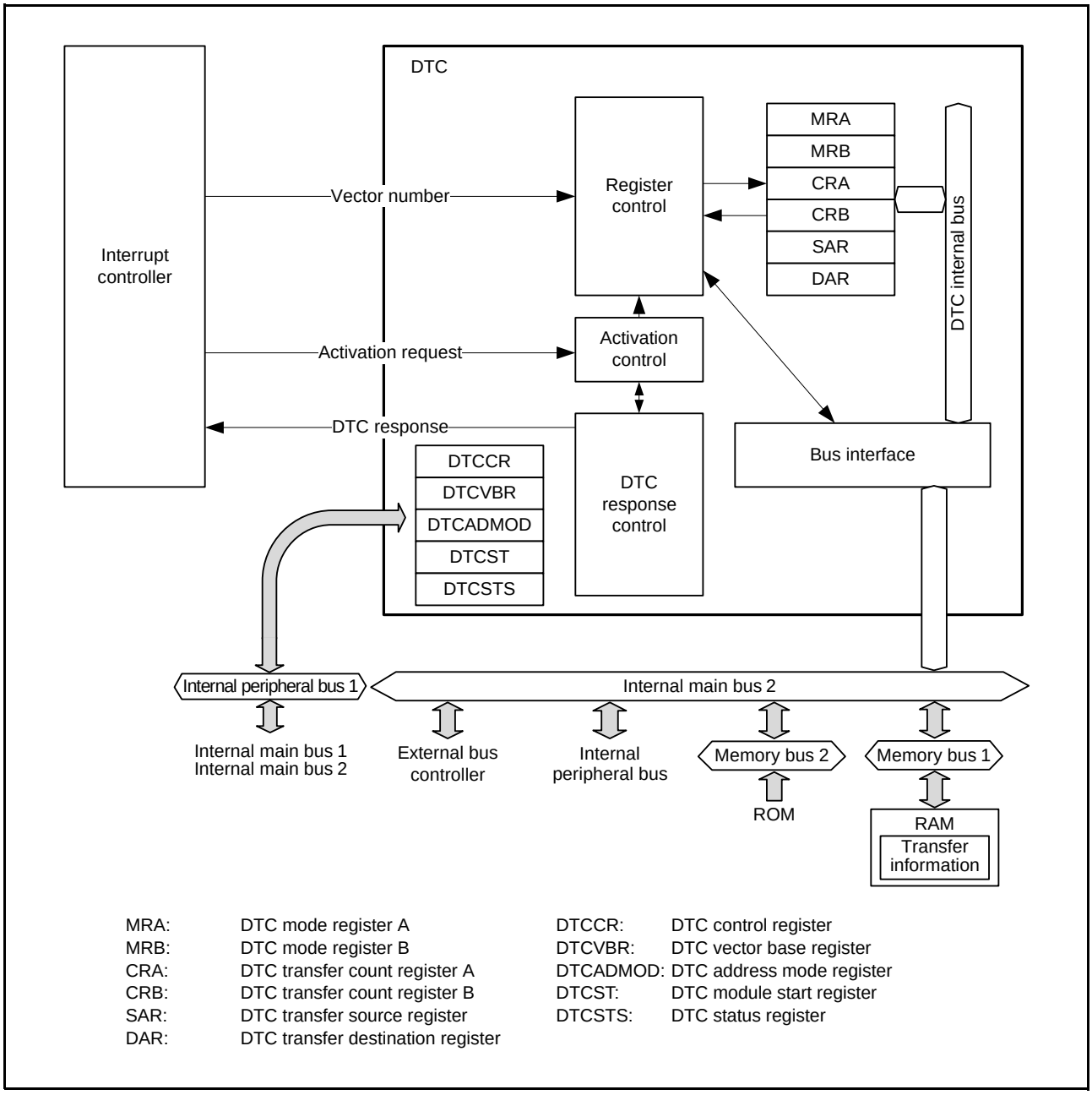


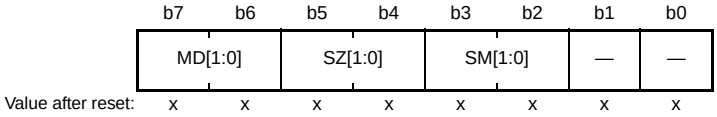
Figure 18.1 DTC Block Diagram

18.2 Register Descriptions

Registers MRA, MRB, 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 an activation request is generated, the DTC reads the transfer information from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents 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)



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: Address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer. (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: SAR value is decremented after data transfer. (–1 when SZ[1:0] bits = 00b, –2 when SZ[1:0] bits = 01b, –4 when SZ[1:0] bits = 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 cannot be accessed directly from the CPU.

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]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: Address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: Address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: DAR value is incremented after data transfer. (+1 when MRA.SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer. (–1 when SZ[1:0] bits = 00b, –2 when SZ[1:0] bits = 01b, –4 when MRA.SZ[1:0] bits = 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 when specified data transfer is completed. 1: An interrupt request to the CPU is generated each time DTC data transfer is performed.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed continuously. 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.	—

MRB cannot be accessed directly from the CPU.

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 18.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag 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.

18.2.3 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 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 cannot be accessed directly from the CPU.

18.2.4 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

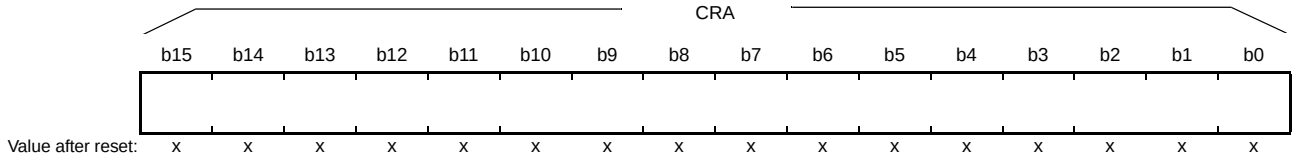
x: Undefined

DAR 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 cannot be accessed directly from the CPU.

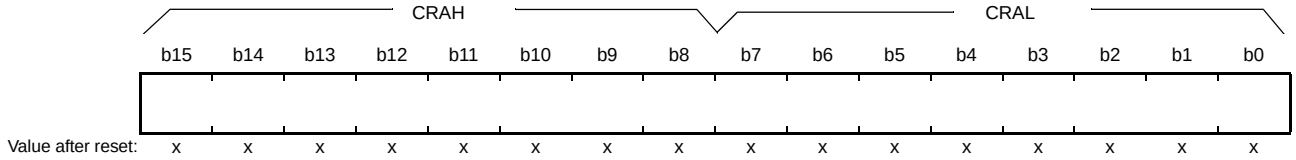
18.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA 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 transferred to CRAL.

(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 transferred to CRAL.

18.2.6 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 is used to set the block transfer count for block transfer mode.
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.
CRB cannot be accessed directly from the CPU.

18.2.7 DTC Control Register (DTCCR)

Address(es): 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	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

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous activation process.
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.

18.2.8 DTC Vector Base Register (DTCVBR)

Address(es): 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
							0	0	0	0	0	0	0	0	0	0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit Name	Description	R/W
b9 to b0	DTC Vector Base Address (Lower 10 bits)	These bits are read as 0. The write value should be 0.	R
b31 to b10	DTC Vector Base Address (Upper 22 bits)	Writing to the upper 4 bits (b31 to b28) is ignored, and the address of this register is extended by the value specified by b27.	R/W

DTCVBR is used to set the base address for calculating the DTC vector table address.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

18.2.9 DTC Address Mode Register (DTCADMOD)

Address(es): 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	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

DTCADMOD is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

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).

18.2.10 DTC Module Start Register (DTCST)

Address(es): 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.

Before making a transition to the module stop state, all-module clock stop mode, or software standby mode, or deep software standby mode, the DTCST bit must be set to 0.

For details on transitions to the module stop state, all-module clock stop mode, software standby mode, and deep software standby mode, refer to section 18.9, Low Power Consumption Function, and section 11, Low Power Consumption.

18.2.11 DTC Status Register (DTCSTS)

Address(es): 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-Activating Vector Number Monitoring	These bits indicate the vector number for the activation source when DTC transfer is in progress. The value is only valid if DTC 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: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activation source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer).

For the correspondence between the DTC activation sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

ACT Flag (DTC Active Flag)

This flag indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC is completed in response to a transfer request.

18.3 Activation Sources

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (n = interrupt vector number) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.
For the correspondence between the DTC activation sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For activation by software, refer to section 14.2.5, Software Interrupt Activation Register (SWINTR) in section 14, Interrupt Controller (ICUb).
Once the DTC has accepted an activation request, it does not accept another activation request until transfer for that single request is completed, regardless of the priority of the requests. When multiple activation requests are generated during DMAC/DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple activation requests are generated while the DTC module start bit (DTCST.DTCST) is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 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 round of data transfer, the 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 activation 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 activation 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. In the RAM area, the start address of the transfer information (n) with vector number n should be 4n added to the base address in the vector table.
Transfer information can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 18.2 shows the relationship between the DTC vector table and transfer information.
Figure 18.3 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.

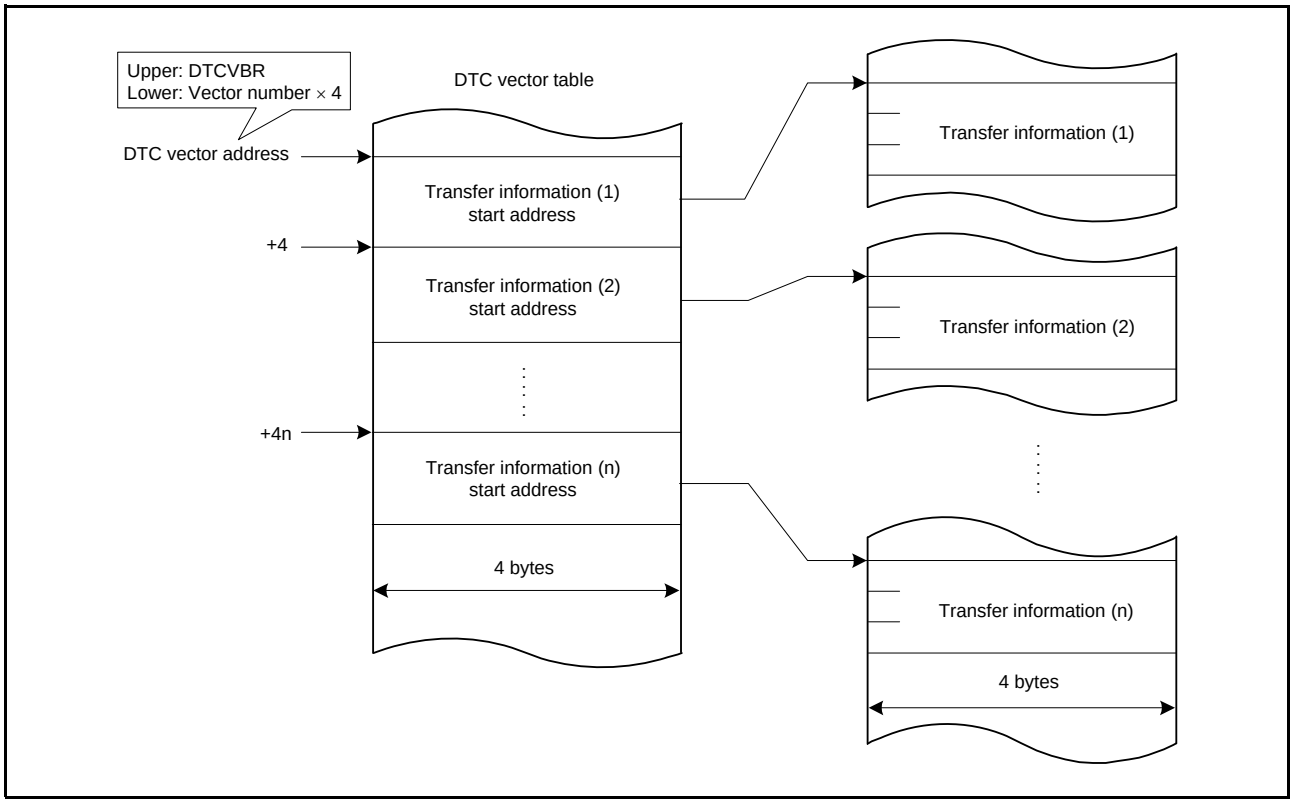


Figure 18.2 DTC Vector Table and Transfer Information

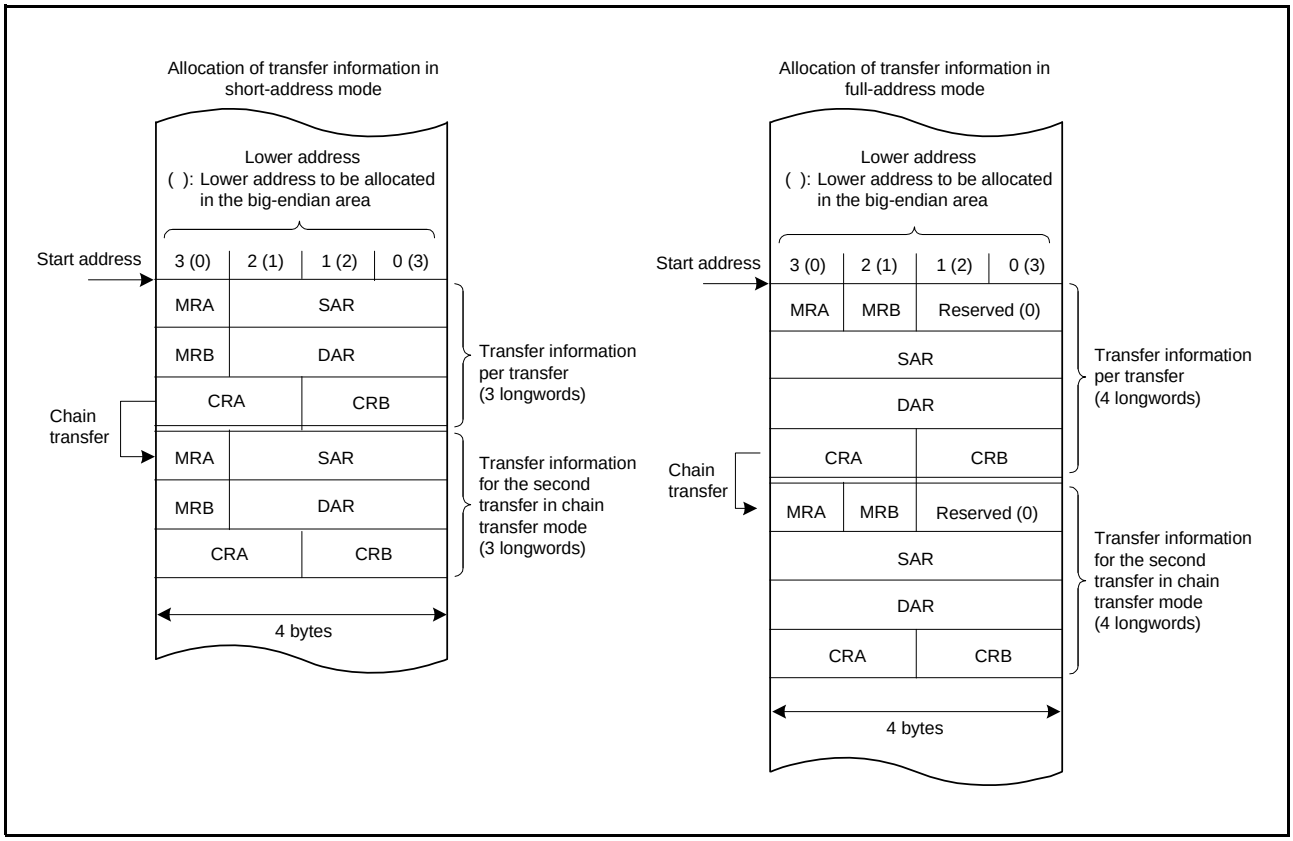


Figure 18.3 Allocation of Transfer Information in the RAM Area

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 is activated, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the transfer information store address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Storing 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.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 18.2 lists transfer modes of the DTC.

Table 18.2 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 activation source. Setting the MRB.CHNS bit also enables chain transfer when specified data transfer is completed.

Figure 18.4 shows the operation flowchart of the DTC. Table 18.3 lists chain transfer conditions.

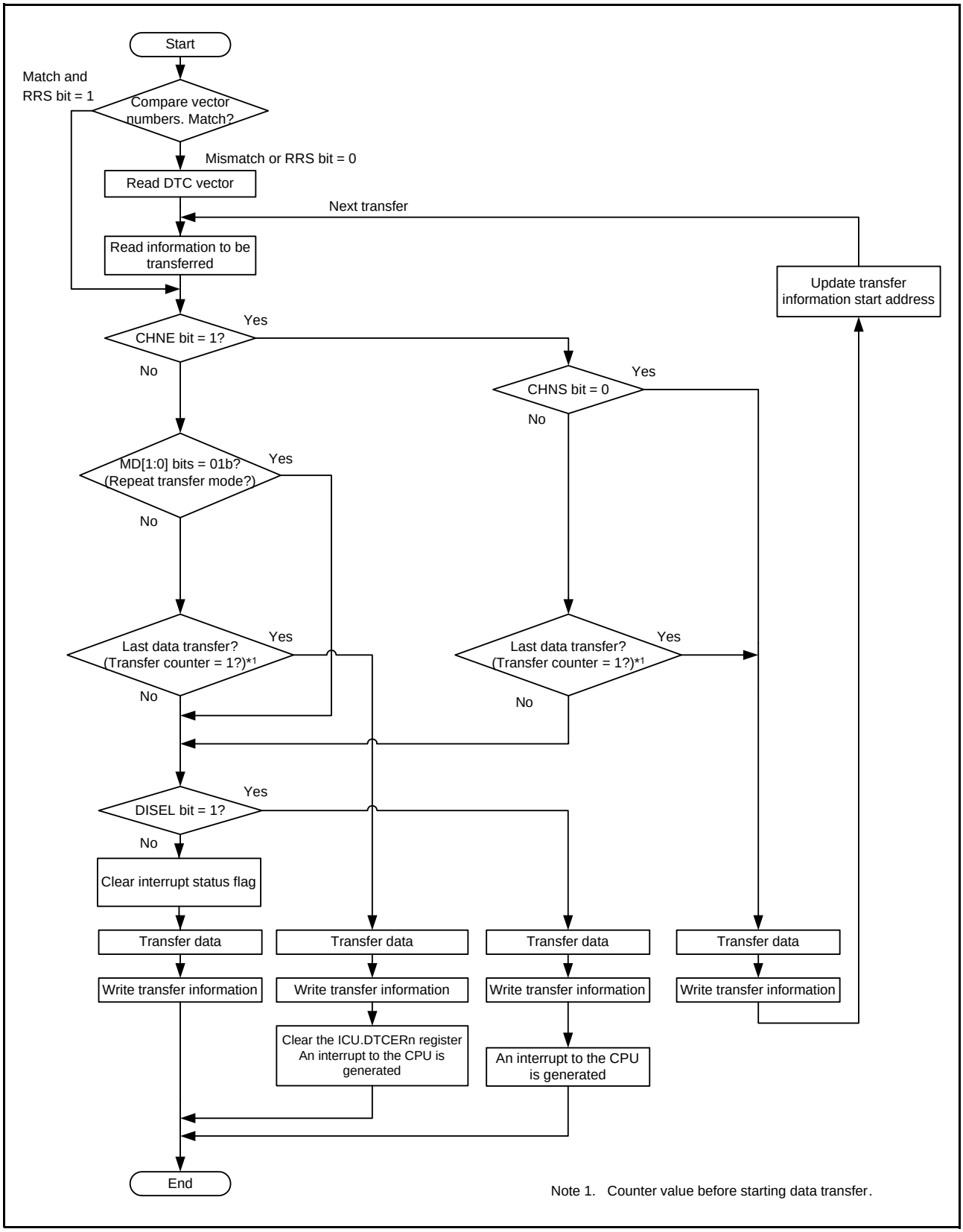


Figure 18.4 Operation Flowchart of the DTC

Table 18.3 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	DTC 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 CHNE bit = 1” is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer was chain transfer, the vector address 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 18.13 shows an example of transfer information read skip.

To update 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. The retained vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

18.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address fixed”, 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 18.4 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 the MRA and MRB registers are skipped.

Table 18.4 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		

18.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source. 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 18.5 lists register functions in normal transfer mode, and Figure 18.5 shows the memory map of normal transfer mode.

Table 18.5 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	Increment/decrement/fixed*1
DAR	Transfer destination address	Increment/decrement/fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

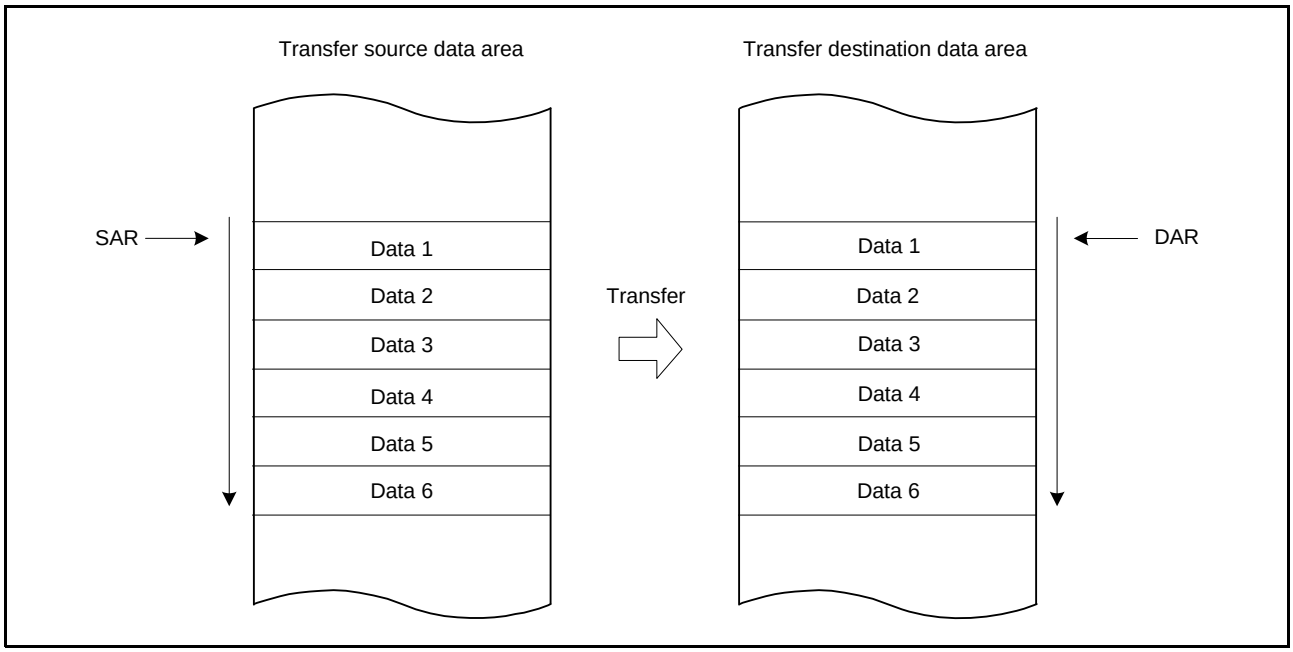


Figure 18.5 Memory Map of Normal Transfer Mode

18.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source.
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 when specified data transfer is completed).
Table 18.6 lists the register functions in repeat transfer mode, and Figure 18.6 shows the memory map of repeat transfer mode.

Table 18.6 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information	
Register	**Description**	**When CRAL is not 1**	**When CRAL is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	(When the MRB.DTS bit is 0) Increment/decrement/fixed*1 (When the MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	(When the MRB.DTS bit is 0) DAR register initial value (When the MRB.DTS bit is 1) Increment/decrement/fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

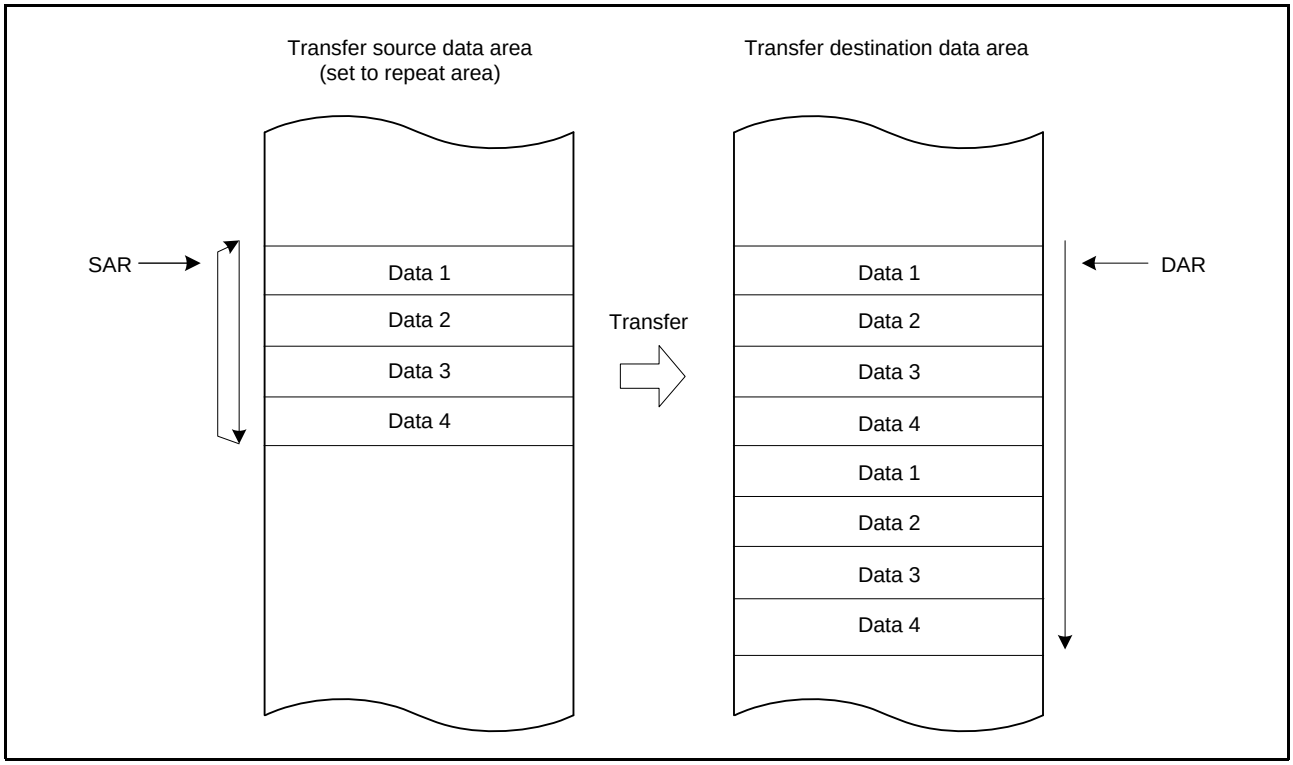


Figure 18.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

18.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single activation source.
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 = 1 or the DAR register when the DTS bit = 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 18.7 lists register functions in block transfer mode, and Figure 18.7 shows the memory map of block transfer mode.

Table 18.7 Register Functions in Block Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	(When MRB.DTS bit is 0) Increment/decrement/fixed*1 (When MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	(When MRB.DTS bit is 0) DAR register initial value (When MRB.DTS bit is 1) Increment/decrement/fixed*1
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

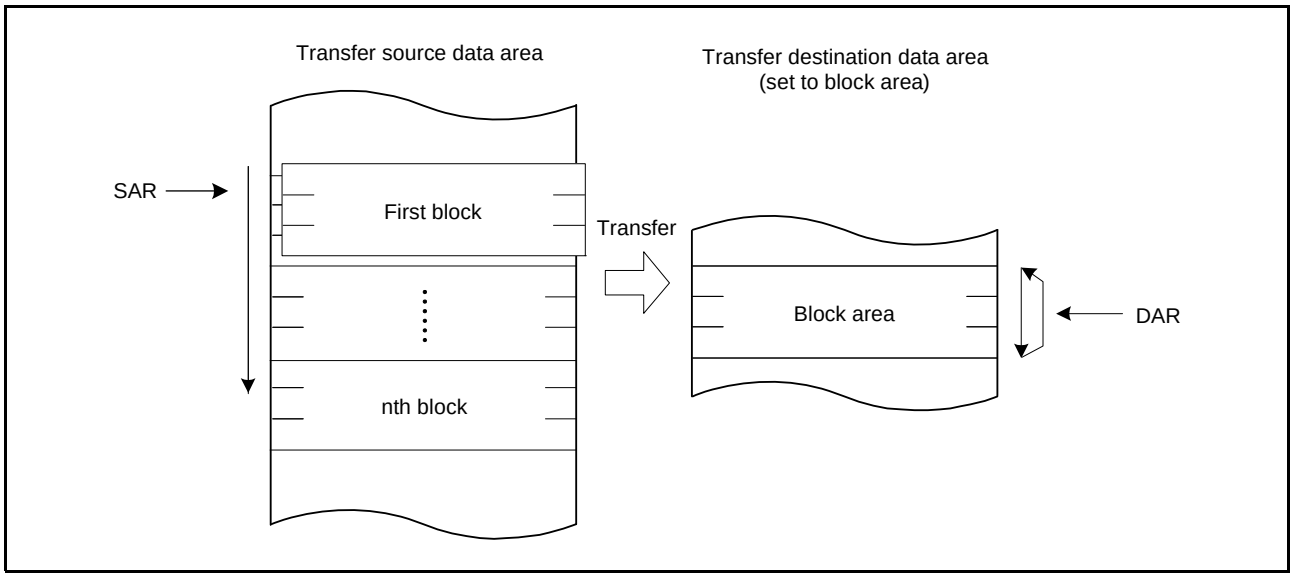


Figure 18.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

18.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source.
If the MRB.CHNE and CHNS bits are set to 1 and 0, respectively, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the MRB.DISEL bit to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed), and data transfer has no effect on the interrupt status flag of the activation source.
The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer.
Figure 18.8 shows chain transfer operation.

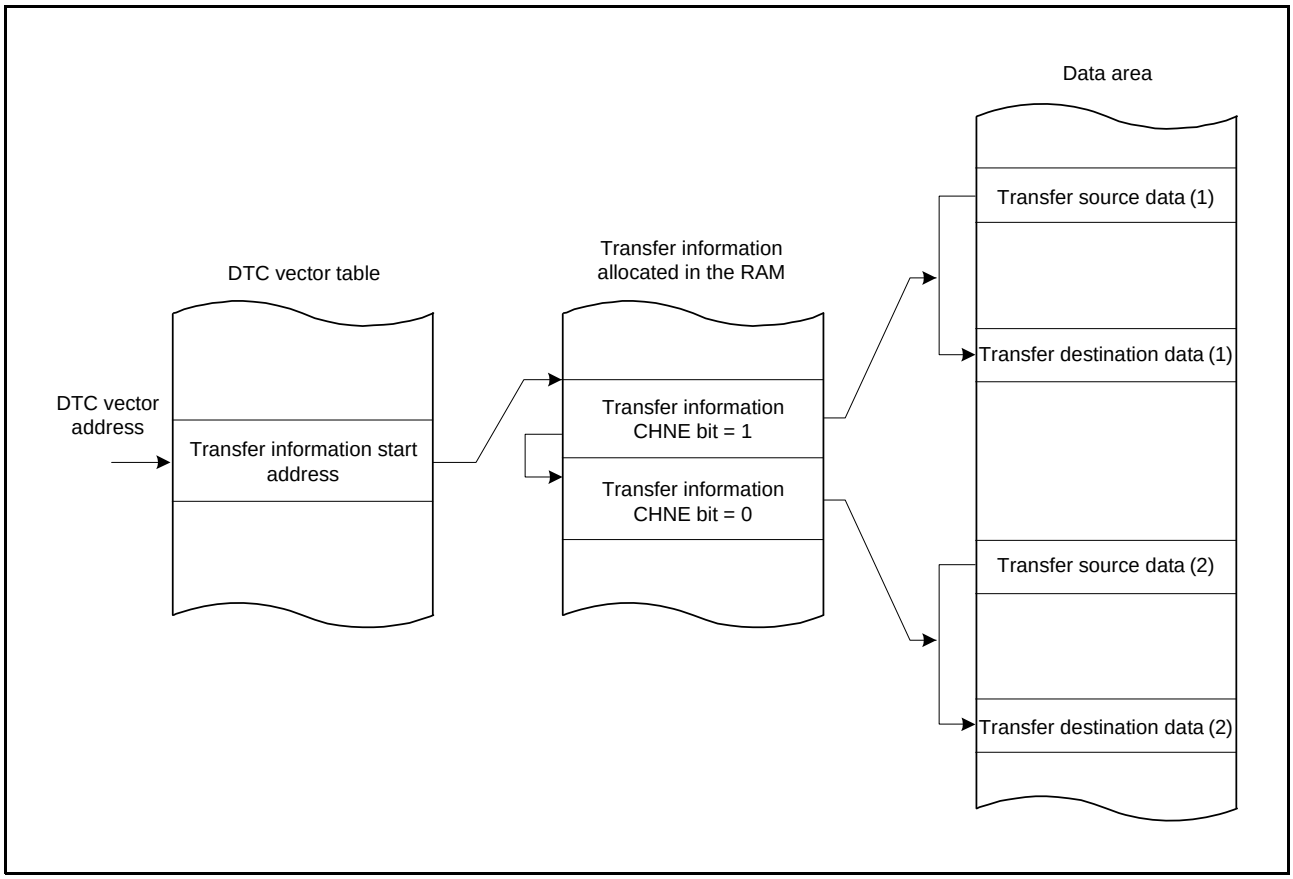


Figure 18.8 Chain Transfer Operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of specified data transfer. In repeat transfer mode, chain transfer is performed after completion of specified data transfer.
For details on chain transfer conditions, see Table 18.3, Chain Transfer Conditions.

18.4.7 Operation Timing

Figure 18.9 to Figure 18.13 show examples of DTC operation timing.

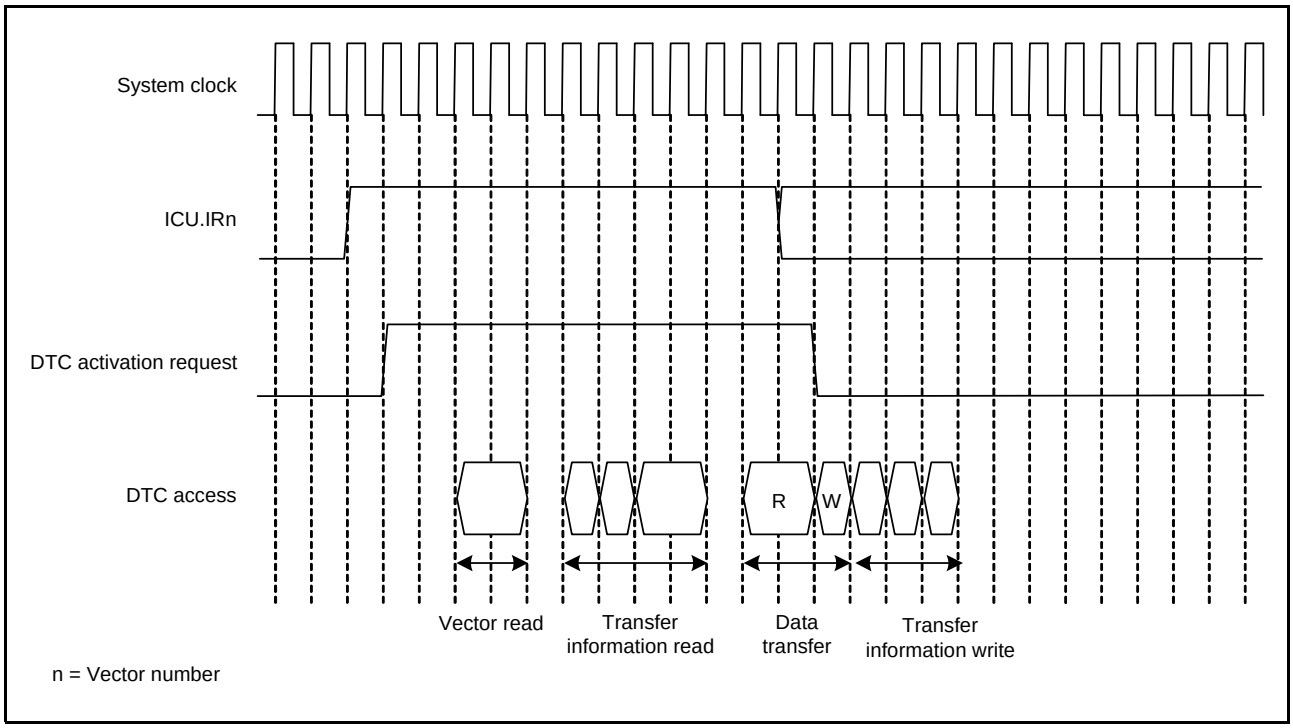


Figure 18.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

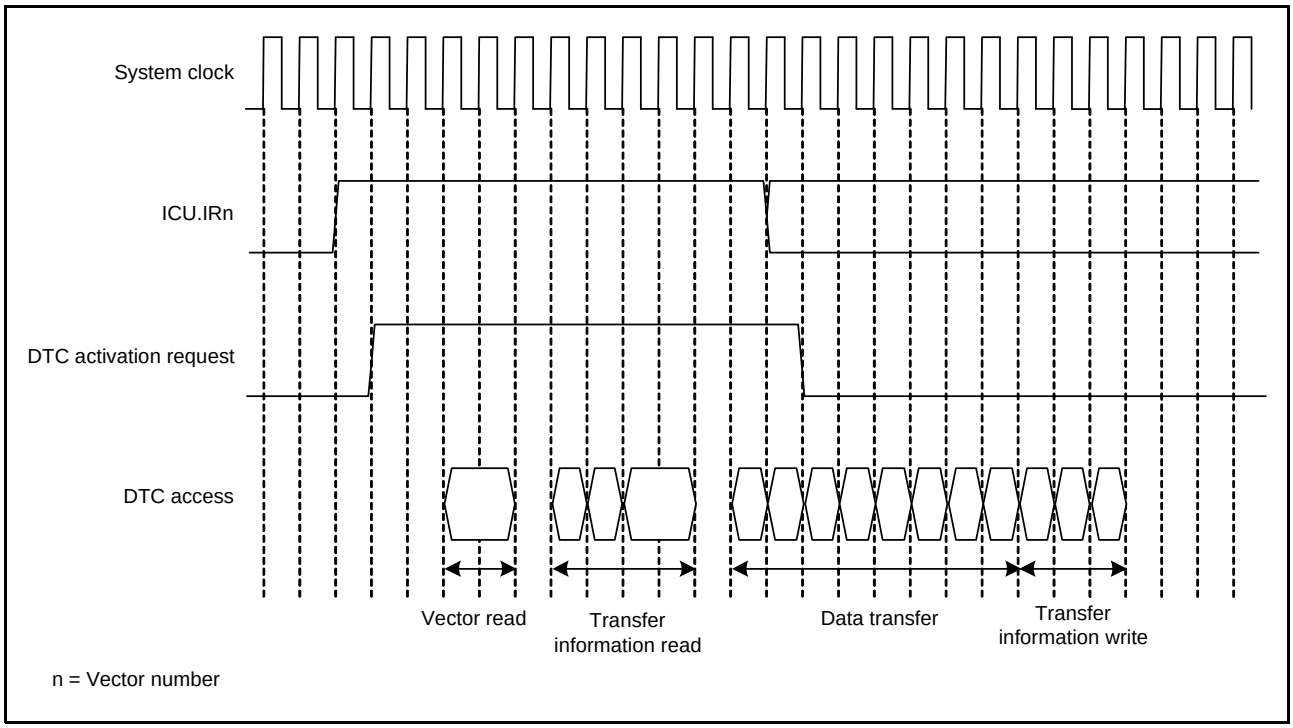


Figure 18.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

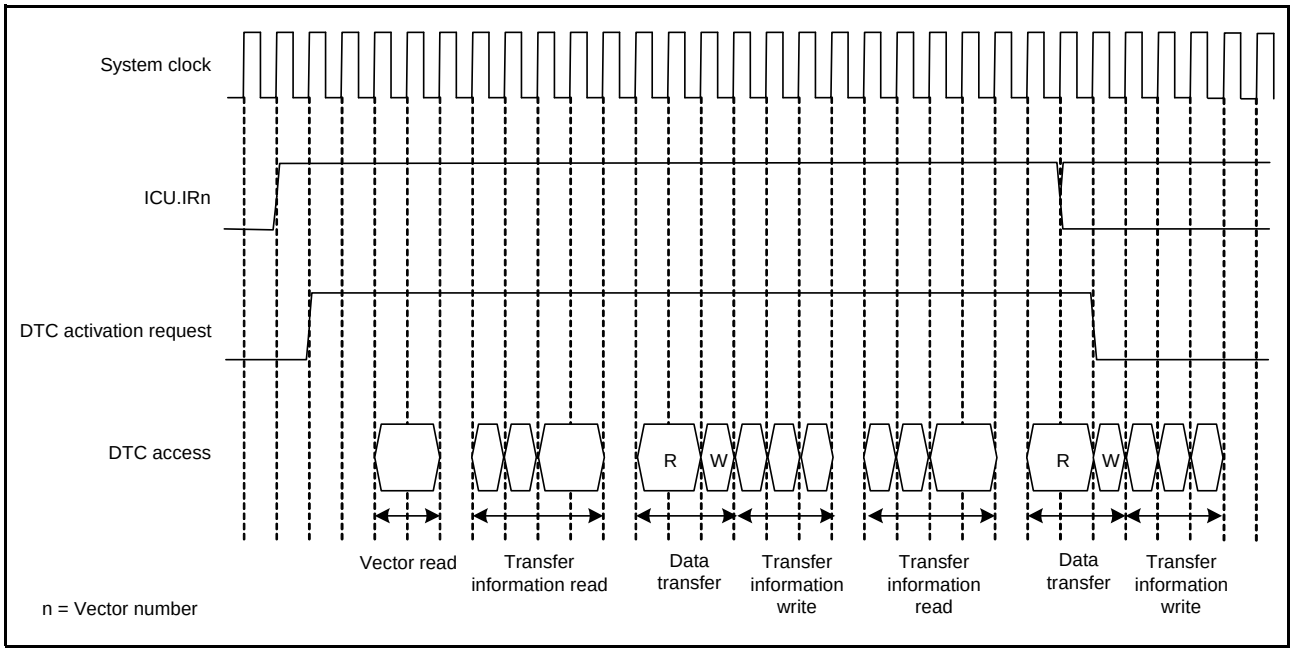


Figure 18.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

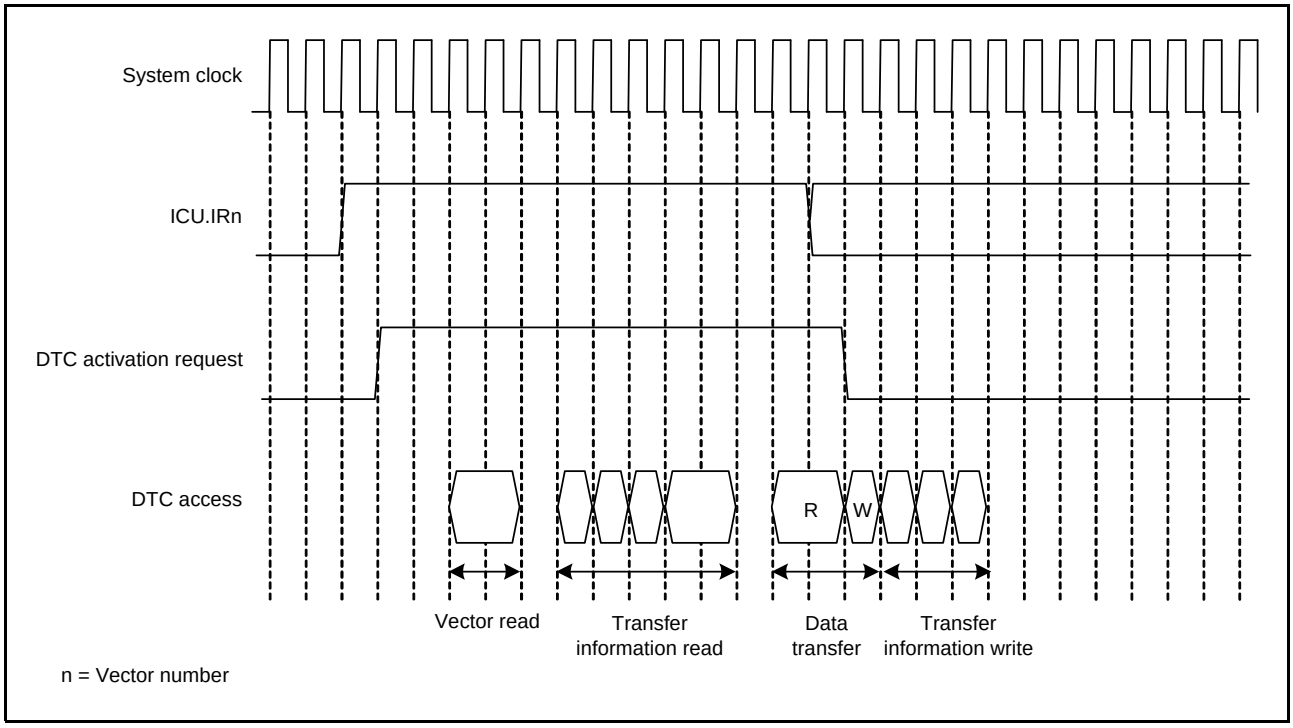
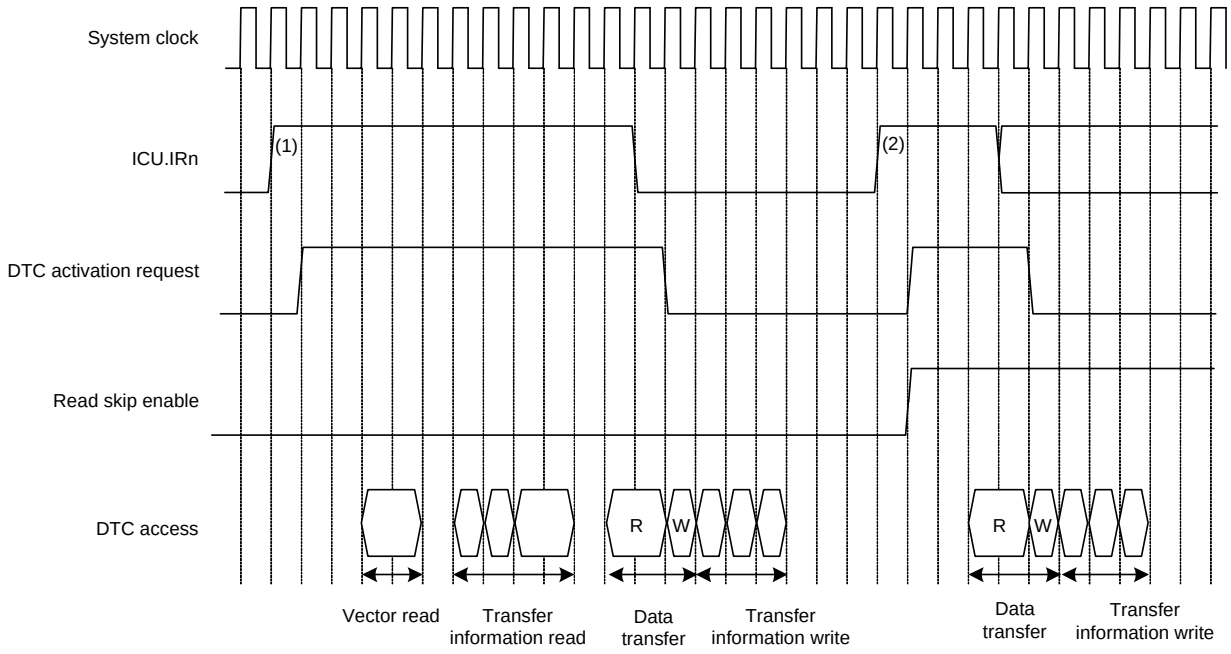


Figure 18.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer information read for request (2) is skipped.

Figure 18.13 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 18.8 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 18.8 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 mode
Note 5. When SAR or DAR is set to address-fixed mode
Note 6. When SAR and DAR are set to address-fixed mode
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 39, RAM, section 40, Flash Memory, section 5, I/O Registers, and section 15.2.6, External Bus.)

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.

18.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 18.14 shows the procedure to set the DTC.

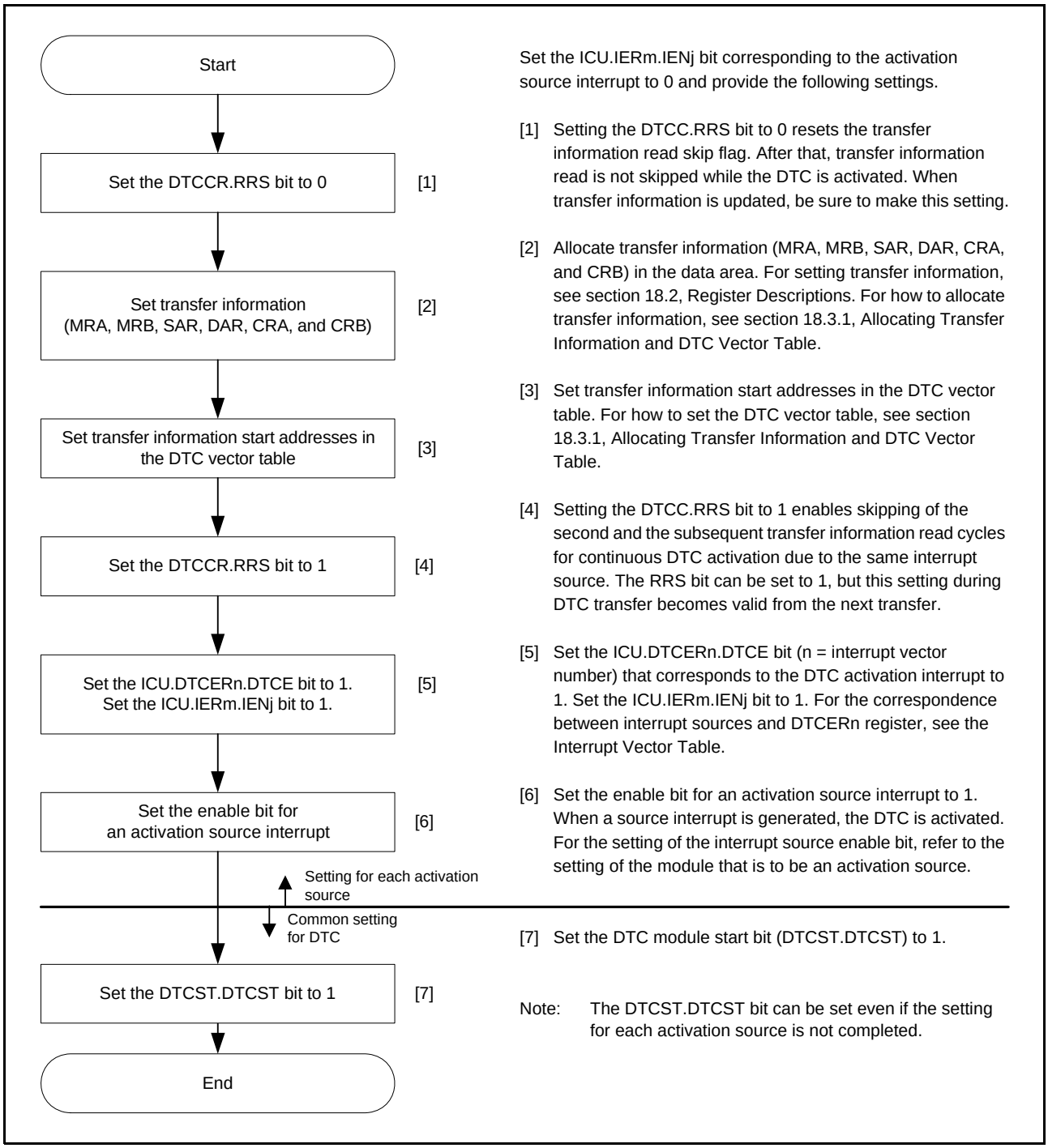


Figure 18.14 Procedure to Set the DTC

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

In the MRA register, select a fixed source address (MRA.SM[1:0] bits = 00b), normal transfer mode (MRA.MD[1:0] bits = 00b), and byte-sized transfer (MRA.SZ[1:0] bits = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] bits = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 0). 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.IERi.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 activate the DTC. 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 rounds of data transfer have been completed and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

18.6.2 Chain Transfer

As an example of chain transfer by the DTC, its employment in the output of pulses by a PPG is described below. Lower-case letters i, j, m, and n in this text indicate a unit, channel, or bit number.
Chain transfer is used to transfer pulse output data and change the period of the output trigger for the PPG. For the first half of the chain transfer, repeat transfer mode for transfer to the PPGm.NDRH and PPGm.NDRL registers is specified. For the second half, normal transfer mode for transfer to the MTUn.TGR registers is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of rounds of transfer are restricted to the second half of the chain transfer (transfer while MRB.CHNE bit = 0).
An example of using the compare match interrupt for an MTUn.TGRA register as an activation source for the DTC is provided below.

(1) First Transfer Information Setting

Settings should be made for transfer to the PPGm.NDRH and PPGm.NDRL registers. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] bits = 10b), transfer in repeat transfer mode (MRA.MD[1:0] bits = 01b), and word-sized transfer (MRA.SZ[1:0] bits = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] bits = 00b) and for chain transfer (MRB.CHNE bit = 1 and MRB.CHNS bit = 0). Set the source side on the repeat area (MRB.DTS bit = 1). Set the SAR to the first address of the data table, the DAR register to the address of the PPGm.NDRH register, and the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second Transfer Information Setting

Settings should be made for transfer to the MTUn.TGRA register. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] bits = 10b), transfer in normal mode (MRA.MD[1:0] bits = 00b), and word-sized transfer (MRA.SZ[1:0] bits = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] bits = 00b) and for the single data transfer per interrupt (MRB.CHNE bit = 0, MRB.DISEL bit = 0). The MRB.DTS bit can be set to any value. Set the SAR register to the first address of the data table, the DAR register to the address of the MTUn.TGRA register, and the CRA register to the size of the data table. The CRB register can be set to any value.

(3) Transfer Information Assignment

Place the transfer information for use in transfer to the MTU immediately after the transfer control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers.

(4) DTC Vector Table

In the DTC vector table, set the address where the transfer control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers starts.

(5) ICU Setting and DTC Module Activation

Set the ICU.DTCERn.DTCE bit corresponding to the TGIA interrupt and the ICU.IERi.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(6) MTU Setting

In the given MTUn, set the TIOR register so that the TGRA register operates as an output compare register (with output disabled) and make the TIER setting to enable TGIAn interrupt requests.

(7) PPG Setting

Set the default output values in the PPGm.PODRH and PPGm.PODRL registers and the next output values in the PPGm.NDRH and PPGm.NDRL registers. Set 1 to the output bits in PORTm.PDR and PPGm.NDRH, and PPGm.NDRL. Also, select a compare match signal of the MTU as the output trigger in the PPGm.PCR register.

(8) MTU Activation

Set the MTU.TSTR.CSTj bits to 1 to start counting operation of the MTUn.TCNT counter.

(9) DTC Transfer

Every time a compare-match with the MTUn.TGRA register is generated, next output values are transferred to the PPGm.NDRH and PPGm.NDRL registers and the setting for the next output trigger period is transferred to the MTUn.TGRA register.

(10) Interrupt Handling

After the specified number of rounds of data transfer has been completed (i.e. when the value in the CRA register for MTU transfer has become 0), a TGIAn interrupt request is issued for the CPU. Complete the process in the handling routine for this interrupt.

18.6.3 Chain Transfer When the Counter = 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 transfer. Repeating this chain transfer enables transfers to be repeated 256 times or more.
The following shows an example of configuring a 128-Kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 18.15 shows a chain transfer when the counter = 0.

1. Set normal transfer mode for input data for the first data transfer. Set the following:
   Transfer source address: Fixed, the CRA register = 0000h (65,536 times), the MRB.CHNE bit = 1 (chain transfer is enabled), the MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0), and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
2. Prepare the upper 8-bit address 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). For example, when setting the input buffer to 20 0000h to 21 FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination. At this time, set the MRB.CHNE bit = 0 (chain transfer is disabled) and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed). When setting the input buffer mentioned above to 20 0000h to 21 FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

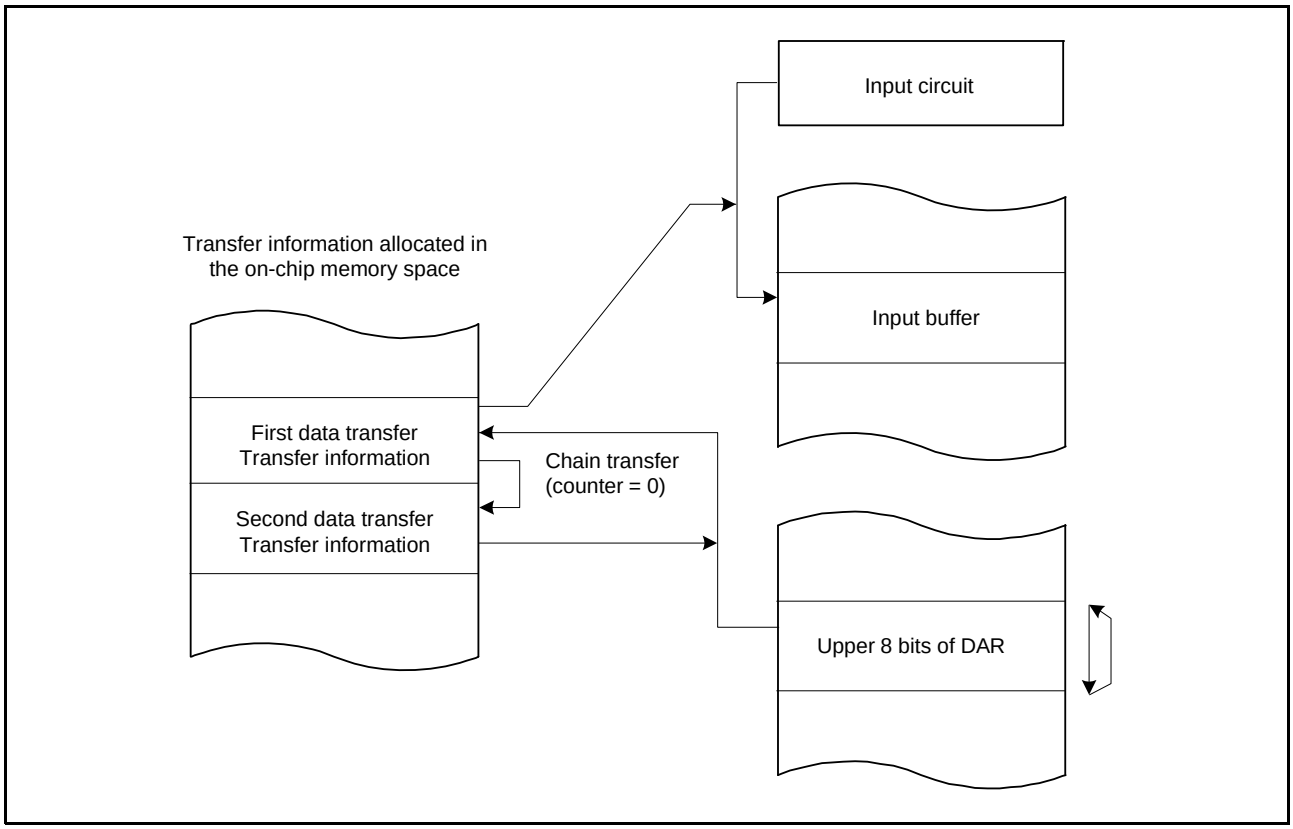


Figure 18.15 Chain Transfer When the Counter = 0

18.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC activation 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 is capable of producing an event link request on completing transfer in response to one request. When the destination for transfer is an external bus or an internal peripheral bus, however, the event link request will be issued after completion of writing to the write buffer rather than after completion of writing to the actual destination for transfer.

18.9 Low Power Consumption Function

Before making a transition to the module stop function, all-module clock stop mode, software standby mode, or deep software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (this module clock is enabled) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If DTC transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (this module clock is disabled) 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 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If DTC 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 DTC 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 and Deep Software Standby Modes

Make settings according to the procedure under section 11.6.3.1, Transition to Software Standby Mode, or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If DTC transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby mode or deep software standby mode follows the completion of DTC transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.6, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DTC 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 activation request, specify the CPU as the interrupt request destination according to the description in section 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

18.10 Usage Notes

18.10.1 Transfer Information Start Address

Be sure to set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

18.10.2 Allocating Transfer Information

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 18.16.
For example, when writing CRA and CRB setting data with 16 bits in big endian, write the CRA setting data to lower address 0 and the CRB setting data to lower address 2. In little endian, write the CRB setting data to lower address 0 and the CRA setting data to lower address 2. When writing CRA and CRB setting data with 32 bits, place the CRA setting data at the MSB side and the CRB setting data at the LSB side regardless of endian, and then write the data to lower address 0.

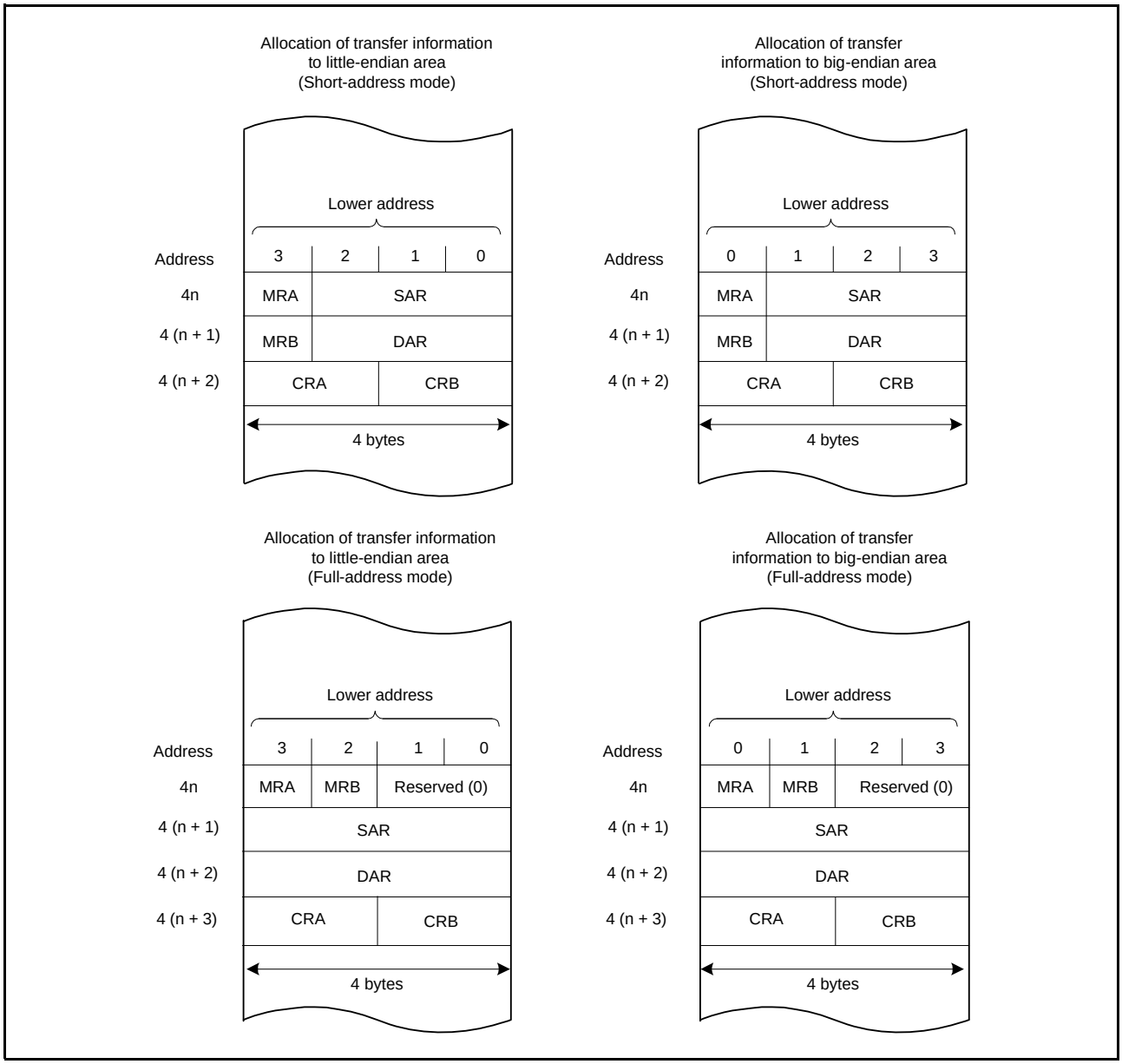


Figure 18.16 Allocation of Transfer Information

18.10.3 Setting the DTC Activation Enable Register (ICU.DTCERn) of the Interrupt Controller

The DMAC should not be activated by setting the DMAC activation request select register (ICU.DMRSRn (n = number of DMAC channel)) to the same vector number that has been specified by setting the ICU.DTCERn register to 1 (DTC transfer enabled). For details on the ICU.DTCERn and ICU.DMRSRn registers (n = number of DMACA channel), refer to section 14, Interrupt Controller (ICUb).

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 to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit.

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	• 56 types of event signals can be directly connected to modules. • The operation of timer modules can be selected when an event is input to the timer module. • Event link operation is possible for port B and port E. Single port*1: An event link can be set for a single bit specified in a port. Port group*1: An event link can be set for a group of single bits specified within eight I/O ports.
Low power consumption function	Module stop state can be set.

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value.

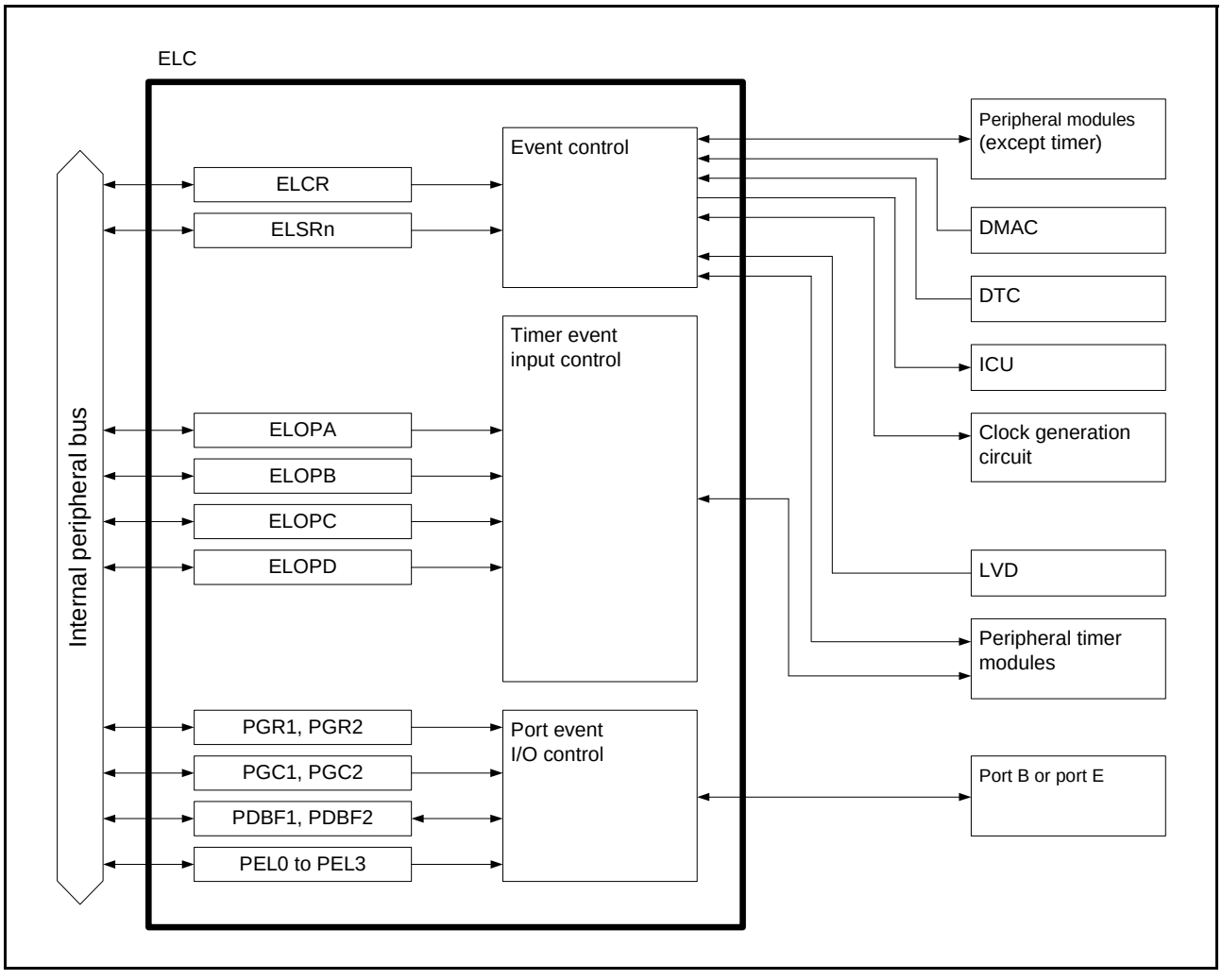


Figure 19.1 ELC Block Diagram (n = 1 to 4, 7, 10, 12, 15, 16, 18 to 29)

19.2 Register Descriptions

19.2.1 Event Link Control Register (ELCR)

Address(es): 0008 B100h

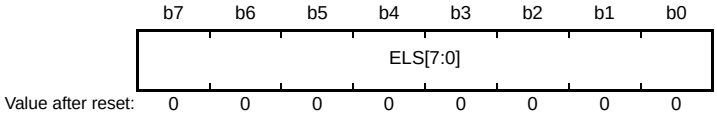
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

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 = 1 to 4, 7, 10, 12, 15, 16, 18 to 29)

Address(es): ELSR1 0008 B102h, ELSR2 0008 B103h, ELSR3 0008 B104h, ELSR4 0008 B105h, ELSR7 0008 B108h, ELSR10 0008 B10Bh, ELSR12 0008 B10Dh, ELSR15 0008 B110h, ELSR16 0008 B111h, ELSR18 0008 B113h, ELSR19 0008 B114h, ELSR20 0008 B115h, ELSR21 0008 B116h, ELSR22 0008 B117h, ELSR23 0008 B118h, ELSR24 0008 B119h, ELSR25 0008 B11Ah, ELSR26 0008 B11Bh, ELSR27 0008 B11Ch, ELSR28 0008 B11Dh, ELSR29 0008 B11Eh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	b7 b0 00000000: Event output to the corresponding peripheral module is disabled. 00001000 to 01101001: 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 the event signal names set in the ELSRn register and the signal numbers.

Table 19.2 Correspondence between the ELSRn Register and the Peripheral Functions

Register Name	Peripheral Function (Module)
ELSR1	MTU1
ELSR2	MTU2
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR12	TMR2
ELSR15	12-bit A/D converter
ELSR16	DA0
ELSR18	Interrupt 1*1
ELSR19	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*2
ELSR25	Single port 1*2
ELSR26	Single port 2*2
ELSR27	Single port 3*2
ELSR28	Clock source switching to LOCO
ELSR29	POE

Note 1. Specify an event number from among 01100011b (63h) to 01101001b (69h). Do not set other settings.
Note 2. Do not set the DOC data operation condition met signal (01101010b (6Ah)) in the ELSR24, ELSR25, ELSR26, or ELSR27 register.

Table 19.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (1/2)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
08h	Multifunction timer pulse unit 2	MTU1 compare match 1A
09h		MTU1 compare match 1B
0Ah		MTU1 overflow
0Bh		MTU1 underflow
0Ch		MTU2 compare match 2A
0Dh		MTU2 compare match 2B
0Eh		MTU2 overflow
0Fh		MTU2 underflow
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
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
31h	Independent watchdog timer	IWDT underflow or refresh error
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I^2C bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end
52h	Serial peripheral interface	RSPI0 error (mode fault, overrun, or parity error)
53h		RSPI0 idle
54h		RSPI0 receive data full
55h		RSPI0 transmit data empty
56h	Serial peripheral interface	RSPI0 transmit end
58h	12-bit A/D converter	A/D conversion end of 12-bit A/D converter
5Bh	Voltage detection circuit	LVD1 voltage detection
5Ch		LVD2 voltage detection

Table 19.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (2/2)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
5Dh	DMA controller	DMAC0 transfer end
5Eh		DMAC1 transfer end
5Fh		DMAC2 transfer end
60h		DMAC3 transfer end
61h	Data transfer controller	DTC transfer end
62h	Clock generation circuit	Oscillation stop detection of clock generation circuit
63h	I/O ports	Input edge detection of input port group 1
64h		Input edge detection of input port group 2
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
67h		Input edge detection of single input port 2
68h		Input edge detection of single input port 3
69h	Event link controller	Software event
Settings other than above are prohibited.		

19.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): 0008 B11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU3MD[1:0]		MTU2MD[1:0]		MTU1MD[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	MTU1MD[1:0]	MTU1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event is disabled.	R/W
b5, b4	MTU2MD[1:0]	MTU2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event is disabled.	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*3 1 1: Event is disabled.	R/W

Note 1. The MTU1.TCNT value is captured into MTU1.TGRA.
Note 2. The MTU2.TCNT value is captured into MTU2.TGRA.
Note 3. The MTU3.TCNT value is captured into MTU3.TGRA.

ELOPA determines the operation of MTU1 to MTU3 when an event is input. The event setting should be disabled when the ELC function is not to be used.

19.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): 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 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 MTU4.TGRA.

ELOPB determines the operation of MTU4 when an event is input. The event setting should be disabled when the ELC function is not to be used.

19.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): 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 is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPC determines the operation of CMT1 when an event is input. The event setting should be disabled when the ELC function is not to be used.

19.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TMR2MD[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 is disabled.	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	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 is disabled.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPD determines the operation of TMR0 and TMR2 when an event is input. The event setting should be disabled when the ELC function is not to be used.

19.2.7 Port Group Setting Register n (PGRn) (n = 1, 2)

Address(es): PGR1 0008 B123h, 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: The port bit is not specified as a member of the same group. 1: The port bit is specified as a member of the same 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

PGRn specifies a group for I/O port bits. PGRn specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required. 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 (PGR)	Port Group Control Register (PGC)	Port Buffer Register (PDBF)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

19.2.8 Port Group Control Register n (PGCn) (n = 1, 2)

Address(es): PGC1 0008 B125h, 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 is generated upon detection of the rising edge of the external input signal. 0 1: Event is generated upon detection of the falling edge of the external input signal. 1 x: Event is generated upon detection of both the rising and falling edges of the external input signal.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting PDBFn register is disabled. 1: Overwriting 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: 0 is output when the event is input. 0 0 1: 1 is output when the event is input. 0 1 0: The toggled (inverted) value is output when the event is input. 0 1 1: The buffer value is output when the event is input. 1 x x: The bit value is rotated out in the group (from MSB to LSB) when the event 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 output port group, PGCn specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, PGCn enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).
Refer to Table 19.4 for the PGCn register and corresponding ports.

19.2.9 Port Buffer Register n (PDBFn) (n = 1, 2)

Address(es): PDBF1 0008 B127h, 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	Data is transferred between PODR and PDBF when an event is input. 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

PDBFn is an 8-bit readable/writable register used in combination with PGRn. Refer to section 19.3.5, I/O Port Operation upon Event Input and Event Generation for PDBFn operations. Refer to Table 19.4 for the PDBFn register and corresponding ports.

19.2.10 Event Link Port Setting Register n (PELn) (n = 0 to 3)

Address(es): PEL0 0008 B129h, PEL1 0008 B12Ah, PEL2 0008 B12Bh, PEL3 0008 B12Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSM[1:0]		PSP[1:0]		PSB[2:0]		
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	A bit number in eight I/O ports is specified.	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, data to be output from the port is specified. b6 b5 0 0: 0 is output when the event is input. 0 1: 1 is output when the event is input. 1 x: The toggled (inverted) value is output when the event is input. • For the input port, the edge on which the event is to be output is specified. b6 b5 0 0: Event is output upon detection of the rising edge. 0 1: Event is output upon detection of the falling edge. 1 x: Event 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

PELn specifies the single port to which an event is to be linked, the port operation upon the event signal input, and the conditions of event generation. In this MCU, a total of 4 bits in port B or port E can be specified as single ports.

19.2.11 Event Link Software Event Generation Register (ELSEGR)

Address(es): 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, the data will not be stored.

WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
[Setting condition]
If 1 is written to this bit while the WI bit is 0, this bit becomes 1.
[Clearing condition]
If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

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 bits to enable/disable these interrupt requests. When an interrupt request is generated in a module, the corresponding interrupt request status flag is set. If the corresponding interrupt request is enabled then, the interrupt requested is issued to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit.

Figure 19.2 shows the relation between the interrupt handling and ELC.

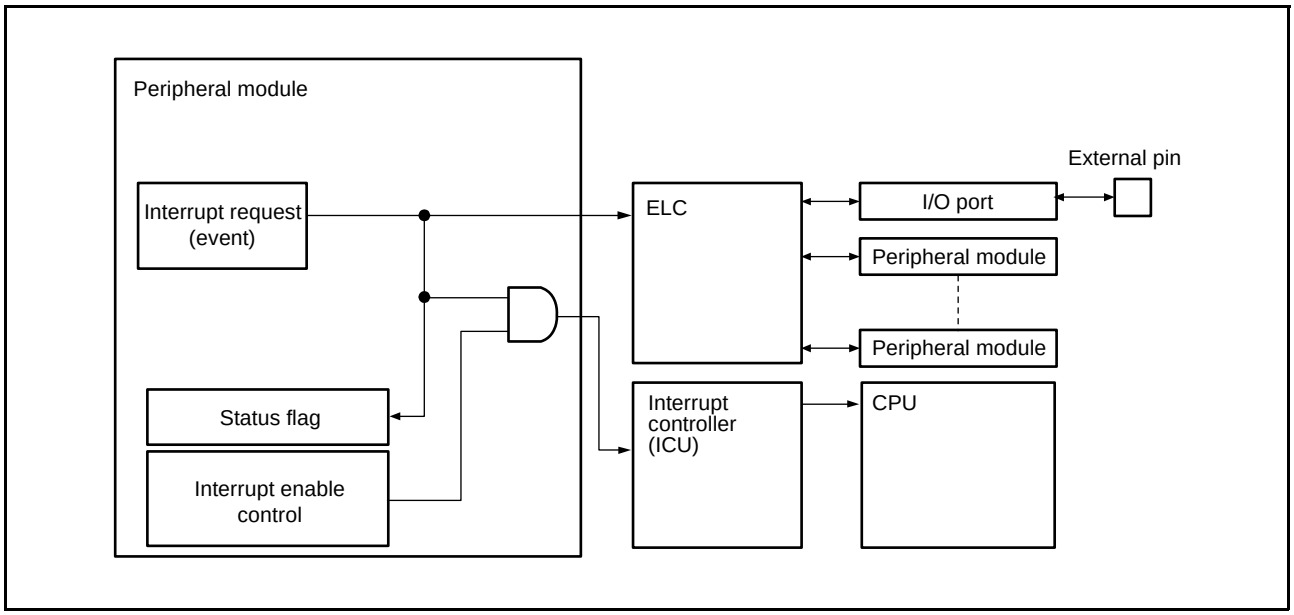


Figure 19.2 Relation between Interrupt Handling and ELC

19.3.2 Event Linkage

When an event has been set as a trigger in an event link setting register (ELSRn) and then occurs, the corresponding module is activated. Only one type of event can be connected with one module. When a module is to be activated by the ELC, the operation of the module must be set up in advance. Table 19.5 lists the operations of modules when an event is input.

Table 19.5 Operations of Modules When Event is Input

Module	Operations When Event is Input		
MTU CMTTMR	Each timer operates differently depending on the ELOPA to ELOPD registers as below. • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (CMT, TMR). • Performs input-capture operation when an event is input (MTU).		
POE	Places the MTU complementary PWM output pins and MTU0 output pins in the high-impedance state 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 (port output data register) changes when an event signal is input. (The value output from the relevant external pin changes.)	Port group	• Changes the PODR value to the specified value. • Transfers the PDBFn value to the PODR register. • Rotates out the bit value.
		Single port	Changes the PODR value to the specified value.
I/O ports (input)	When the signal value of the input pin changes	Port group	Generates an event.
		Single port	
	When an event is input	Port group	Transfers the signal value of the external pin to the PDBFn register.
		Single port	Event connection is not possible.
Clock generation circuit	Switches the clock source to the low-speed on-chip oscillator when an event signal is input.*1		
Interrupt controller	Issues an event to the CPU, starts DMAC data transfer, and starts DTC data 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 is Input

The operations are performed depending on the ELOPA to ELOPD registers when an event is input.

(1) Count Start Operation

When an event is input, the timer starts counting, which sets the count start bit*1 in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

(2) Count Restart Operation

When an event is input, the timer counter*1 is initialized. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event is input while the count start bit is set to 1.

(3) Event Counter Operation

Event input is selected as the timer clock source and the timer counts events.

(4) Input Capture Operation

When an event 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 A/D and D/A Converters When Event is Input

The A/D and D/A converter start A/D and D/A conversion, respectively, when the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

19.3.5 I/O Port Operation upon Event Input and Event Generation

The I/O port operation to be performed upon event input and the operation to generate an event can be set.

(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 connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.
A single port can be set by specifying any bit in the I/O port*1 to which an event can be connected using the PEL0 to PEL3 registers. A port group can be set by specifying any 2 or more bits in the I/O port*1 to which an event can be connected using the PGCn register. One input port group and one output port group can be set in the same I/O port.
If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.
Set the PDR register to select the direction of the I/O ports.

Note 1. Port B and port E

(2) Event Generation by Single Input Ports

An single input port generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the PEL0 to PEL3 registers. An example of operation is shown in Figure 19.3.

(3) Single Output Ports Operation upon Event Input

When an event is input to a single output port, the signal of the external pin connected to the relevant port changes according to the settings of the PEL0 to PEL3 registers. This changes the signal value of the external pin connected to the relevant port. An example of operation is shown in Figure 19.3.

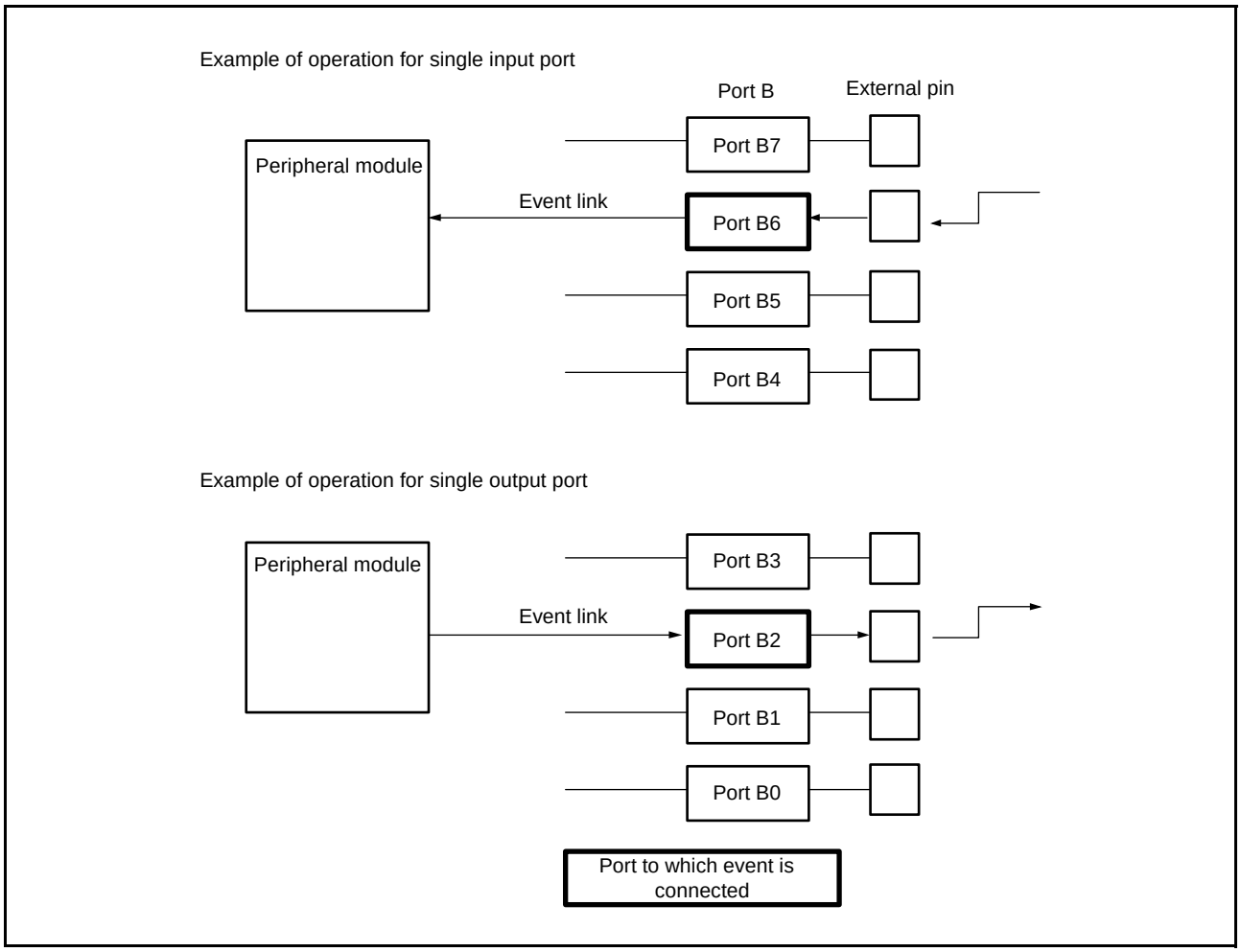


Figure 19.3 Event Linkage Related to Single Ports (Port B)

(4) Input Port Group Operation upon Event Input and Event Generation

An input port group generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGCn register. When an event is input to an input port group, the signal value of the external pin upon event input is transferred to the PDBFn register. In this case, only the values of the bits specified as members of the input port group are transferred. An example of operation is shown in Figure 19.4.

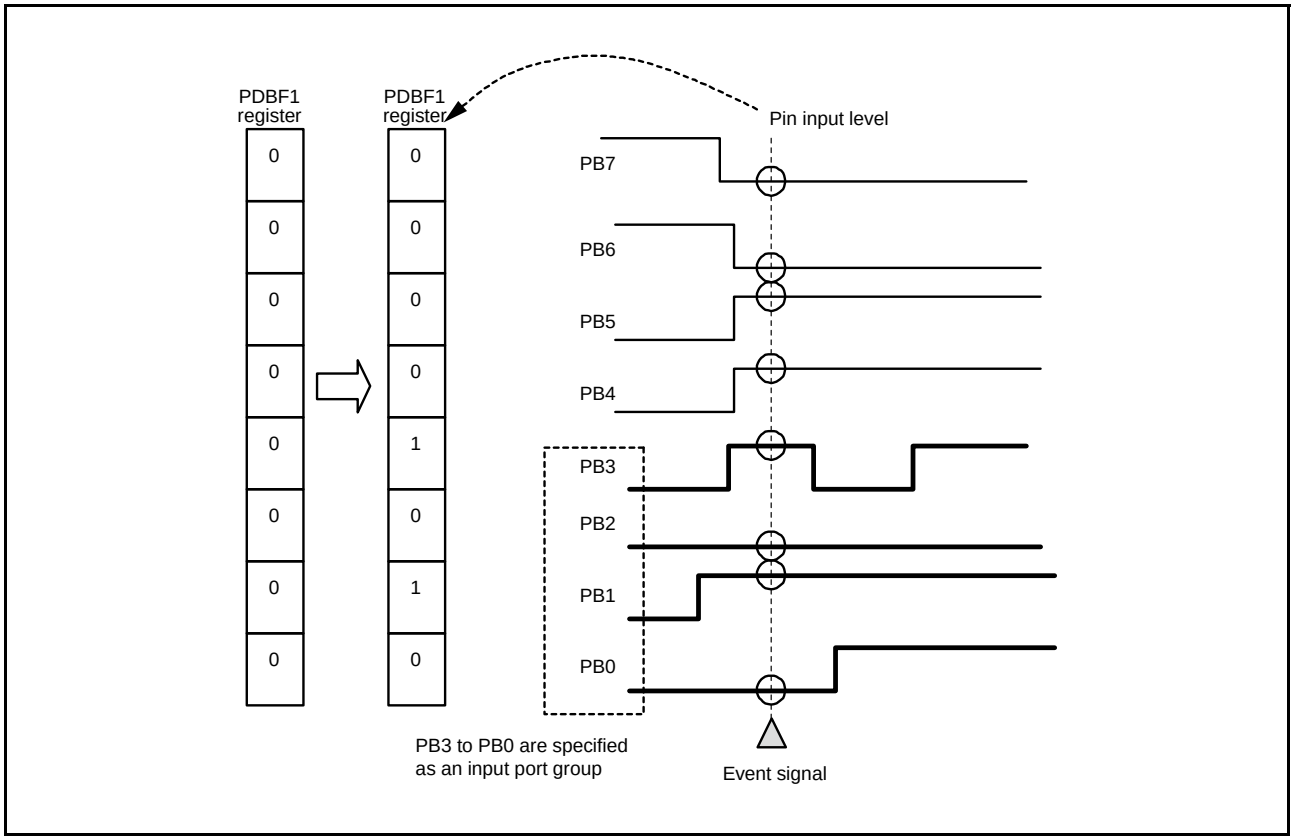


Figure 19.4 Event Linkage Related to Input Port Groups (Port B)

(5) Output Port Group Operation upon Event Input

When an event is input to an output port group, the PODR values change to the values according to the PGCn settings. An example of operation is shown in Figure 19.5.

(6) Operation of Port Buffer Registers

(a) Input Port Groups

When an event is input to an input port group, the signal value of the external pin of the bit specified as a member of the input port group is transferred to the PDBFn register. If another event is input to the input port group in this state, operations are performed depending on the PGCn.PGCOVE bit setting as described below.

- PGCn.PGCOVE = 0 (overwriting is disabled)
  If the PDBFn value that has been transferred upon the latest event input has already been read by the CPU (or transferred by the DTC), the signal value of the external pin is transferred to the PDBFn register. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGCn.PGCOVE = 1 (overwriting is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to the PDBFn register.

(b) Output Port Groups

If an output port group is specified so that it should output the PDBFn value, the PDBFn value is transferred to the PODR register when an event is input to the output port group. In this case, only the values of the bits specified as members of the output port group are transferred.
If an output port group is specified so that it should rotate out the bit values in the group (PGCn.PGCO[2:0] bits = 1xxb), the PDBFn data is transferred to the PODR register, and then the PODR value is rotated bit by bit from MSB to LSB. The

initial value to be output to the port group should be provided in the PDBFn register.
Examples of operation are shown in Figure 19.5 and Figure 19.6.

(7) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is set to 1, write access to the following registers is disabled.

- If bits are specified as members of the input port group and the event linkage is set for the port group, write access to the relevant bits in the PDBFn register is disabled. However, when the DOC is selected for event input, write access is enabled.
- If port bits are specified as members of the output port group, write access to the relevant bits in the PODR register is disabled.
- If a port bit is specified as a single output port and the event linkage is set (by the ELSRn register) for the port, write access to the relevant bit in the PODR register is disabled. However, when the DOC is selected for event input, write access is enabled.

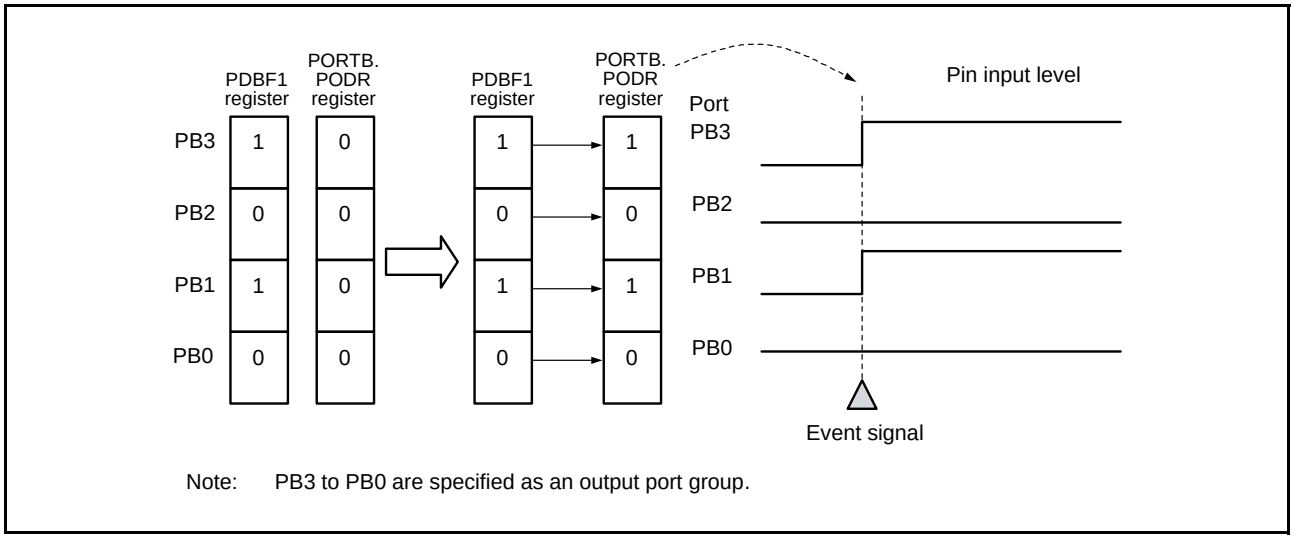
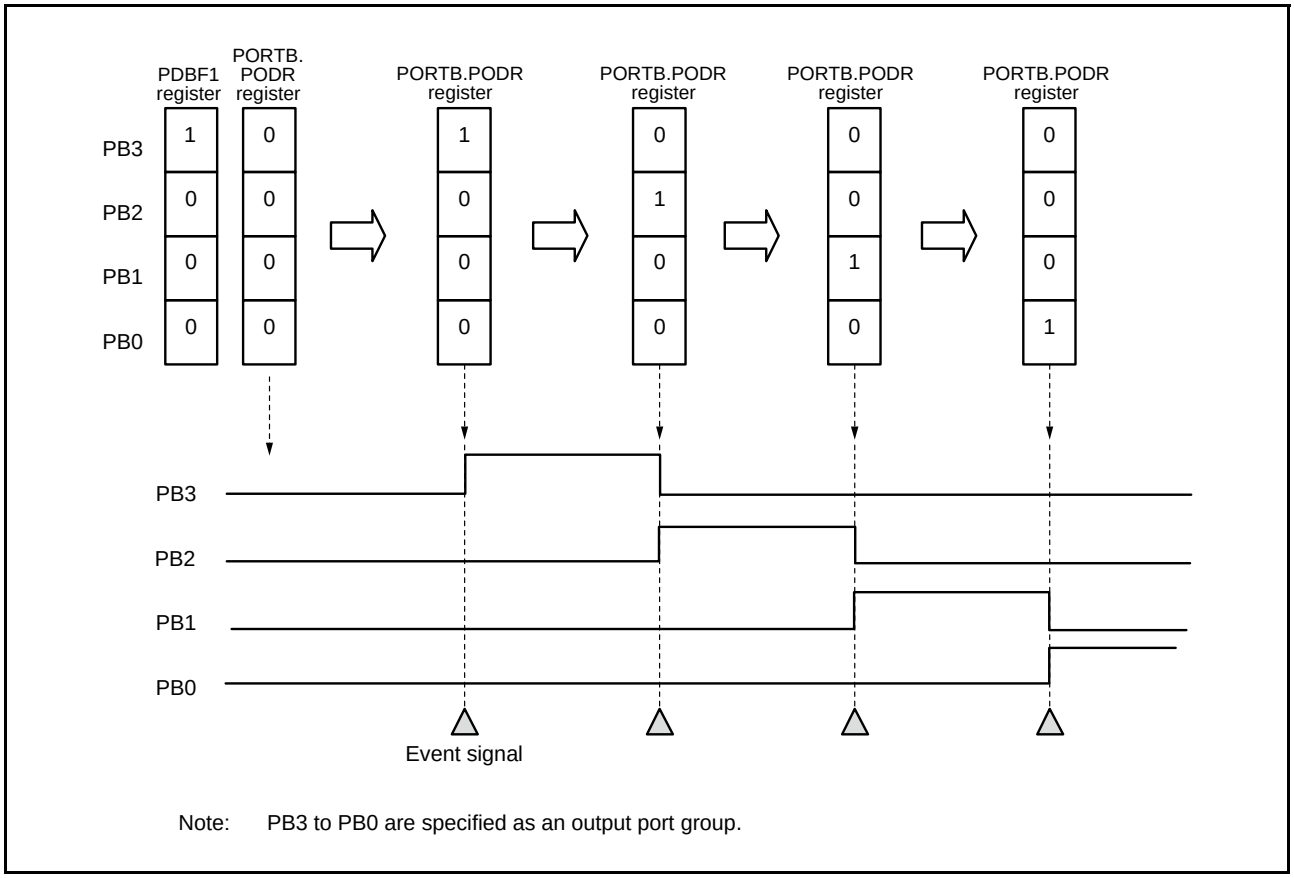


Figure 19.5 Event Linkage Related to Output Port Groups (Port B)



Note: PB3 to PB0 are specified as an output port group.

Figure 19.6 Bit-Rotating Operation of Output Port Groups (Port B)

19.3.6 Example of Procedure for Linking Events

The following describes the procedure for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers corresponding to the ports as below.
   PODR: Set the initial values of the output ports.
   PDR: Set the I/O direction of the ports.
   PGRn: If ports are used as a port group, set the ports (in bit units) to be grouped.
   PGCn: Set the operation of the port group.
   PELn: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
3. To the ELSRn register corresponding to the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA to ELOPD registers corresponding to the timers as required.
5. Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
6. Set the operation of the module from which an event is output, and activate the module. This allows the event output from the module to start the module to which an event is linked as preset.
7. To stop event linkage of independent modules, set 00000000b to the ELSRn.ELS[7:0] bits corresponding to the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

19.4 Usage Notes

19.4.1 Setting ELSRn Register

(1) Setting ELSR18 and ELSR19 Registers

Specify an event number from among 01100011b (63h) to 01101001b (69h). Do not set other settings.

(2) Setting ELSR24, ELSR25, ELSR26, and ELSR27 Registers

Do not set the DOC data operation condition met signal (01101010b (6Ah)).

19.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBFn register are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. When events are used during bit-rotating operation, generate an event after an interval of one PCLKB cycle. If not, the normal operation cannot be provided.

19.4.3 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is completed.

19.4.4 Setting Clocks

To link events, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module stop state or in the specific low power consumption mode in which the module is stopped (all-module clock stop mode, software standby mode, or deep software standby mode).

19.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting allows the ELC to be stopped. 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 I/O ports function as a general I/O port, an I/O pin of a peripheral module, an input pin for an interrupt, or a bus control pin.

Some of the pins are 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 on-chip peripheral modules.

Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, the driving ability control register (DSCR) that selects the driving ability, and the port mode register (PMR) that specifies the pin function of each port.

For details on the PMR register, see section 21, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the product. Table 20.1 lists the specifications of I/O ports, and Table 20.2 lists the port functions.

Table 20.1 Specifications of I/O Ports

	Voltage Specification		Voltage Specification	
Port Symbol	**3-V Package**	**Number of Pins**	**5-V Package**	**Number of Pins**
PORT0	P00 to P03, P05, P07	6	P00 to P03, P05, P07	6
PORT1	P12 to P17	6	P12 to P17	6
PORT2	P20 to P27	8	P20 to P27	8
PORT3	P30 to P35	6	P30 to P35	6
PORT4	P40 to P47	8	P40 to P47	8
PORT5	P50 to P56	7	P50 to P56	7
PORT6	P60 to P67	8	P60 to P67	8
PORT7	P70 to P72, P74 to P77	7	P70 to P77	8
PORT8	P80 to P83, P86, P87	6	P80 to P83, P86, P87	6
PORT9	P90 to P93	4	P90 to P93	4
PORTA	PA0 to PA7	8	PA0 to PA7	8
PORTB	PB0 to PB7	8	PB0 to PB7	8
PORTC	PC0 to PC7	8	PC0 to PC7	8
PORTD	PD0 to PD7	8	PD0 to PD7	8
PORTE	PE0 to PE7	8	PE0 to PE7	8
PORTF	PF5	1	PF5	1
PORTH	PH0 to PH3	4	PH0 to PH3	4
PORTJ	PJ1 to PJ5	5	PJ1 to PJ5	5
PORTK	PK2 to PK5	4	PK2 to PK5	4
PORTL	PL0, PL1, PL5	3	PL0, PL1	2
	Total of pins	123	Total of pins	123

Table 20.2 Port Functions

Port Symbol	Port	Input Pull-up	Open Drain Output	Driving Ability Switching
PORT0	P00 to P02	○	○	○
	P03, P05, P07	—	—	Fixed to normal output
PORT1	P12 to P17	○	○	○
PORT2	P20 to P27	○	○	○
PORT3	P30 to P34	○	○	○
	P35	—	—	—
PORT4	P40 to P47	—	—	—
PORT5	P50 to P52, P54	○	○	○
	P53, P55, P56	○	—	○
PORT6	P60, P61	○	○	○
	P62 to P67	○	—	Fixed to normal output
PORT7	P70, P74 to P77	○	○	○
	P71 to P73	○	—	Fixed to normal output
PORT8	P80 to P83	○	○	○
	P86, P87	○	—	○
PORT9	P90 to P93	○	○	○
PORTA	PA0 to PA7	○	○	○
PORTB	PB0 to PB7	○	○	○
PORTC	PC0 to PC7	○	○	○
PORTD	PD0 to PD7	○	—	○
PORTE	PE0 to PE7	○	○	○
PORTF	PF5	○	—	Fixed to normal output
PORTH	PH0 to PH3	○	—	○
PORTJ	PJ1, PJ3	○	—	○
	PJ2, PJ4, PJ5	○	—	Fixed to normal output
PORTK	PK2 to PK5	○	○	○
PORTL	PL0, PL1	○	—	Fixed to normal output
	PL5	—	○*1	Fixed to normal output

Note 1. The function of the PL5 pin is NMOS open drain output only and cannot be changed.

Specifying input pull-up, open-drain output, or switching of driving ability is available for other signals on pins that also function as general I/O pins.

20.2 I/O Port Configuration

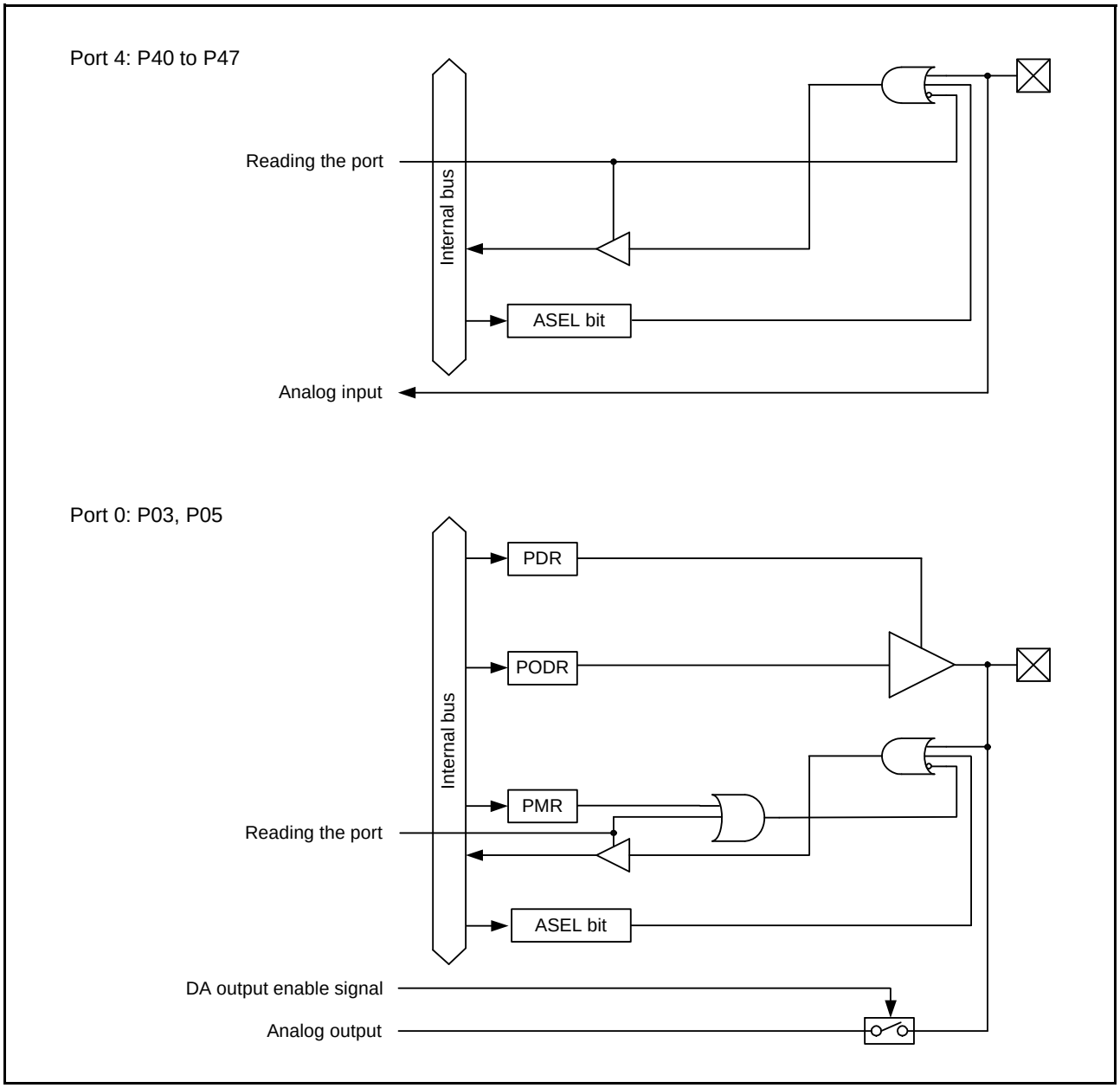
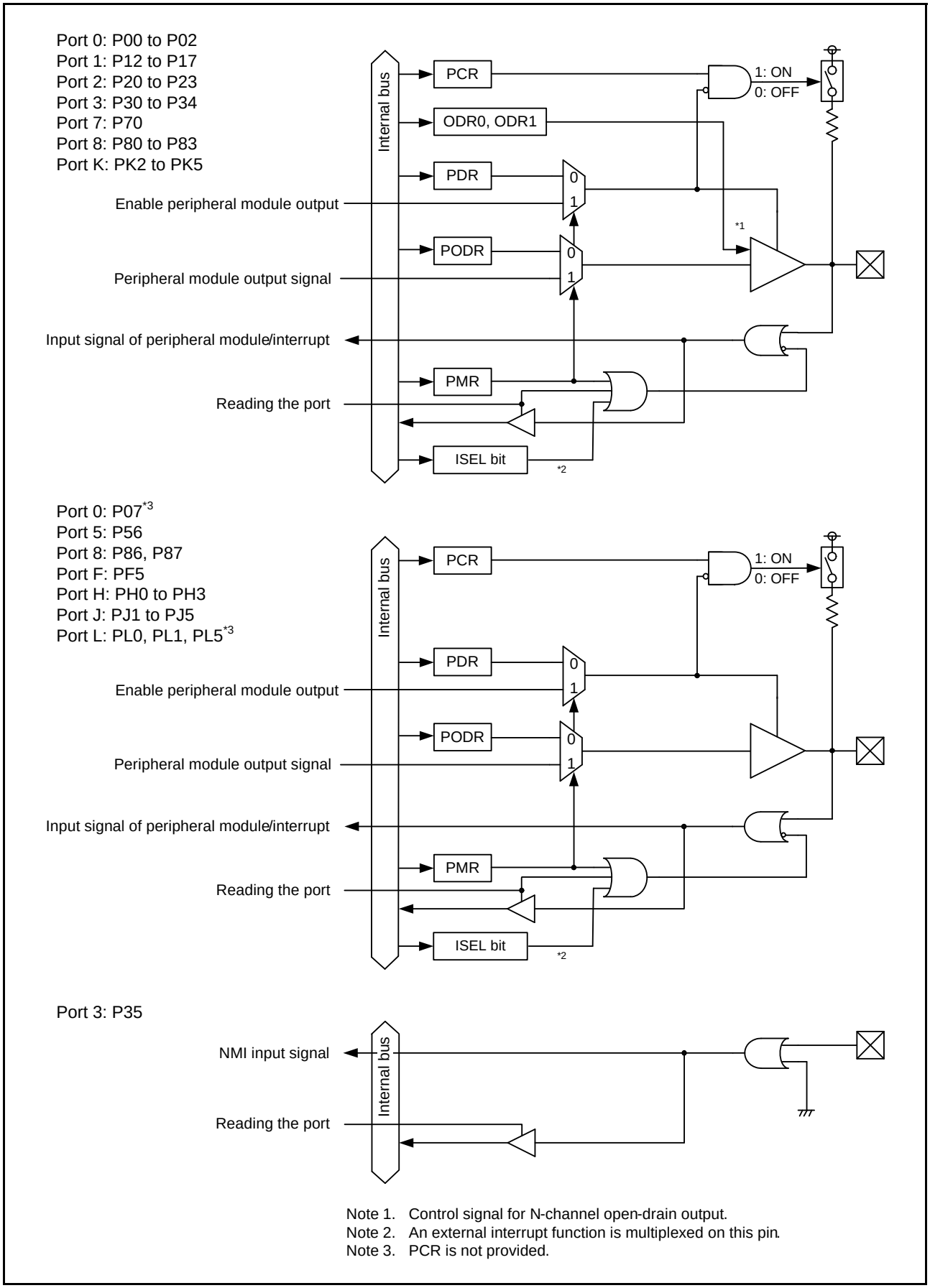
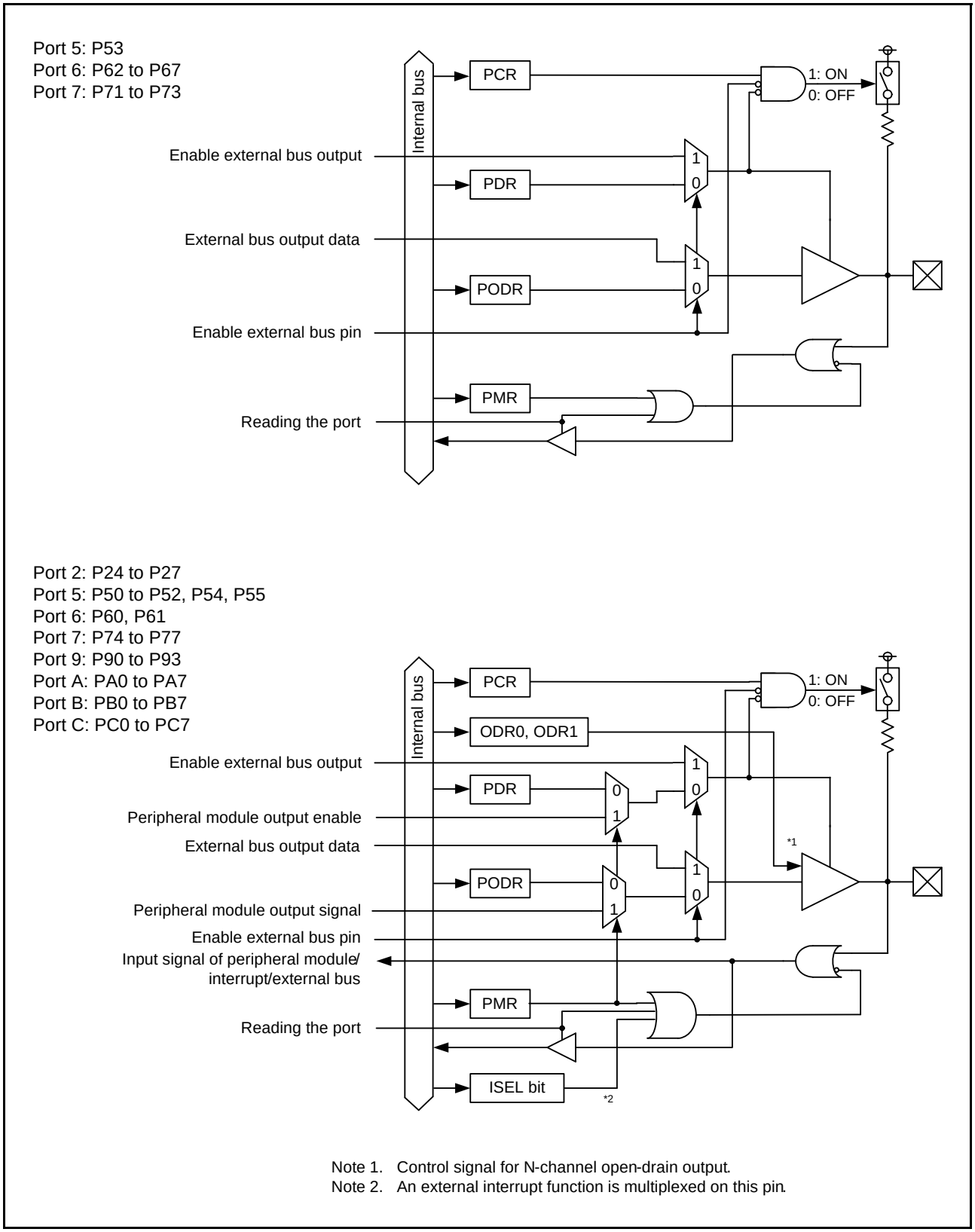


Figure 20.1 I/O Port Configuration (1)



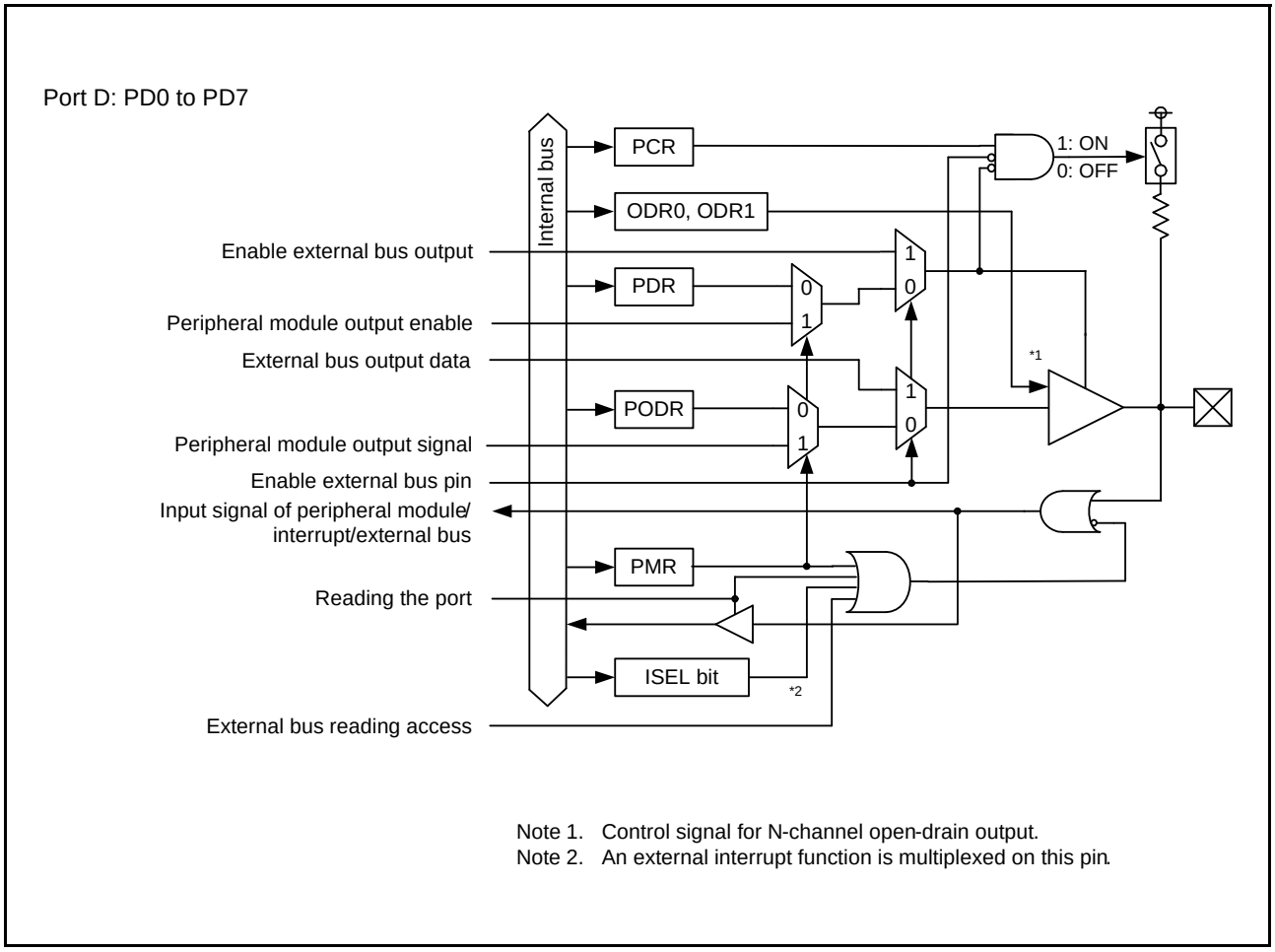
Note 1. Control signal for N-channel open-drain output.
Note 2. An external interrupt function is multiplexed on this pin.
Note 3. PCR is not provided.

Figure 20.2 I/O Port Configuration (2)



Note 1. Control signal for N-channel open-drain output.
Note 2. An external interrupt function is multiplexed on this pin.

Figure 20.3 I/O Port Configuration (3)



Note 1. Control signal for N-channel open-drain output.
Note 2. An external interrupt function is multiplexed on this pin.

Figure 20.4 I/O Port Configuration (4)

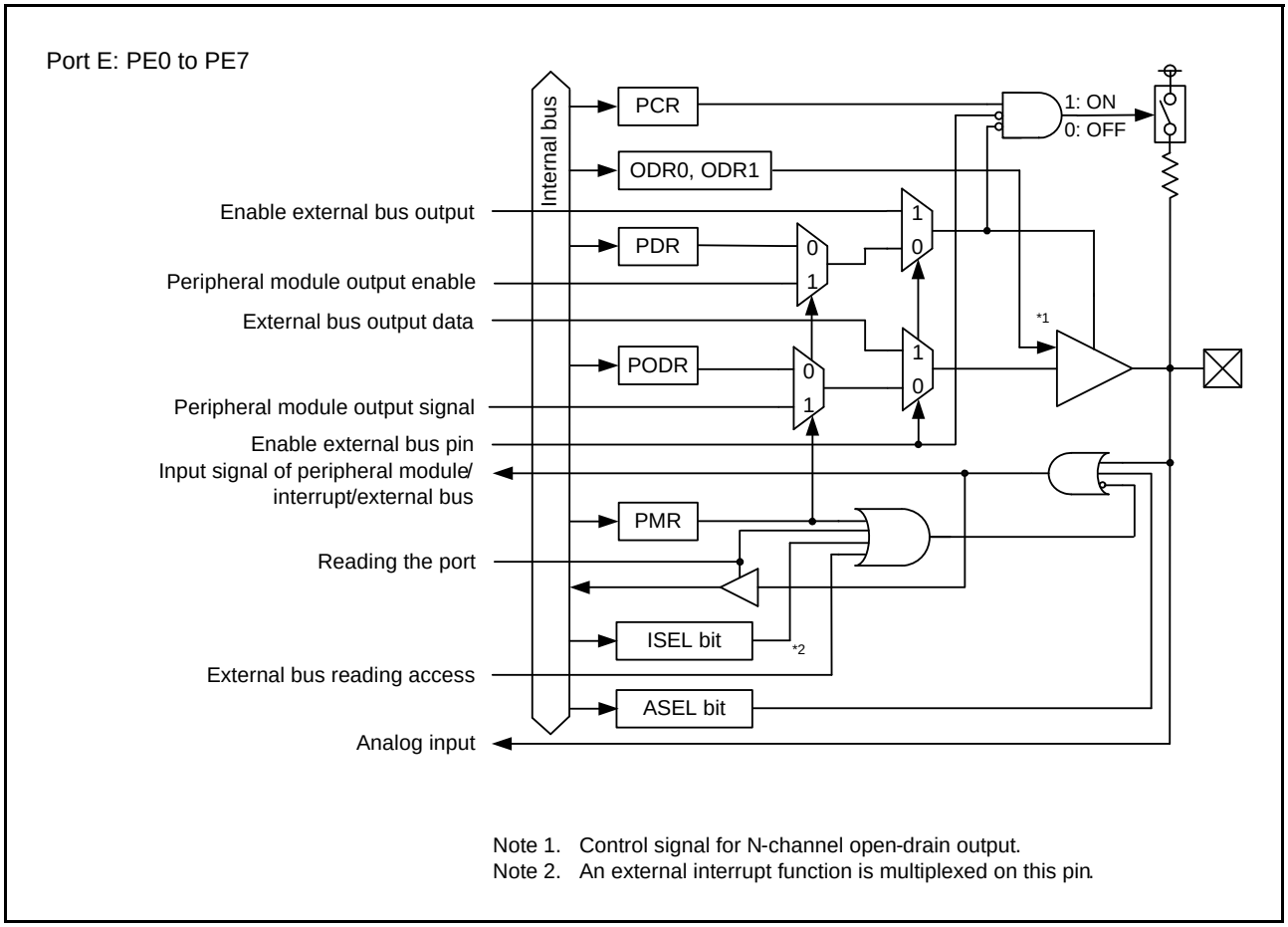


Figure 20.5 I/O Port Configuration (5)

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, 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, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTF.PDR 0008 C00Fh, PORTH.PDR 0008 C011h, PORTJ.PDR 0008 C012h, PORTK.PDR 0008 C013h, PORTL.PDR 0008 C014h

	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 3, 5 to 9, A to F, H, J to L

PDR is used to select the input or output direction for individual pins of the corresponding port m 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.

The PORT3.PDR.B5 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should be 0.

The P73 pin does not exist in the 3-V packages. The PL5 pin does not exist in the 5-V packages. The bit corresponding to a pin which does not exist is also reserved. Write 1 (output) to this bit.

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, 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, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTF.PODR 0008 C02Fh, PORTH.PODR 0008 C031h, PORTJ.PODR 0008 C032h, PORTK.PODR 0008 C033h, PORTL.PODR 0008 C034h

	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	Holds output data.	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 3, 5 to 9, A to F, H, J to L

PODR holds the data to be output from the pins used for general output ports.
The PORT3.PODR.B5 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

20.3.3 Port Input Data 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, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTF.PIDR 0008 C04Fh, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h, PORTK.PIDR 0008 C053h, PORTL.PIDR 0008 C054h

	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	Indicates individual pin states of the corresponding port.	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 to F, H, J to L

PIDR indicates individual pin states of port m.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the P35 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, 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, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTF.PMR 0008 C06Fh, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h, PORTK.PMR 0008 C073h, PORTL.PMR 0008 C074h

	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: Use pin as general I/O port. 1: Use pin as I/O port for peripheral functions.	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 to F, H, J to L

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 read as 0. The write value should 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, 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, PORTC.ODR0 0008 C098h, PORTE.ODR0 0008 C09Ch, PORTK.ODR0 0008 C0A6h

	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 Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm1 Output Type Select	• P01, P21, P31, P51, P61, P81, P91, PA1, PB1, PC1 b2 0: CMOS output 1: N-channel open-drain b3 This bit is read as 0. The write value should be 0. • PE1 b3 b2 0 0: CMOS output 0 1: N-channel open-drain 1 0: P-channel open-drain 1 1: Hi-Z	R/W
b3	B3			R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 0 to 3, 6 to 9, A to C, E, K

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

20.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT5.ODR0 0008 C08Bh, PORT7.ODR1 0008 C08Fh, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTE.ODR1 0008 C09Dh, PORTK.ODR1 0008 C0A7h

	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	Pm4 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1 to 3, 7, A to C, E, K

The PORT3.ODR1.B2 bit is reserved, because the P35 pin is input only.
The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

20.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, 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, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTH.PCR 0008 C0D1h, PORTJ.PCR 0008 C0D2h, PORTK.PCR 0008 C0D3h, PORTL.PCR 0008 C0D4h

	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 3, 5 to 9, A to F, H, J to L

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.

PCR retains pull-up state even when a transition to deep software standby mode is made.

When a pin is used as an external bus pin other than the WAIT# pin, a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the settings of the PCR register.

The pull-up resistor is also disabled in the reset state.

The B5 bit in PORT3.PCR is reserved. The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should 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, PORT5.DSCR 0008 C0E5h, PORT6.DSCR 0008 C0E6h, PORT7.DSCR 0008 C0E7h, PORT8.DSCR 0008 C0E8h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTH.DSCR 0008 C0F1h, PORTJ.DSCR 0008 C0F2h, PORTK.DSCR 0008 C0F3h

	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, 5 to 9, A to E, H, J, K

The bit corresponding to a pin with the fixed drive capacity can be read from or written to. However, the drive capacity cannot be changed.

When high-drive output is selected, switching noise increases compared to when normal output is selected. Carefully evaluate the effect of noise on the MCU caused by adjacent pins before selecting high-drive output.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

20.4 Unused Pin Configuration

The configuration of unused pins is listed in Table 20.3.

Table 20.3 Unused Pin Configuration

Pin Name	Description
EMLE	Connect this pin to VSS via a pull-down resistor.
MD	(Used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
P35/NMI	Connect this pin to VCC via a pull-up resistor.
EXTAL	Connect this pin to VSS via a pull-down resistor.
XTAL	When the external clock is input to the EXTAL pin, leave this pin open.
Ports 0 to 9 Ports A to F, H, J to L	• 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), the pin is released.*1, *2
VREFH0	Connect this pin to VCC.
VREFL0	Connect this pin to VSS.
VREFH	Connect this pin to VCC.
VREFL	Connect this pin to VSS.

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.

20.5 Usage Notes

- Port setting when A/D converter input is used

If either one of port 0 and port E is used as an input pin for the 12-bit A/D converter, output from port 07 and port 4 should not be used. This is because these ports use the analog power supply for the A/D converter.

- Port setting when D/A converter output is used

If either one of port 05 and port 03 is used as an output pin for the D/A converter, output from these ports should not be used. This is because these ports use the analog power supply for the D/A converter.

21. Multi-Function Pin Controller (MPC)

21.1 Overview

The multi-function pin controller (MPC) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports. It is also used to allocate external-bus related signals to port pins. Table 21.1 shows the allocation of pin functions to multiple pins. Allocating the same function to more than one pin is prohibited.

Table 21.1 Allocation of Pin Functions to Multiple Pins (1 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Interrupt		NMI (input)	P35
Interrupt	IRQ0	IRQ0-DS (input)	P30
		IRQ0 (input)	PD0
			PH1
	IRQ1	IRQ1-DS (input)	P31
		IRQ1 (input)	PD1
			PH2
	IRQ2	IRQ2-DS (input)	P32
		IRQ2 (input)	P12
			PD2
	IRQ3	IRQ3-DS (input)	P33
		IRQ3 (input)	P13
			PD3
	IRQ4	IRQ4-DS (input)	PB1
		IRQ4 (input)	P14
			P34
			PD4
			PF5
	IRQ5	IRQ5-DS (input)	PA4
		IRQ5 (input)	P15
			PD5
			PE5
	IRQ6	IRQ6-DS (input)	PA3
		IRQ6 (input)	P16
			PD6
			PE6
	IRQ7	IRQ7-DS (input)	PE2
		IRQ7 (input)	P17
			PD7
			PE7
	IRQ8	IRQ8 (input)	P00
			P20
	IRQ9	IRQ9 (input)	P01
			P21
	IRQ10	IRQ10 (input)	P02
			P55

Table 21.1 Allocation of Pin Functions to Multiple Pins (2 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Interrupt	IRQ11	IRQ11 (input)	P03
			PA1
	IRQ12	IRQ12 (input)	P73*1
			PB0
			PC1
			PL5*2
Multi-function timer unit 2	MTU0	MTIOC0A (input/output)	P34
			PB3
		MTIOC0B (input/output)	P13
			P15
			PA1
		MTIOC0C (input/output)	P32
			PB1
		MTIOC0D (input/output)	P33
			PA3
	MTU1	MTIOC1A (input/output)	P20
			PE4
		MTIOC1B (input/output)	P21
			PB5
	MTU2	MTIOC2A (input/output)	P26
			PB5
		MTIOC2B (input/output)	P27
			PE5
	MTU3	MTIOC3A (input/output)	P14
			P17
			PC1
			PC7
			PJ1
		MTIOC3B (input/output)	P17
			P22
			P80
			PB7
			PC5
		MTIOC3C (input/output)	P16
			P56
			PC0
			PC6
			PJ3
		MTIOC3D (input/output)	P16
			P23
			P81
			PB6
			PC4

Table 21.1 Allocation of Pin Functions to Multiple Pins (3 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Multi-function timer unit 2	MTU4	MTIOC4A (input/output)	P24
			P82
			PA0
			PB3
			PE2
		MTIOC4B (input/output)	P30
			P54
			PC2
			PD1
			PE3
		MTIOC4C (input/output)	P25
			P83
			PB1
			PE1
			PE5
		MTIOC4D (input/output)	P31
			P55
			PC3
			PD2
			PE4
	MTU5	MTIC5U (input)	PA4
			PD7
		MTIC5V (input)	PA6
			PD6
		MTIC5W (input)	PB0
			PD5
	MTU	MTCLKA (input)	P14
			P24
			PA4
			PC6
		MTCLKB (input)	P15
			P25
			PA6
			PC7
		MTCLKC (input)	P22
			PA1
			PC4
		MTCLKD (input)	P23
			PA3
			PC5
Port output enable 2	POE0	POE0# (input)	PC4
			PD7
	POE1	POE1# (input)	PB5
			PD6

Table 21.1 Allocation of Pin Functions to Multiple Pins (4 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Port output enable 2	POE2	POE2# (input)	P34
			PA6
			PD5
	POE3	POE3# (input)	P33
			PB3
			PD4
	POE8	POE8# (input)	P17
			P30
			PD3
			PE3
16-bit timer pulse unit	TPU0	TIOCA0 (input/output)	P86
			PA0
		TIOCB0 (input/output)	P17
			PA1
		TIOCC0 (input/output)	P32
		TIOCD0 (input/output)	P33
			PA3
	TPU1	TIOCA1 (input/output)	P56
			PA4
		TIOCB1 (input/output)	P16
			PA5
	TPU2	TIOCA2 (input/output)	P87
			PA6
		TIOCB2 (input/output)	P15
			PA7
	TPU3	TIOCA3 (input/output)	P21
			PB0
		TIOCB3 (input/output)	P20
			PB1
		TIOCC3 (input/output)	P22
			PB2
		TIOCD3 (input/output)	P23
			PB3
	TPU4	TIOCA4 (input/output)	P25
			PB4
		TIOCB4 (input/output)	P24
			PB5
	TPU5	TIOCA5 (input/output)	P13
			PB6
		TIOCB5 (input/output)	P14
			PB7
	TPU	TCLKA (input)	P14
			PC2

Table 21.1 Allocation of Pin Functions to Multiple Pins (5 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
16-bit timer pulse unit	TPU	TCLKB (input)	P15
			PA3
			PC3
		TCLKC (input)	P16
			PB2
			PC0
		TCLKD (input)	P17
			PB3
			PC1
Programmable pulse generator	PPG0	PO0 (output)	P20
		PO1 (output)	P21
		PO2 (output)	P22
		PO3 (output)	P23
		PO4 (output)	P24
		PO5 (output)	P25
		PO6 (output)	P26
		PO7 (output)	P27
		PO8 (output)	P30
		PO9 (output)	P31
		PO10 (output)	P32
		PO11 (output)	P33
		PO12 (output)	P34
		PO13 (output)	P13
			P15
		PO14 (output)	P16
		PO15 (output)	P14
			P17
8-bit timer	TMR0	TMO0 (output)	P22
			PB3
			PH1
		TMCI0 (input)	P01
			P21
			PB1
			PH3
		TMRI0 (input)	P00
			P20
			PA4
			PH2
	TMR1	TMO1 (output)	P17
			P26
		TMCI1 (input)	P02
			P12
			P54
			PC4

Table 21.1 Allocation of Pin Functions to Multiple Pins (6 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
8-bit timer	TMR1	TMRI1 (input)	P24
			PB5
	TMR2	TMO2 (output)	P16
			PC7
		TMCI2 (input)	P15
			P31
			PC6
		TMRI2 (input)	P14
			PC5
	TMR3	TMO3 (output)	P13
			P32
			P55
		TMCI3 (input)	P27
			P34
			PA6
		TMRI3 (input)	P30
			P33
Serial communications interface	SCI0	RXD0 (input)/ SMISO0 (input/output)/ SSCL0 (input/output)	P21
			P33
		TXD0 (output)/ SMOSI0 (input/output)/ SSDA0 (input/output)	P20
			P32
		SCK0 (input/output)	P22
			P34
		CTS0# (input)/ RTS0# (output)/ SS0# (input)	P23
			PJ3
	SCI1	RXD1 (input)/ SMISO1 (input/output)/ SSCL1 (input/output)	P15
			P30
		TXD1 (output)/ SMOSI1 (input/output)/ SSDA1 (input/output)	P16
			P26
		SCK1 (input/output)	P17
			P27
		CTS1# (input)/ RTS1# (output)/ SS1# (input)	P14
			P31
	SCI2	RXD2 (input)/ SMISO2 (input/output)/ SSCL2 (input/output)	P12
			P52
		TXD2 (output)/ SMOSI2 (input/output)/ SSDA2 (input/output)	P13
			P50
		SCK2 (input/output)	P51
		CTS2# (input)/ RTS2# (output)/ SS2# (input)	P54

Table 21.1 Allocation of Pin Functions to Multiple Pins (7 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Serial communications interface	SCI3	RXD3 (input)/ SMISO3 (input/output)/ SSCL3 (input/output)	P16
			P25
		TXD3 (output)/ SMOSI3 (input/output)/ SSDA3 (input/output)	P17
			P23
		SCK3 (input/output)	P15
			P24
		CTS3# (input)/ RTS3# (output)/ SS3# (input)	P26
	SCI4	RXD4 (input)/ SMISO4 (input/output)/ SSCL4 (input/output)	PB0
			PK4
		TXD4 (output)/ SMOSI4 (input/output)/ SSDA4 (input/output)	PB1
			PK5
		SCK4 (input/output)	P70
			PB3
		CTS4# (input)/ RTS4# (output)/ SS4# (input)	PB2
			PE6
	SCI5	RXD5 (input)/ SMISO5 (input/output)/ SSCL5 (input/output)	PA2
			PA3
			PC2
		TXD5 (output)/ SMOSI5 (input/output)/ SSDA5 (input/output)	PA4
			PC3
		SCK5 (input/output)	PA1
			PC1
			PC4
		CTS5# (input)/ RTS5# (output)/ SS5# (input)	PA6
			PC0
	SCI6	RXD6 (input)/ SMISO6 (input/output)/ SSCL6 (input/output)	P01
			P33
			PB0
		TXD6 (output)/ SMOSI6 (input/output)/ SSDA6 (input/output)	P00
			P32
			PB1
		SCK6 (input/output)	P02
			P34
			PB3
		CTS6# (input)/ RTS6# (output)/ SS6# (input)	PB2
			PJ3
	SCI7	RXD7 (input)/ SMISO7 (input/output)/ SSCL7 (input/output)	P92
		TXD7 (output)/ SMOSI7 (input/output)/ SSDA7 (input/output)	P90

Table 21.1 Allocation of Pin Functions to Multiple Pins (8 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Serial communications interface	SCI7	SCK7 (input/output)	P91
		CTS7# (input)/ RTS7# (output)/ SS7# (input)	P93
	SCI8	RXD8 (input)/ SMISO8 (input/output)/ SSCL8 (input/output)	PC6
		TXD8 (output)/ SMOSI8 (input/output)/ SSDA8 (input/output)	PC7
		SCK8 (input/output)	PC5
		CTS8# (input)/ RTS8# (output)/ SS8# (input)	PC4
	SCI9	RXD9 (input)/ SMISO9 (input/output)/ SSCL9 (input/output)	PB6
			PK3
		TXD9 (output)/ SMOSI9 (input/output)/ SSDA9 (input/output)	PB7
			PK2
		SCK9 (input/output)	P60
			PB5
		CTS9# (input)/ RTS9# (output)/ SS9# (input)	P61
			PB4
	SCI10	RXD10 (input)/ SMISO10 (input/output)/ SSCL10 (input/output)	P81
		TXD10 (output)/ SMOSI10 (input/output)/ SSDA10 (input/output)0	P82
		SCK10 (input/output)	P80
		CTS10# (input)/ RTS10# (output)/ SS10# (input)	P83
	SCI11	RXD11 (input)/ SMISO11 (input/output)/ SSCL11 (input/output)	P76
		TXD11 (output)/ SMOSI11 (input/output)/ SSDA11 (input/output)0	P77
		SCK11 (input/output)	P75
		CTS11# (input)/ RTS11# (output)/ SS11# (input)	P74
	SCI12	RXD12 (input)/ SMISO12 (input/output)/ SSCL12 (input/output)/ RXDX12 (input)	PE2
		TXD12 (output)/ SMOSI12 (input/output)/ SSDA12 (input/output)/ TXDX12 (output)/ SIOX12 (input/output)	PE1
		SCK12 (input/output)	PE0

Table 21.1 Allocation of Pin Functions to Multiple Pins (9 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
Serial communications interface	SCI12	CTS12# (input)/ RTS12# (output)/ SS12# (input)	PE3
I2C bus interface	RIIC0	SCL0-DS (input/output)	P16
		SCL0 (input/output)	P12
		SDA0-DS (input/output)	P17
		SDA0 (input/output)	P13
	RIIC1	SCL1 (input/output)	P21
		SDA1 (input/output)	P20
	RIIC3	SCL3 (input/output)	PC0
		SDA3 (input/output)	PC1
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5
			PB0
			PC5
		MOSIA (input/output)	P16
			PA6
			PC6
		MISOA (input/output)	P17
			PA7
			PC7
		SSLA0 (input/output)	PA4
			PC4
		SSLA1 (output)	PA0
			PC0
		SSLA2 (output)	PA1
			PC1
		SSLA3 (output)	PA2
			PC2
	RSPI1	RSPCKB (input/output)	P27
			PE1
			PE5
		MOSIB (input/output)	P26
			PE2
			PE6
		MISOB (input/output)	P30
			PE3
			PE7
		SSLB0 (input/output)	P31
			PE4
		SSLB1 (output)	P50
			PE0
		SSLB2 (output)	P51
			PE1
		SSLB3 (output)	P52
			PE2

Table 21.1 Allocation of Pin Functions to Multiple Pins (10 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
12-bit A/D converter		AN000 (input)*3	P40
		AN001 (input)*3	P41
		AN002 (input)*3	P42
		AN003 (input)*3	P43
		AN004 (input)*3	P44
		AN005 (input)*3	P45
		AN006 (input)*3	P46
		AN007 (input)*3	P47
		AN008 (input)*3	PE0
		AN009 (input)*3	PE1
		AN010 (input)*3	PE2
		AN011 (input)*3	PE3
		AN012 (input)*3	PE4
		AN013 (input)*3	PE5
		AN014 (input)*3	PE6
		AN015 (input)*3	PE7
		ADTRG0# (input)	P07
			P16
			P25
D/A converter		DA0 (output)*3	P03
		DA1 (output)*3	P05
Clock frequency accuracy measurement circuit		CACREF (input)	PA0
			PC7
			PH0
HDMI-CEC		CECIO (input/output)	PL5*2
Remote control signal reception	RCR0	PMC0 (input)	P00*2
	RCR1	PMC1 (input)	P01*2
External bus*4		CS0# (output)	P24
			PC7
		CS1# (output)	P25
			PC6
		CS2# (output)	P26
			PC5
		CS3# (output)	P27
			PC4
		A0 to A7 (output)	PA0 to PA7
		A8 to A15 (output)	PB0 to PB7
		A16 to A23 (output)	PC0 to PC7
		D0 to D7 (input/output)	PD0 to PD7
		D8 to D15 (input/output)	PE0 to PE7
		BCLK (output)	P53
		RD# (output)	P52
		WR# (output)	P50
		WR0# (output)	P50
		WR1# (output)	P51

Table 21.1 Allocation of Pin Functions to Multiple Pins (11 / 11)

Module/Function	Channel	Pin Functions	Allocation Port
External bus*4		BC0# (output)	PA0
		BC1# (output)	P51
		WAIT# (input)	P51
			P55
			PC5
		ALE (output)	P54

Note 1. The assigned port is available only in 5-V packages. It is not available in 3-V packages.
Note 2. The assigned port is available only in 3-V packages. It is not available in 5-V packages.
Note 3. Select general input (by setting the Bm bits for the given pin in the PDR and PMR for the given port to 0) for the pin if this pin function is to be used.
Note 4. Regarding setting for the external bus, refer to section 21.3, Settings for the External Bus Interface.

21.2 Register Descriptions

Registers and bits for pins that are not present are reserved. Write the value after a reset when writing to such bits.

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 to 3, 5, 7)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h, P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 P00: IRQ8 P01: IRQ9 P02: IRQ10 P03: IRQ11	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin 1: Used as analog pin P03: DA0 P05: DA1	R/W

The Pmn pin function control register (PmnPFS) selects the pin function. Bits PSEL[4:0] select the peripheral function assigned to each port pin.
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 switching a pin to analog using the ASEL bit, set the corresponding port mode register bit (PORTm.PMR) to “general I/O port” and the port direction register bit (PORTm.PDR) to “input”. The pin state cannot be read at this point. 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

	Pin			
PSEL[4:0] Settings	**P00**	**P01**	**P02**	**P07**
00000b (initial value)	Hi-Z			
00101b	TMRI0	TMCI0	TMCI1	—
01001b	—	—	—	ADTRG0#
01010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6	—
11100b*1	PMC0	PMC1	—	—

—: Do not specify this value.
Note 1. Not supported in 5-V packages.

21.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 2 to 7)

Address(es): P12PFS 0008 C14Ah, P13PFS 0008 C14Bh, P14PFS 0008 C14Ch, P15PFS 0008 C14Dh, P16PFS 0008 C14Eh, P17PFS 0008 C14Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 P12: IRQ2 P13: IRQ3 P14: IRQ4 P15: IRQ5 P16: IRQ6 P17: IRQ7	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

PSEL[4:0] Settings	Pin P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z					
00001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
00110b	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	—	POE8#
01001b	—	—	—	—	ADTRG0#	—
01010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
01101b	—	—	—	—	MOSIA	MISOA
01111b	SCL0	SDA0	—	—	SCL0-DS	SDA0-DS

—: Do not specify this value.

21.2.4 P2n Pin Function Control Register (P2nPFS) (n = 0 to 7)

Address(es): P20PFS 0008 C150h, P21PFS 0008 C151h, P22PFS 0008 C152h, P23PFS 0008 C153h, P24PFS 0008 C154h, P25PFS 0008 C155h, P26PFS 0008 C156h, P27PFS 0008 C157h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 P20: IRQ8 P21: IRQ9	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.4 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin							
	P20	P21	P22	P23	P24	P25	P26	P27
00000b (initial value)	Hi-Z							
00001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
00011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
00101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
00110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
01101b	—	—	—	—	—	—	MOSIB	RSPCKB
01111b	SDA1	SCL1	—	—	—	—	—	—

—: Do not specify this value.

21.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 4)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P34PFS 0008 C15Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 P30: IRQ0-DS P31: IRQ1-DS P32: IRQ2-DS P33: IRQ3-DS P34: IRQ4	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.5 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin				
	P30	P31	P32	P33	P34
00000b (initial value)	Hi-Z				
00001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
00011b	—	—	TIOCC0	TIOCD0	—
00101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
00110b	PO8	PO9	PO10	PO11	PO12
00111b	POE8#	—	—	POE3#	POE2#
01010b	RXD1 SMISO1 SSCL1	—	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
01101b	MISOB	SSLB0	—	—	—

—: Do not specify this value.

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: Not used as an analog pin 1: Used as an analog pin P40: AN000 P41: AN001 P42: AN002 P43: AN003 P44: AN004 P45: AN005 P46: AN006 P47: AN007	R/W

21.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 0 to 2, 4 to 6)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah, P54PFS 0008 C16Ch, P55PFS 0008 C16Dh, P56PFS 0008 C16Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 P55: IRQ10	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.6 Register Settings for Input/Output Pin Function in 145-Pin and 144-Pin

	Pin					
PSEL[4:0] Settings	**P50**	**P51**	**P52**	**P54**	**P55**	**P56**
00000b (initial value)	Hi-Z					
00001b	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
00011b	—	—	—	—	—	TIOCA1
00101b	—	—	—	TMCI1	TMO3	—
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—	—
01101b	SSLB1	SSLB2	SSLB3	—	—	—

—: Do not specify this value.

21.2.8 P6n Pin Function Control Registers (P6nPFS) (n = 0, 1)

Address(es): P60PFS 0008 C170h, P61PFS 0008 C171h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.7 Register Settings for Input/Output Pin Function

	Pin	
PSEL[4:0] Settings	**P60**	**P61**
00000b (initial value)		Hi-Z
01010b	SCK9	—
01011b	—	CTS9# RTS9# SS9#

—: Do not specify this value.

21.2.9 P7n Pin Function Control Registers (P7nPFS) (n = 0, 3 to 7)

Address(es): P70PFS 0008 C178h, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh, P77PFS 0008 C17Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b5	—	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 P73: IRQ12	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

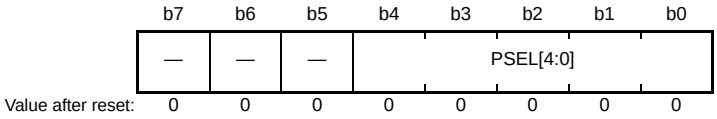
Table 21.8 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin				
	P70	P74	P75	P76	P77
00000b (initial value)	Hi-Z				
01010b	SCK4	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
01011b	—	CTS11# RTS11# SS11#	—	—	—

—: Do not specify this value.

21.2.10 P8n Pin Function Control Registers (P8nPFS) (n = 0 to 3, 6, 7)

Address(es): P80PFS 0008 C180h, P81PFS 0008 C181h, P82PFS 0008 C182h, P83PFS 0008 C183h, P86PFS 0008 C186h, P87PFS 0008 C187h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.9 Register Settings for Input/Output Pin Function

	Pin					
PSEL[4:0] Settings	**P80**	**P81**	**P82**	**P83**	**P86**	**P87**
00000b (initial value)	Hi-Z					
00001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	—	—
00011b	—	—	—	—	TIOCA0	TIOCA2
01010b	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10	—	—	—
01011b	—	—	—	CTS10# RTS10# SS10#	—	—

—: Do not specify this value.

21.2.11 P9n Pin Function Control Registers (P9nPFS) (n = 0 to 3)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.10 Register Settings for Input/Output Pin Function in 145-Pin and 144-Pin

	Pin			
PSEL[4:0] Settings	**P90**	**P91**	**P92**	**P93**
00000b (initial value)	Hi-Z			
01010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—
01011b	—	—	—	CTS7# RTS7# SS7#

—: Do not specify this value.

21.2.12 PAn Pin Function Control Registers (PAnPFS) (n = 0 to 7)

Address(es): PA0PFS 0008 C190h, PA1PFS 0008 C191h, PA2PFS 0008 C192h, PA3PFS 0008 C193h, PA4PFS 0008 C194h, PA5PFS 0008 C195h, PA6PFS 0008 C196h, PA7PFS 0008 C197h

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 PA1: IRQ11 PA3: IRQ6-DS PA4: IRQ5-DS	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.11 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
00000b (initial value)					Hi-Z			
00001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
00010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
00011b	TIOCA0	TIOCB0	—	TIOCD0	TIOCA1	TIOCB1	TIOCA2	TIOCB2
00100b	—	—	—	TCLKB	—	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3	—
00111b	CACREF	—	—	—	—	—	POE2#	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA

—: 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,

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 PB0: IRQ12 PB1: IRQ4-DS	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

PSEL[4:0] Settings	Pin PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (initial value)					Hi-Z			
00001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
00010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
00100b	—	—	TCLKC	TCLKD	—	—	—	—
00101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
00111b	—	—	—	POE3#	—	POE1#	—	—
01010b	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4	CTS4# RTS4# SS4#	SCK4	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—

—: Do not specify this value.

21.2.14 PCn Pin Function Control Registers (PCnPFS) (n = 0 to 7)

Address(es): PC0PFS 0008 C1A0h, PC1PFS 0008 C1A1h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h, PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 PC1: IRQ12	R/W
b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.13 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z							
00001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKD	TCLKA	TCLKB	—	—	—	—
00101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00111b	—	—	—	—	POE0#	—	—	CACREF
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
01111b	SCL3	SDA3	—	—	—	—	—	—

—: Do not specify this value.

21.2.15 PDn Pin Function Control Registers (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[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 PD0: IRQ0 PD1: IRQ1 PD2: IRQ2 PD3: IRQ3 PD4: IRQ4 PD5: IRQ5 PD6: IRQ6 PD7: IRQ7	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.14 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin						
	PD1	PD2	PD3	PD4	PD5	PD6	PD7
00000b (initial value)	Hi-Z						
00001b	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
00111b	—	—	POE8#	POE3#	POE2#	POE1#	POE0#

—: Do not specify this value.

21.2.16 PEn Pin Function Control Registers (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h, PE6PFS 0008 C1B6h, PE7PFS 0008 C1B7h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5	—	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 PE2: IRQ7-DS PE5: IRQ5 PE6: IRQ6 PE7: IRQ7	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PE0:AN008 PE1:AN009 PE2:AN010 PE3:AN011 PE4:AN012 PE5:AN013 PE6:AN014 PE7:AN015	R/W

Table 21.15 Register Settings for Input/Output Pin Function

PSEL[4:0] Settings	Pin							
	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
00111b	—	—	—	POE8#	—	—	—	—
01011b	—	—	—	—	—	—	CTS4# RTS4# SS4#	—
01100b	SCK12	TXD12 TXDX12 SIOX12 SMOSI12 SSDA12	RXD12 RXDX12 SMISO12 SSCL12	CTS12# RTS12# SS12#	—	—	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB	MOSIB	MISOB
01110b	—	RSPCKB	MOSIB	—	—	—	—	—

—: Do not specify this value.

21.2.17 PF5 Pin Function Control Registers (PF5PFS)

Address(es): 0008 C1BDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	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 PF5: IRQ4	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

21.2.18 PHn Pin Function Control Registers (PHnPFS) (n = 0 to 3)

Address(es): PH0PFS 0008 C1C8h, PH1PFS 0008 C1C9h, PH2PFS 0008 C1CAh, PH3PFS 0008 C1CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b5	—	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 PH1: IRQ0 PH2: IRQ1	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.16 Register Settings for Input/Output Pin Function

	Pin			
PSEL[4:0] Settings	**PH0**	**PH1**	**PH2**	**PH3**
00000b (initial value)	Hi-Z			
00101b	—	TMO0	TMRI0	TMCI0
00111b	CACREF	—	—	—

—: Do not specify this value.

21.2.19 PJn Pin Function Control Registers (PJnPFS) (n = 1, 3)

Address(es): PJ1PFS 0008 C1D1h, PJ3PFS 0008 C1D3h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

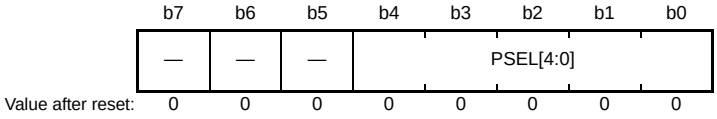
Table 21.17 Register Settings for Input/Output Pin Function

	Pin	
PSEL[4:0] Settings	**PJ1**	**PJ3**
00000b (initial value)	Hi-Z	
00001b	MTIOC3A	MTIOC3C
01010b	—	CTS0# RTS0# SS0#
01011b	—	CTS6# RTS6# SS6#

—: Do not specify this value.

21.2.20 PKn Pin Function Control Registers (PKnPFS) (n = 2 to 5)

Address(es): PK2PFS 0008 C1DAh, PK3PFS 0008 C1DBh, PK4PFS 0008 C1DCh, PK5PFS 0008 C1DDh



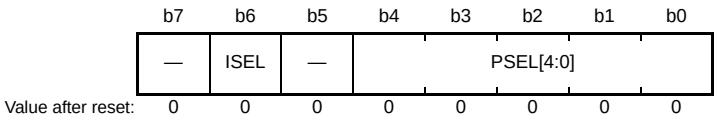
Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.18 Register Settings for Input/Output Pin Function

	Pin			
PSEL[4:0] Settings	**PK2**	**PK3**	**PK4**	**PK5**
00000b (initial value)	Hi-Z			
01010b	TXD9 SMOSI9 SSDA9	RXD9 SMISO9 SSCL9	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4

21.2.21 PLn Pin Function Control Registers (PLnPFS) (n = 5)

Address(es): PL5PFS 0008 C1E5h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b5	—	Reserved	This bit is 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 PL5: IRQ12	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.19 Register Settings for Input/Output Pin Function

	Pin
PSEL[4:0] Settings	**PL5**
00000b (initial value)	Hi-Z
11011b	CECIO

21.2.22 CS Output Enable Register (PFCSE)

Address(es): 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Enable of PC7	0: Configures the PC7 as an I/O pin. 1: Configures the PC7 as an CS0# output pin.	R/W
b1	CS1E	CS1 Enable of PC6	0: Configures the PC6 as an I/O pin. 1: Configures the PC6 as an CS1# output pin.	R/W
b2	CS2E	CS2 Enable of P26	0: Configures the P26 as an I/O pin. 1: Configures the P26 as an CS2# output pin.	R/W
b3	CS3E	CS3 Enable of P27	0: Configures the P27 as an I/O pin. 1: Configures the P27 as an CS3# output pin.	R/W
b4	CS4E	CS0 Enable of P24	0: Configures the P24 as an I/O pin. 1: Configures the P24 as an CS0# output pin.	R/W
b5	CS5E	CS1 Enable of P25	0: Configures the P25 as an I/O pin. 1: Configures the P25 as an CS1# output pin.	R/W
b6	CS6E	CS2 Enable of PC5	0: Configures the PC5 as an I/O pin. 1: Configures the PC5 as an CS2# output pin.	R/W
b7	CS7E	CS3 Enable of PC4	0: Configures the PC4 as an I/O pin. 1: Configures the PC4 as an CS3# output pin.	R/W

The PFCSE register selects enabling or disabling of the CSn# (n = 0 to 3) outputs.
If CSn signals are to be output, set the external bus enable (EXBE) bit in system control register 0 (SYSCR0) to 1 after setting the PFCSE register. See section 3.2.3, System Control Register 0 (SYSCR0), for more information on the SYSCR0.EXBE bit.
Table 21.20 lists the settings for CS2# output or WAIT# input through PC5.

Table 21.20 How to Set CS2# Output or WAIT# Input through PC5

		External Bus Control Register 1 (PFBCR1) WAITS[1:0] Bit	
External Bus Enable Bit (SYSCR0.EXBE) Enabled		**10**	**Setting other than left**
CS Output Enable Register (PFCSE) CS6E Bit	1	*1	CS2# (output)
	0	WAIT# (input)	*2

Note 1. Do not make this setting.
Note 2. This can be switched between general-port and peripheral-module functions.

21.2.23 Address Output Enable Register 0 (PFAOE0)

Address(es): 0008 C104h

	b7	b6	b5	b4	b3	b2	b1	b0
	A15E	A14E	A13E	A12E	A11E	A10E	A9E	A8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A8E	Address A8 Output Enable	0: Disables A8 output. 1: Enables A8 output.	R/W
b1	A9E	Address A9 Output Enable	0: Disables A9 output. 1: Enables A9 output.	R/W
b2	A10E	Address A10 Output Enable	0: Disables A10 output. 1: Enables A10 output.	R/W
b3	A11E	Address A11 Output Enable	0: Disables A11 output. 1: Enables A11 output.	R/W
b4	A12E	Address A12 Output Enable	0: Disables A12 output. 1: Enables A12 output.	R/W
b5	A13E	Address A13 Output Enable	0: Disables A13 output. 1: Enables A13 output.	R/W
b6	A14E	Address A14 Output Enable	0: Disables A14 output. 1: Enables A14 output.	R/W
b7	A15E	Address A15 Output Enable	0: Disables A15 output. 1: Enables A15 output.	R/W

The PFAOE0 register selects enabling or disabling of address output.

AnE Bit (Address An Output Enable) (n = 8 to 15)

Each bit enables or disables output of the corresponding address signal (An).

When An signals are to be output, set the external bus enable (EXBE) bit in system control register 0 (SYSCR0) to 1 after setting the PFAOE0 register. See section 3.2.3, System Control Register 0 (SYSCR0), for more information on the SYSCR0.EXBE bit.

21.2.24 Address Output Enable Register 1 (PFAOE1)

Address(es): 0008 C105h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: Disables A16 output. 1: Enables A16 output.	R/W
b1	A17E	Address A17 Output Enable	0: Disables A17 output. 1: Enables A17 output.	R/W
b2	A18E	Address A18 Output Enable	0: Disables A18 output. 1: Enables A18 output.	R/W
b3	A19E	Address A19 Output Enable	0: Disables A19 output. 1: Enables A19 output.	R/W
b4	A20E	Address A20 Output Enable	0: Disables A20 output. 1: Enables A20 output.	R/W
b5	A21E	Address A21 Output Enable	0: Disables A21 output. 1: Enables A21 output.	R/W
b6	A22E	Address A22 Output Enable	0: Disables A22 output. 1: Enables A22 output.	R/W
b7	A23E	Address A23 Output Enable	0: Disables A23 output. 1: Enables A23 output.	R/W

The PFAOE1 register selects enabling or disabling of address output.

AnE Bit (Address An Output Enable) (n = 16 to 23)

Each bit enables or disables output of the corresponding address signal (An).

When An signals are to be output, set the external bus enable (EXBE) bit in system control register 0 (SYSCR0) to 1 after setting the PFAOE1 register. See section 3.2.3, System Control Register 0 (SYSCR0), for more information on the SYSCR0.EXBE bit.

21.2.25 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	WR1BC1E	—	DHE	—	—	—	ADRLE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADRLE	A0 to A7 Output Enable	0: Configures PA0 to PA7 as the I/O port pins. 1: Configures PA0 to PA7 as the external address bus A0 to A7.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DHE	D8 to D15 Output Enable	0: Configures PE0 to PE7 as the I/O port pins. 1: Configures PE0 to PE7 as the external data bus D8 to D15.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: Configures P51 as the I/O port pin. 1: Configures P51 as the WR1# or BC1# pin.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The PFBCR0 register controls input/output pins for the external bus.

ADRLE Bit (A0 to A7 Output Enable)

This bit selects enabling or disabling of output of the A0 to A7 signals for the address bus.

DHE Bit (D8 to D15 Output Enable)

This bit selects enabling or disabling of output of the D8 to D15 signals for the data bus.
Ensure that the setting corresponds to the external bus width as set by the external bus width selection bits (BSIZE[1:0]) in the CSi control register (CSnCR). Operation cannot be guaranteed if the DHE bit is set to 0 while the setting is for a 16-bit external bus. Regarding the CSnCR.BSIZE[1:0] bits, see section 15.3.1, CSn Control Register (CSnCR) (n = 0 to 3).

WR1BC1E Bit (WR1#/BC1# Output Enable)

This bit selects enabling or disabling of output of the WR1#/BC1# signals.

To enable the data D8 to D15 and output of WR1#/BC1#, set the external bus enable (EXBE) bit in system control register 0 (SYSCR0) to 1 after setting the PFBCR0 register. See section 3.2.3, System Control Register 0 (SYSCR0), for more information on the SYSCR0.EXBE bit.
Table 21.21 shows how to set WR1#/BC1# output or WAIT# input through P51.

Table 21.21 How to Set WR1#/BC1# Output or WAIT# Input through P51

		External Bus Control Register1 (PFBCR1) WAITS[1:0] Bit	
External Bus Enable Bit (SYSCR0.EXBE) Enabled		11	Setting other than left
External bus control register 0 (PFBCR0) WR1BC1E Bit	1	*1	WR1#/BC1# (output)
	0	WAIT# (input)	*2

Note 1. Do not make this setting.
Note 2. This can be switched between general-port and peripheral-module functions.

21.2.26 External Bus Control Register 1 (PFBCR1)

Address(es): 0008 C107h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 0: Configures P55 as the WAIT# input pin. 0 1: Configures P55 as the WAIT# input pin. 1 0: Configures PC5 as the WAIT# input pin. 1 1: Configures P51 as the WAIT# input pin.	R/W
b2	ALEOE	ALE Output Enable	0: Configures P54 as an I/O port pin. 1: Configures P54 as the ALE output pin.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PFBCR1 register selects enabling or disabling of address output.

WAITS[1:0] Bits (WAIT Selection)

The port pin specified by the WAIT[1:0] bits becomes the WAIT# pin when the external bus is enabled. However, if the specified port pin is not to be used as the WAIT# pin, the external wait enable bit (EWENB) in the CSn mode register (CSnMOD) can be cleared (disabling external wait) to make the pin available for use as a general input port pin.

When the WAIT# signal is to be input, set the external bus enable (EXBE) bit in system control register 0 (SYSCR0) to 1 after setting the PFBCR1 register. See section 3.2.3, System Control Register 0 (SYSCR0), for more information on the SYSCR0.EXBE bit.
Table 21.22 shows how to set CS2# output and WAIT# input through PC5 and Table 21.23 shows how to set WR1#/BC1# output or WAIT# input through P51.

Table 21.22 How to Set CS2# Output or WAIT# Input through PC5

		External Bus Control Register 1 (PFBCR1) WAITS[1:0] Bit	
External Bus Enable Bit (SYSCR0.EXBE) Enabled		**10**	**Setting other than left**
CS Output Enable Register (PFCSE) CS6E Bit	1	*1	CS2# (output)
	0	WAIT# (input)	*2

Table 21.23 How to Set WR1#/BC1# Output or WAIT# Input through P51

		External Bus Control Register 1 (PFBCR1) WAITS[1:0] Bit	
External Bus Enable Bit (SYSCR0.EXBE) Enabled		**11**	**Setting other than left**
External bus control register 0 (PFBCR0) WR1BC1E Bit	1	*1	WR1#/BC1# (output)
	0	WAIT# (input)	*2

Note 1. Do not make this setting.
Note 2. This can be switched between general-port and peripheral-module functions.

21.3 Settings for the External Bus Interface

To use the external bus interface, set the MPC registers as listed in Table 21.24 and set the external bus enable bit in system control register 0 (SYSCR0.EXBE) to 1.
Table 21.24 shows how to make external bus interface settings for the various port pins.
Refer to the description of the register for details on a given external bus interface control register.

Table 21.24 Setting up the External Bus Interface (1 / 2)

Port	Module Name	Signal Name	External Bus Interface Control Register
P24	External bus (CS)	CS0#	PFCSE.CS4E = 1
P25	External bus (CS)	CS1#	PFCSE.CS5E = 1
P26	External bus (CS)	CS2#	PFCSE.CS2E = 1
P27	External bus (CS)	CS3#	PFCSE.CS3E = 1
P50	External bus	WR# WR0#	
P51	External bus	WR1# BC1#	PFBCR0.WR1BC1E = 1 PFBCR1.WAITS[1:0] = 00, 01, or 10
	External bus (WAIT)	WAIT#	PFBCR0.WR1BC1E = 0 PFBCR1.WAITS[1:0] = 11
P52	External bus	RD#	
P53	External bus	BCLK	
P54	External bus	ALE	PFBCR1.ALEOE=1
P55	External bus (WAIT)	WAIT#	PFBCR1.WAITS[1:0] = 00 or 01
PA0	External bus (address)	A0 BC0#	PFBCR0.ADRLE=1
PA1	External bus (address)	A1	PFBCR0.ADRLE=1
PA2	External bus (address)	A2	PFBCR0.ADRLE=1
PA3	External bus (address)	A3	PFBCR0.ADRLE=1
PA4	External bus (address)	A4	PFBCR0.ADRLE=1
PA5	External bus (address)	A5	PFBCR0.ADRLE=1
PA6	External bus (address)	A6	PFBCR0.ADRLE=1
PA7	External bus (address)	A7	PFBCR0.ADRLE=1
PB0	External bus (address)	A8	PFAOE0.A8E = 1
PB1	External bus (address)	A9	PFAOE0.A9E = 1
PB2	External bus (address)	A10	PFAOE0.A10E = 1
PB3	External bus (address)	A11	PFAOE0.A11E = 1
PB4	External bus (address)	A12	PFAOE0.A12E = 1
PB5	External bus (address)	A13	PFAOE0.A13E = 1
PB6	External bus (address)	A14	PFAOE0.A14E = 1
PB7	External bus (address)	A15	PFAOE0.A15E = 1
PC0	External bus (address)	A16	PFAOE1.A16E = 1
PC1	External bus (address)	A17	PFAOE1.A17E = 1
PC2	External bus (address)	A18	PFAOE1.A18E = 1
PC3	External bus (address)	A19	PFAOE1.A19E = 1
PC4	External bus (address)	A20	PFAOE1.A20E = 1 PFCSE.CS7E = 0
	External bus (CS)	CS3#	PFAOE1.A20E = 0 PFCSE.CS7E = 1

Table 21.24 Setting up the External Bus Interface (2 / 2)

Port	Module Name	Signal Name	External Bus Interface Control Register
PC5	External bus (address)	A21	PFAOE1.A21E = 1 PFCSE.CS6E = 0 PFBCR1.WAITS[1:0] = 00, 01, or 11
	External bus (CS)	CS2#	PFAOE1.A21E = 0 PFCSE.CS6E = 1 PFBCR1.WAITS[1:0] = 00, 01, or 11
	External bus (WAIT)	WAIT#	PFAOE1.A21E = 0 PFCSE.CS6E = 0 PFBCR1.WAITS[1:0] = 10
PC6	External bus (address)	A22	PFAOE1.A22E = 1 PFCSE.CS1E = 0
	External bus (CS)	CS1#	PFAOE1.A22E = 0 PFCSE.CS1E = 1
PC7	External bus (address)	A23	PFAOE1.A23E = 1 PFCSE.CS0E = 0
	External bus (CS)	CS0#	PFAOE1.A23E = 0 PFCSE.CS0E = 1
PD0	External bus (data)	D0	
PD1	External bus (data)	D1	
PD2	External bus (data)	D2	
PD3	External bus (data)	D3	
PD4	External bus (data)	D4	
PD5	External bus (data)	D5	
PD6	External bus (data)	D6	
PD7	External bus (data)	D7	
PE0	External bus (data)	D8	PFBCR0.DHE = 1
PE1	External bus (data)	D9	PFBCR0.DHE = 1
PE2	External bus (data)	D10	PFBCR0.DHE = 1
PE3	External bus (data)	D11	PFBCR0.DHE = 1
PE4	External bus (data)	D12	PFBCR0.DHE = 1
PE5	External bus (data)	D13	PFBCR0.DHE = 1
PE6	External bus (data)	D14	PFBCR0.DHE = 1
PE7	External bus (data)	D15	PFBCR0.DHE = 1

21.4 Usage Notes

21.4.1 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) to 0 to select the general I/O port function.
(2) Specify the assignments of input/output signals for peripheral functions 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 to F, H, J, K; n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[4:0] 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.

21.4.2 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 cleared to 0. If a Pmn pin function control 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) Analog input functions for the A/D converter are multiplexed with pins of ports 4 and E. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, i.e. configuring the pin as a general-purpose input, and setting the PmnPFS.ASEL bit to 1.

(5) Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 21.25.

Table 21.25 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the 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 21.18)	Set the 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.
External bus	0	x	0	0	x	Set the PMR.Bn bit to 0 to deselect peripheral functions.
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.

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 (if the IRQ is selected from the multiplexed functions).
Note 1. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.

Note: The pin state is readable when the PmnPFS.ASEL bit is 0.
Note: If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
Note: If an RIIC function is assigned to a port pin, clear the PCR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
Note: Do not make settings to assign multiple external bus signals to a single pin.

21.4.3 Note on Using Analog Functions

(1) When an analog function is in use, configure the pin as a general-purpose input by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, and then set the ASEL bit in the Pmn pin function control register (PmnPFS) to 1.

22. Multi-Function Timer Pulse Unit 2 (MTU2a)

In this section, “PCLK” is used to refer to PCLKB.

22.1 Overview

This MCU has an on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with six channels (MTU0 to MTU5).
Table 22.1 lists the specifications of the MTU, and Table 22.2 lists the functions of the MTU. Figure 22.1 shows a block diagram of the MTU.

Table 22.1 MTU Specifications

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clocks	Eight clocks or seven clocks for each channel (four clocks for MTU5)
Available operations	[MTU0 to MTU4] • Waveform output at compare match • Input capture function (noise filter set function) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match or input capture • Simultaneous register input/output by synchronous counter operation • A maximum of 12-phase PWM output is available in combination with synchronous operation
	[MTU0, MTU3, MTU4] • Buffer operation specifiable • AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset-synchronized PWM output is settable and the selection of two types of waveform outputs (chopping and level) is possible.
	[MTU1, MTU2] • Phase counting mode specifiable independently • Cascade connection operation
	[MTU3, MTU4] • A total of 6-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset PWM output, by interlocking operation
	[MTU5] • Dead time compensation counter • Input capture function (noise filter set function) • Counter clear operation
Complementary PWM mode	• Interrupts at the crest and trough of the counter value • A/D converter start triggers can be skipped
Interrupt sources	28 sources
Buffer operation	Automatic transfer of register data
Trigger generation	A/D converter start trigger can be generated Programmable pulse generator (PPG) output trigger can be generated
Low power consumption function	Module stop state can be set.

Table 22.2 MTU Functions (1/3)

Item		MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 MTCLKA MTCLKB MTCLKC MTCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 MTCLKA MTCLKB MTCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64
External clocks for phase counting mode		—	MTCLKA MTCLKB	MTCLKC MTCLKD	—	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW
General registers/ buffer registers		TGRC TGRD TGRF	—	—	TGRC TGRD	TGRC TGRD	—
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	Input pins MTIC5U MTIC5V MTIC5W
Counter clear function		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	Low output	○	○	○	○	○	—
	High output	○	○	○	○	○	—
	Toggle output	○	○	○	○	○	—
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	—
PWM mode 1		○	○	○	○	○	—
PWM mode 2		○	○	○	—	—	—
Complementary PWM mode		—	—	—	○	○	—
Reset-synchronized PWM		—	—	—	○	○	—
AC synchronous motor drive mode		○	—	—	○	○	—
Phase counting mode		—	○	○	—	—	—
Buffer operation		○	—	—	○	○	—
Dead time compensation counter function		—	—	—	—	—	○
DMAC activation		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture and TCNT overflow or underflow	—
DTC activation		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 and TCNT overflow or underflow	TGR compare match or input capture

Table 22.2 MTU Functions (2/3)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
A/D converter start trigger	TGRA compare match or input capture TGRB compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture TCNT underflow (trough) in complementary PWM mode	—
PPG trigger	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	—	—
Interrupt sources	7 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	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 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	5 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	3 sources • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W
Event link function (output)	—	4 sources • Compare match 1A • Compare match 1B • Overflow • Underflow	4 sources • Compare match 2A • Compare match 2B • Overflow • Underflow	6 sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow • Underflow	6 sources • Compare match 4A • Compare match 4B • Compare match 4C • Compare match 4D • Overflow • Underflow	—
Event link function (input)	—	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	—
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—

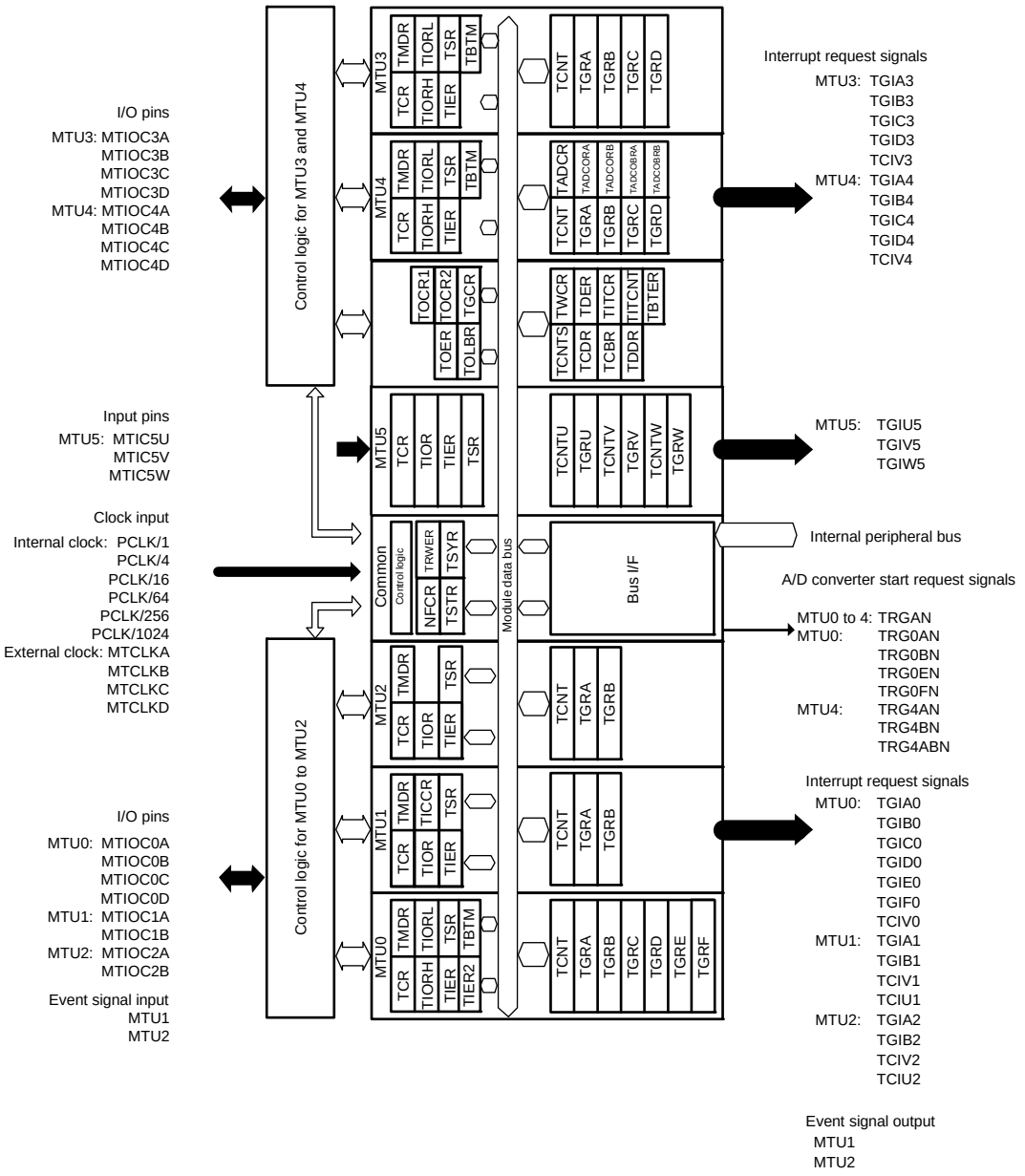
Table 22.2 MTU Functions (3/3)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Interrupt skipping function	—	—	—	• Skips TGRA compare match interrupts	• Skips TCIV interrupts	—
Module stop function	MSTPCRA.MSTPA9*1					

○: Possible

—: Not possible

Note 1. For details on the module stop function, refer to section 11, Low Power Consumption.



TSTR:	Timer start register
TSYR:	Timer synchronous register
TCR:	Timer control register
NFCR:	Noise filter control register
TMDR:	Timer mode register
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
TGCR:	Timer gate control register
TOER:	Timer output master enable register
TOCR1:	Timer output control register 1
TOCR2:	Timer output control register 2
TSR:	Timer status register
TBTM:	Timer buffer operation transfer mode register
TCNT, TCNTU, TCNTV, TCNTW:	Timer counter
TCNTS:	Timer subcounter
TRWER:	Timer read/write enable register
TOLBR:	Timer output level buffer register
TCDR:	Timer cycle data register
TCBR:	Timer cycle buffer register
TDDR:	Timer dead time data register
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
TICCR:	Timer input capture control register
TWCR:	Timer waveform control register
TADCR:	Timer A/D converter start request control register
TADCORA:	Timer A/D converter start request cycle set register A
TADCORB:	Timer A/D converter start request cycle set register B
TADCOBRA:	Timer A/D converter start request cycle set buffer register A
TADCOBRB:	Timer A/D converter start request cycle set buffer register B
TITCR:	Timer interrupt skipping set registers
TITCNT:	Timer interrupt skipping counters register
TBTER:	Timer buffer transfer set register
TDER:	Timer dead time enable register

Figure 22.1 MTU Block Diagram

Table 22.3 lists the I/O pins to be used by the MTU.

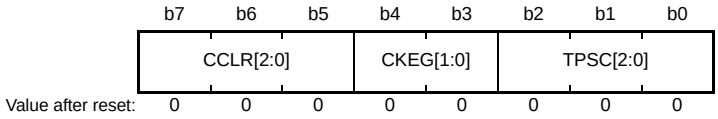
Table 22.3 MTU I/O Pins

Module Symbol	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 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)
MTU0	MTIOC0A	I/O	TGRA0 input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	TGRB0 input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	TGRC0 input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	TGRD0 input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	TGRA1 input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	TGRB1 input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	TGRA2 input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	TGRB2 input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	TGRA3 input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	TGRB3 input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	TGRC3 input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	TGRD3 input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	TGRA4 input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	TGRB4 input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	TGRC4 input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	TGRD4 input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	TGRU5 input capture input/external pulse input pin
	MTIC5V	Input	TGRV5 input capture input/external pulse input pin
	MTIC5W	Input	TGRW5 input capture input/external pulse input pin

22.2 Register Descriptions

22.2.1 Timer Control Register (TCR)

Address(es): MTU0.TCR 0008 8700h, MTU1.TCR 0008 8780h, MTU2.TCR 0008 8800h, MTU3.TCR 0008 8600h, MTU4.TCR 0008 8601h, MTU5.TCRU 0008 8884h, MTU5.TCRV 0008 8894h, MTU5.TCRW 0008 88A4h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	See Table 22.6 to Table 22.10.	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	See Table 22.4 and Table 22.5.	R/W

x: Don’t care

The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5.
The TCR register controls the TCNT operation for each channel. 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. The clock source can be selected for each channel. See Table 22.6 to Table 22.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the input clock edge. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLK/4 clock at both edges = PCLK/2 clock 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 input clock is PCLK/4 clock or slower. When PCLK/1 clock or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear)

These bits select the TCNT counter clearing source. See Table 22.4 and Table 22.5 for details.

Table 22.4 CCLR[2:0] (MTU0, MTU3, and MTU4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
MTU0, MTU3, MTU4	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 TSYR.SYNCn bit (n = 0, 3, 4) 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 CCLR1	Bit 5 CCLR0	Description
MTU1, MTU2	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

Note 1. Synchronous operation is selected by setting the TSYR.SYNCn bit (n = 1, 2) to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. This bit is read as 0. The write value should be 0.

Table 22.6 TPSC[2:0] (MTU0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU0	0	0	0	Internal clock: counts on PCLK/1 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	External clock: counts on MTCLKD pin input

Table 22.7 TPSC[2:0] (MTU1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU1	0	0	0	Internal clock: counts on PCLK/1 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256 clock
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: This setting is ignored when MTU1 is in phase counting mode.

Table 22.8 TPSC[2:0] (MTU2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU2	0	0	0	Internal clock: counts on PCLK/1 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024 clock

Note: This setting is ignored when MTU2 is in phase counting mode.

Table 22.9 TPSC[2:0] (MTU3 and MTU4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU3, MTU4	0	0	0	Internal clock: counts on PCLK/1 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	1	0	0	Internal clock: counts on PCLK/256 clock
	1	0	1	Internal clock: counts on PCLK/1024 clock
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

Table 22.10 TPSC[1:0] (MTU5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU5	0	0	Internal clock: counts on PCLK/1 clock
	0	1	Internal clock: counts on PCLK/4 clock
	1	0	Internal clock: counts on PCLK/16 clock
	1	1	Internal clock: counts on PCLK/64 clock

Note: Bits 7 to 2 are reserved in MTU5. These bits are read as 0. The write value should be 0.

22.2.2 Timer Mode Register (TMDR)

Address(es): MTU0.TMDR 0008 8701h, MTU1.TMDR 0008 8781h, MTU2.TMDR 0008 8801h, MTU3.TMDR 0008 8602h, MTU4.TMDR 0008 8603h

	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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. See 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 operate normally 1: MTU0.TGRE and MTU0.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 TMDR register specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

Table 22.11 Operating Mode Setting by MD[3:0] Bits

Bit 3	Bit 2	Bit 1	Bit 0	
MD3	MD2	MD1	MD0	Description
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*1
0	1	0	0	Phase counting mode 1*2
0	1	0	1	Phase counting mode 2*2
0	1	1	0	Phase counting mode 3*2
0	1	1	1	Phase counting mode 4*2
1	0	0	0	Reset-synchronized PWM mode*3
1	0	0	1	Setting prohibited
1	0	1	x	Setting prohibited
1	1	0	0	Setting prohibited
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*3
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*3
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*3

x: Don't care

Note 1. PWM mode 2 cannot be set for MTU3 and MTU4.

Note 2. Phase counting mode cannot be set for MTU0, MTU3, and MTU4.

Note 3. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3.
When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and conform to the MTU3 setting, respectively. 0 should be set for MTU4.
Reset-synchronized PWM mode and complementary PWM mode cannot be set for MTU0, MTU1 and MTU2.

BFA Bit (Buffer Operation A)

This bit specifies normal operation for TGRA or buffered operation of the combination of TGRA and TGRC. 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 MTU4.TIER.TGIEC bit should be cleared to 0.
When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the BFA bit in MTU4.TMDR 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. See Figure 22.40 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies normal operation for TGRB or buffered operation of the combination of TGRB and TGRD. 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 in the Tb interval in complementary PWM mode, the MTU3.TIER.TGIED or MTU4.TIER.TGIED bit should be cleared to 0.
When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the TMDR.BFB bit in MTU4 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. See Figure 22.40 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies normal operation or buffered operation for MTU0.TGRE and MTU0.TGRF. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

22.2.3 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 0008 8702h, MTU1.TIOR 0008 8782h, MTU2.TIOR 0008 8802h, MTU3.TIORH 0008 8604h, MTU4.TIORH 0008 8606h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	See the following tables.*1 MTU0.TIORH: Table 22.20 MTU1.TIOR: Table 22.22 MTU2.TIOR: Table 22.23 MTU3.TIORH: Table 22.24 MTU4.TIORH: Table 22.26	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables.*1 MTU0.TIORH: Table 22.12 MTU1.TIOR: Table 22.14 MTU2.TIOR: Table 22.15 MTU3.TIORH: Table 22.16 MTU4.TIORH: Table 22.18	R/W

Note 1. If the IOn[3:0] (n = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high-impedance.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 0008 8703h, MTU3.TIORL 0008 8605h, MTU4.TIORL 0008 8607h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	See the following tables.*1 MTU0.TIORL: Table 22.21 MTU3.TIORL: Table 22.25 MTU4.TIORL: Table 22.27	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables.*1 MTU0.TIORL: Table 22.13 MTU3.TIORL: Table 22.17 MTU4.TIORL: Table 22.19	R/W

Note 1. If the IOn[3:0] (n = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high-impedance.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 0008 8886h, MTU5.TIORV 0008 8896h, MTU5.TIORW 0008 88A6h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	See the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 22.28	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has a total of 11 TIOR registers, two each for MTU0, MTU3, and MTU4, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5.
TIOR should be set when TMDR is set to select normal mode, PWM mode, or phase counting mode.
The initial output specified by TIOR is valid when the counter is stopped (the TSTR.CST bit is set to 0). Note also that, in PWM mode 2, the output at the point at which the counter is set to 0 is specified.
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 22.12 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU0.TGRB 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.13 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU0.TGRD 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the MTU0.TMDR.BFB is set to 1 and MTU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.14 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU1.TGRB Function**	**MTIOC1B Pin Function**
0	0	0	0	MTU1.TGRB works as an 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.TGRC compare match/input capture.

x: Don't care

Table 22.15 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU2.TGRB Function**	**MTIOC2B Pin Function**
0	0	0	0	MTU2.TGRB works as an 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 22.16 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU3.TGRB 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 22.17 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU3.TGRD 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.TMDR.BFB bit is set to 1 and MTU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.18 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	**IOB2**	**IOB1**	**IOB0**	**MTU4.TGRB Function**	**MTIOC4B Pin Function**
0	0	0	0	MTU4.TGRB works as an 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 22.19 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	**IOD2**	**IOD1**	**IOD0**	**MTU4.TGRD 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.TMDR.BFB bit is set to 1 and MTU4.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.20 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU0.TGRA 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	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.21 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU0.TGRC 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the MTU0.TMDR.BFA bit is set to 1 and MTU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.22 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU1.TGRA 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 22.23 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU2.TGRA 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 22.24 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU3.TGRA 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 22.25 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU3.TGRC 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.TMDR.BFA bit is set to 1 and MTU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.26 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	**IOA2**	**IOA1**	**IOA0**	**MTU4.TGRA 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 22.27 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	**IOC2**	**IOC1**	**IOC0**	**MTU4.TGRC 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.TMDR.BFA bit is set to 1 and MTU4.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.28 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC4	IOC3	IOC2	IOC1	IOC0	MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Compare match register	Compare match
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 trough in complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at trough in 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 trough in complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at trough in 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 MTU3 and MTU4. For details, refer to section 22.3.10, External Pulse Width Measurement and section 22.3.11, Dead Time Compensation.

22.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0008 88B6h

	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

The TCNTCMPCLR register specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

22.2.5 Timer Interrupt Enable Register (TIER)

- TIER (MTU0 to MTU4)

Address(es): MTU0.TIER 0008 8704h, MTU1.TIER 0008 8784h, MTU2.TIER 0008 8804h, MTU3.TIER 0008 8608h, MTU4.TIER 0008 8609h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	TCIEU	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 Converter Start Request Enable 2	0: A/D converter start request generation by MTU4.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTU4.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Converter Start Request Enable	0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

The MTU has a total of seven TIER registers, two each for MTU0 and one each for MTU1 to MTU5.
The TIER register enables or disables interrupt requests in each channel.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIn) in MTU0, MTU3, and MTU4 (n = 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 MTU1 and MTU2.
In MTU0, MTU3, and MTU4, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTU4.TCNT underflow (trough) in complementary PWM mode.
In MTU0 to MTU3, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.

- TIER2 (MTU0)

Address(es): MTU0.TIER2 0008 8724h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	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.TGRn (n = E, F).

- TIER (MTU5)

Address(es): MTU5.TIER 0008 88B2h

	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 TGI5W disabled 1: Interrupt requests TGI5W enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGI5V disabled 1: Interrupt requests TGI5V enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGI5U disabled 1: Interrupt requests TGI5U enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5W, TGIE5V, and TGIE5U Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGI5n) (n = W, V, U).

22.2.6 Timer Status Register (TSR)

- TSR (MTU0 to MTU4)

Address(es): MTU0.TSR 0008 8705h, MTU1.TSR 0008 8785h, MTU2.TSR 0008 8805h, MTU3.TSR 0008 862Ch, MTU4.TSR 0008 862Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 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

The MTU has a total of five TSR registers, one each for MTU0 to MTU4.
The TSR register indicates the status of each channel.

TCFD Flag (Count Direction Flag)

Status flag that shows the direction in which TCNT counts in MTU1 to MTU4.
In MTU0, this bit is reserved. It is read as 1. The write value should be 1.

22.2.7 Timer Buffer Operation Transfer Mode Register (TBTM)

Address(es): MTU0.TBTM 0008 8726h, MTU3.TBTM 0008 8638h, MTU4.TBTM 0008 8639h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	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, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has a total of three TBTM registers, one each for MTU0, MTU3, and MTU4.
The TBTM register specifies the timing for transferring data from the buffer register to the timer general register in PWM mode.

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 when they are used together for buffer operation. In MTU3 and MTU4, this bit is read as 0. The write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

22.2.8 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0008 8790h

	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 TMTIOC1B 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

The MTU has one TICCR for MTU1.

The TICCR register specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.

22.2.9 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 0008 8640h

	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: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable *1, *2, *3	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable *1, *2, *3	0: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable *1, *2, *3	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter 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	See Table 22.29 for details.	R/W

Note: TADCR must not be accessed in 8-bit units; it should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the TITCR.T3AEN and T4VEN bits are cleared to 0 or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the TADCR.ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Set b6 and b4 to b0 to 0 when complementary PWM mode is not selected.

The TADCR register enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

Table 22.29 Setting of Transfer Timing by TADCR.BF[1:0] Bits

Bit 15	Bit 14	Description			
BF1	BF0	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

22.2.10 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB)

Address(es): MTU4.TADCORA 0008 8644h, MTU4.TADCORB 0008 8646h

	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: MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers specify the A/D converter start request cycle. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

22.2.11 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB)

Address(es): MTU4.TADCOBRA 0008 8648h, MTU4.TADCOBRB 0008 864Ah

	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: MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers function as buffer registers for TADCORA and TADCORB, respectively. These registers specify the A/D converter start request cycle. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.

22.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 0008 8706h, MTU1.TCNT 0008 8786h, MTU2.TCNT 0008 8806h, MTU3.TCNT 0008 8610h,MTU4.TCNT 0008 8612h, MTU5.TCNTU 0008 8880h, MTU5.TCNTV 0008 8890h, MTU5.TCNTW 0008 88A0h

	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: The TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. TCNT is a readable/writable counter.

22.2.13 Timer General Register (TGR)

Address(es): MTU0.TGRA 0008 8708h, MTU0.TGRB 0008 870Ah, MTU0.TGRC 0008 870Ch, MTU0.TGRD 0008 870Eh, MTU0.TGRE 0008 8720h, MTU0.TGRF 0008 8722h, MTU1.TGRA 0008 8788h, MTU1.TGRB 0008 878Ah, MTU2.TGRA 0008 8808h, MTU2.TGRB 0008 880Ah, MTU3.TGRA 0008 8618h, MTU3.TGRB 0008 861Ah, MTU3.TGRC 0008 8624h, MTU3.TGRD 0008 8626h, MTU4.TGRA 0008 861Ch, MTU4.TGRB 0008 861Eh, MTU4.TGRC 0008 8628h, MTU4.TGRD 0008 862Ah, MTU5.TGRU 0008 8882h, MTU5.TGRV 0008 8892h, MTU5.TGRW 0008 88A2h

	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 TGR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units. TGR registers are initialized to FFFFh.

The MTU has a total of 21 TGR registers, six for MTU0, two each for MTU1 and MTU2, four each for MTU3 and MTU4, and three for MTU5.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for MTU0, MTU3, and MTU4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR 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.

22.2.14 Timer Start Registers (TSTR)

- TSTR (MTU0 to MTU4)

Address(es): MTU.TSTR 0008 8680h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	—	—	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 performs count stop 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT performs count stop 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT performs count stop 1: MTU2.TCNT performs count operation	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT performs count stop 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT performs count stop 1: MTU4.TCNT performs count operation	R/W

The TSTR registers start or stop TCNT operation in MTU0 to MTU4.
Before setting the operating mode in TMDR or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0 to 4)

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 but 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.

- TSTR (MTU5)

Address(es): MTU5.TSTR 0008 88B4h

	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 count operation is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV count operation is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU count operation 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

22.2.15 Timer Synchronous Registers (TSYR)

Address(es): MTU.TSYR 0008 8681h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	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
b5 to b3	—	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

The TSYR registers select independent operation or synchronous operation of TCNT in MTU0 to MTU4.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNCn Bits (Timer Synchronous n Operation) (n = 0 to 4)

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 SYNCn bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNCn bit, the TCNT clearing source must also be set the TCR.CCLR[2:0] bits.

22.2.16 Timer Read/Write Enable Registers (TRWER)

Address(es): MTU.TRWER 0008 8684h

	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

The TRWER registers enable or disable access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

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 the RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUn.TCR, MTUn.TMDR, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUn.TCNT (n = 3, 4)

22.2.17 Timer Output Master Enable Registers (TOER)

Address(es): MTU.TOER 0008 860Ah

	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 a non-active level from each pin when MTU output is disabled, make necessary settings for non-active level output from general I/O ports in the data direction registers (PDR), port output data registers (PODR), and port mode register (PMR) in advance. For details, refer to the I/O Ports section.

The TOER registers enable or disable output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the TOER bits have not been set. In MTU3 and MTU4, set TOER prior to setting TIOR.

Set TOER after clearing the TSTR.CST3 and CST4 bits to 0 (see Figure 22.35 and Figure 22.38).

22.2.18 Timer Output Control Registers 1 (TOCR1)

Address(es): MTU.TOCR 0008 860Eh

	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*2,*3	See Table 22.30.	R/W
b1	OLSN	Output Level Select N*2,*3	See Table 22.31.	R/W
b2	TOCS	TOC Select	0: TOCR1 setting is selected 1: TOCR2 setting is selected	R/W
b3	TOCL	TOC Register Write Protection*1	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*4
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 TOCR1.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 2. Clearing the TOCR1.TOCS bit to 0 makes this bit setting valid.
Note 3. If dead-time is not generated, the negative-phase output is always 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 power-on reset. After 1 is written, 0 cannot be written to the bit.

The TOCR1 registers 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.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode.

TOCS Bit (TOC Select)

This bit selects either the TOCR1 or TOCR2 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 TOCR1.

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Table 22.30 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 22.31 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 22.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

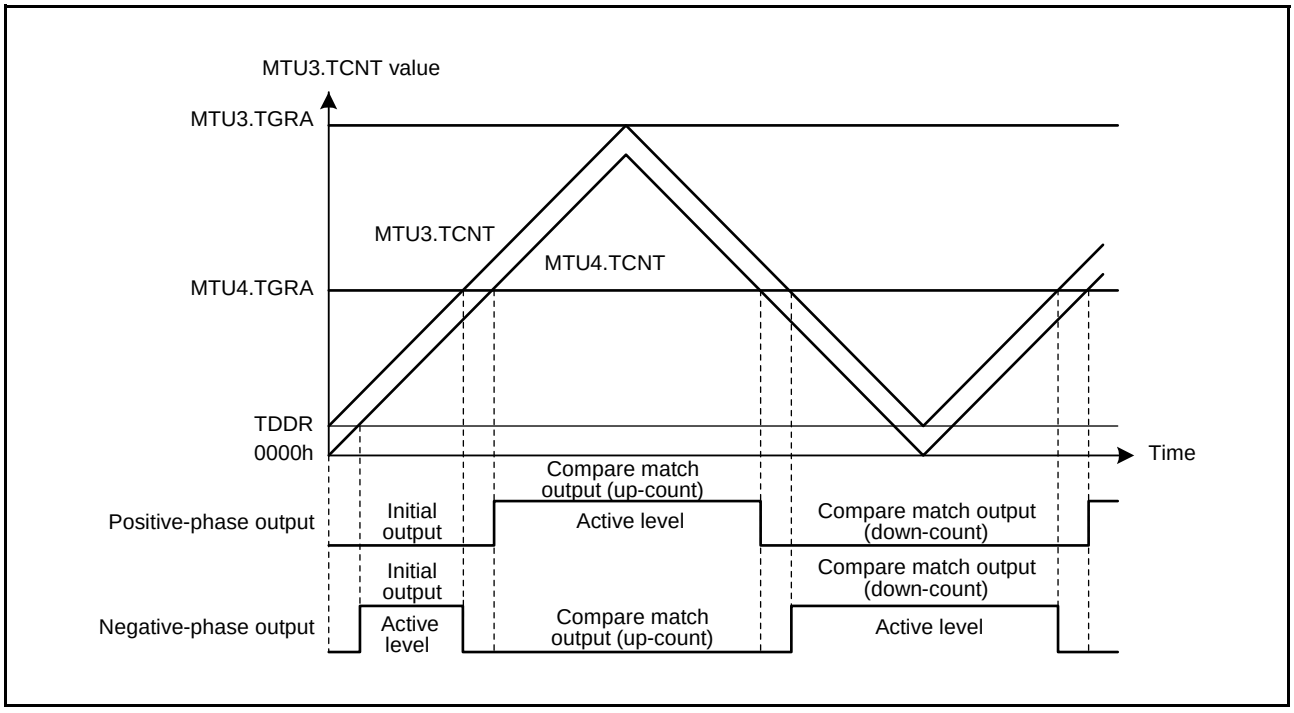


Figure 22.2 Example of Output in Complementary PWM Mode

22.2.19 Timer Output Control Registers 2 (TOCR2)

Address(es): MTU.TOCR2 0008 860Fh

	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 in reset-synchronized PWM mode and complementary PWM mode. See Table 22.32.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. See Table 22.33.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. See Table 22.34.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. See Table 22.35.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. See Table 22.36.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. See Table 22.37.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBR to TOCR2. See Table 22.38 for details.	R/W

Note 1. Setting the TOCR1.TOCS bit to 1 makes this bit setting valid.
Note 2. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In these cases, only the OLSiP bits are valid (i = 1 to 3).

The TOCR2 registers control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Table 22.32 MTIOC3B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 22.33 MTIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.34 MTIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 22.35 MTIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.36 MTIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

Table 22.37 MTIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.38 Setting of TOCR2.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from TOLBR to TOCR2.	Does not transfer data from TOLBR to TOCR2.
0	1	Transfers data from TOLBR to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from TOLBR to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from TOLBR to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from TOLBR to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

22.2.20 Timer Output Level Buffer Registers (TOLBR)

Address(es): MTU.TOLBR 0008 8636h

	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 TOCR2.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TOLBR registers function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 22.3 shows an example of the PWM output level setting procedure in buffer operation.

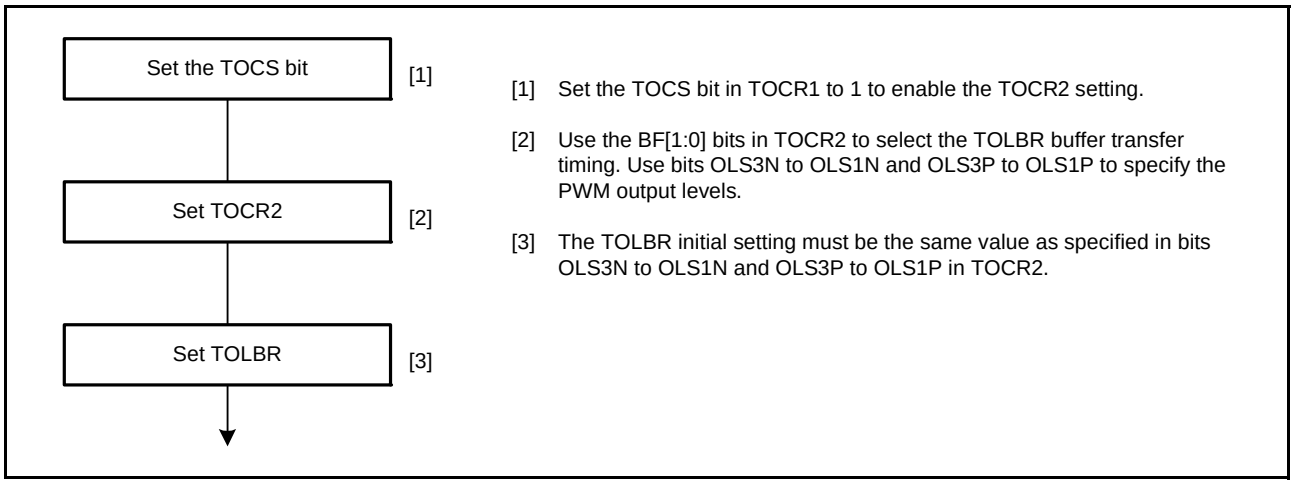


Figure 22.3 Example of PWM Output Level Setting Procedure in Buffer Operation

22.2.21 Timer Gate Control Registers (TGCR)

Address(es): MTU.TGCR 0008 860Dh

	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 TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.	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) 1: Output is switched by software (TGCR'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

The TGCR registers control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These 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 TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.

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 or by writing 0 or 1 to bits 2 to 0 in TGCR.

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, and MTIOC4B 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, and MTIOC4D pins).

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCR effective or ineffective.

Table 22.39 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
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

22.2.22 Timer Subcounters (TCNTS)

Address(es): MTU.TCNTS 0008 8620h

	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: The TCNTS counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCNTS counters are read-only counters that are used only in complementary PWM mode.

22.2.23 Timer Dead Time Data Registers (TDDR)

Address(es): MTU.TDDR 0008 8616h

	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 TDDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TDDR registers specify the MTU3.TCNT and MTU4.TCNT counter offset value in complementary PWM mode. In complementary PWM mode, when the MTU3.TCNT and MTU4.TCNT counters are cleared and then restarted, the TDDR value is loaded into the MTU3.TCNT counter and the count operation starts.

22.2.24 Timer Cycle Data Registers (TCDR)

Address(es): MTU.TCDR 0008 8614h

	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 TCDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCDR registers specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. Set half the PWM cycle as the TCDR value. TCDR is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count).

22.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

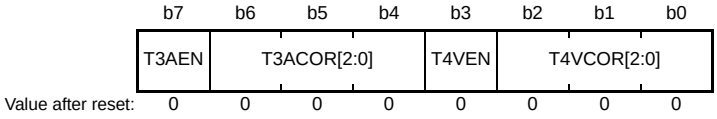
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The TCBR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCBR registers function as buffer registers for TCDR, and specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. The TCBR value is transferred to TCDR with the transfer timing set in TMDR.

22.2.26 Timer Interrupt Skipping Set Registers (TITCR)

Address(es): MTU.TITCR 0008 8630h



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, see Table 22.40.	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, see Table 22.41.	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 clear the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the TITCNT counter.

Table 22.40 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR2	T4VCOR1	T4VCOR0	Description
0	0	0	Does not perform TCIV4 interrupt skipping.
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 22.41 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR2	T3ACOR1	T3ACOR0	Description
0	0	0	Does not perform TGIA3 interrupt skipping.
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.

22.2.27 Timer Interrupt Skipping Counters (TITCNT)

Address(es): MTU.TITCNT 0008 8631h

	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 TITCR is set to 1, the count in these bits is incremented every time a TCIV4 interrupt source occurs.	R
b3	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA3 interrupt source occurs.	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note: To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

The TITCNT counters count the number of interrupt source occurrences for interrupt skipping. TITCNT retain their values even after stopping the count operation of MTU4.TCNT and MTU3.TCNT.

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the T4VCNT[2:0] bits in TITCNT match the T4VCOR[2:0] bits in TITCR
- When the T4VEN bit in TITCR is cleared to 0
- When the T4VCOR[2:0] bits in TITCR are cleared to 000b

T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the T3ACNT[2:0] bits in TITCNT match the T3ACOR[2:0] bits in TITCR
- When the T3AEN bit in TITCR is cleared to 0
- When the T3ACOR[2:0] bits in TITCR are cleared to 000b

22.2.28 Timer Buffer Transfer Set Registers (TBTER)

Address(es): MTU.TBTER 0008 8632h

	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 used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. See Table 22.42 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TBTER registers 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 operation.

Table 22.42 Setting of TBTER.BTE[1:0] Bits

Bit 1	Bit 0	
BTE1	BTE0	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation.
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 operation.*2
1	1	Setting prohibited

Note: Target buffer registers: MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and MTU.TCBRA

Note 1. Data is transferred in accordance with the TMDT.MD[3:0] bit setting. For details, refer to section 22.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the TITCR.T3AEN and T4VEN bits or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), be sure to disable link of buffer transfer with interrupt skipping (clear the TBTER.BTE1 bit to 0).
If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

22.2.29 Timer Dead Time Enable Registers (TDER)

Address(es): MTU.TDER 0008 8634h

	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. TDDR must be set to 1 or a larger value.

The TDER registers specify dead time generation in complementary PWM mode. The MTU3 has one TDER register. TDER 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 the TDER bit after reading the TDER bit = 1

22.2.30 Timer Waveform Control Registers (TWCR)

Address(es): MTU.TWCR 0008 8660h

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

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Initial Output Inhibition Enable	0: Initial value specified in TOCR is output 1: Initial output is inhibited	R/(W) *1
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable	0: Counters are not cleared at MTU3.TGRA compare match 1: Counters are cleared at MTU3.TGRA compare match	R/(W) *2

Note 1. Do not set this bit to 1 unless complementary PWM mode is selected.
Note 2. Do not set this bit to 1 unless complementary PWM mode 1 is selected.

The TWCR registers control the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match.
The TWCR.CCE bit and TWCR.WRE bit should be modified only while TCNT stops.

WRE Bit (Initial Output Inhibition Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is prohibited 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 value specified in TOCR is output regardless of the WRE bit setting. The initial value specified in TOCR is also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.
For the Tb interval at the trough in complementary PWM mode, see Figure 22.40.
[Setting condition]

- When 1 is written to the WRE bit after reading the WRE bit = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode 1.
[Setting condition]

- When 1 is written to the CCE bit after reading the CCE bit = 0

22.2.31 Noise Filter Control Registers (NFCR)

- NFCR (MTU0 to MTU4)

Address(es): MTU0.NFCR 0008 8690h, MTU1.NFCR 0008 8691h, MTU2.NFCR 0008 8692h, MTU3.NFCR 0008 8693h, MTU4.NFCR 0008 8694h

	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	0: The noise filter for the MTIOCnC pin is disabled 1: The noise filter for the MTIOCnC pin is enabled	R/W*1
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTIOCnD pin is disabled 1: The noise filter for the MTIOCnD pin is enabled	R/W*1
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: The clock source for counting is the external clock	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 the NFCR for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

The MTUn.NFCR registers (n = 0 to 4) enable and disable the noise filters for the MTIOCnm (n = 0 to 4; m = A to D) pins and sets the sampling clocks for the noise filters.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

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, selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input-capture function.

- NFCR (MTU5)

Address(es): MTU5.NFCR 0008 8695h

	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: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: The clock source for counting is the external clock.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MTU5.NFCR is 8-bit readable and writable register. This register controls enabling and disabling of the noise filters for the MTIC5m (m = U, V, W) pins and sets the sampling clock for the noise filters.

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 NFUEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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 NFVEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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 NFWEN is changed, select the compare-match function in the timer I/O control register before changing the value.

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.

22.2.32 Bus Master Interface

The timer counters (TCNT), timer general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA/TADCORB), and timer A/D converter start request cycle set buffer registers (TADCORA/TADCORB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Access the registers in 16-bit units.
All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

22.3 Operation

22.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR 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 22.4 shows an example of the count operation setting procedure.

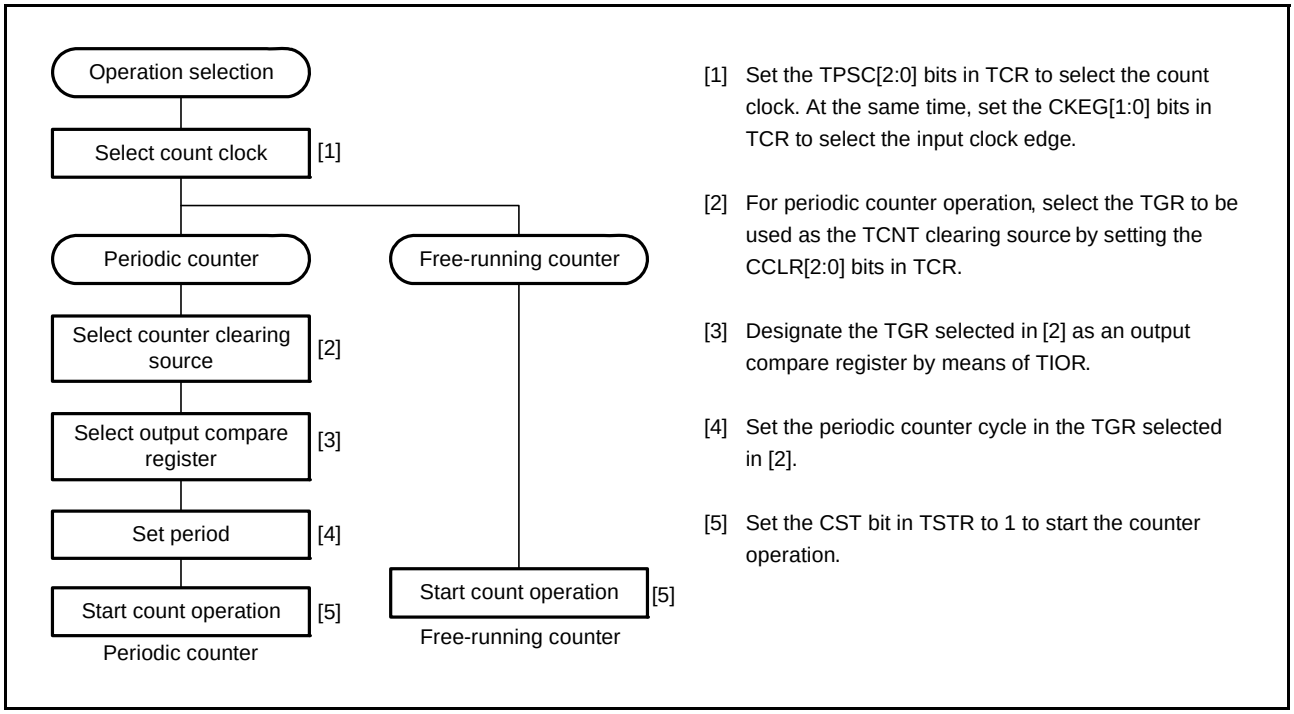


Figure 22.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant CSTn bit in 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), the MTU requests an interrupt if the corresponding TCIEV bit in TIER is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 22.5 illustrates free-running counter operation.

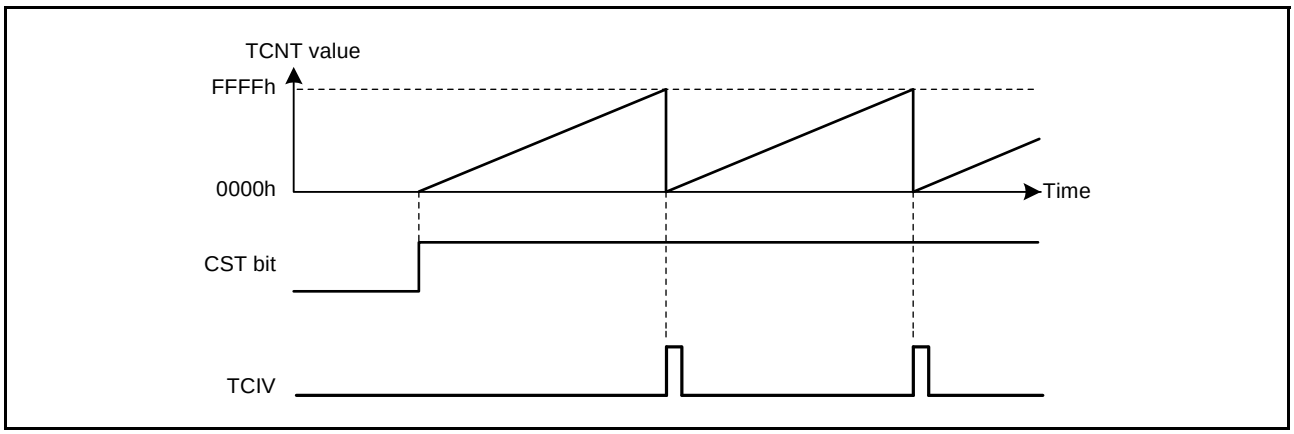


Figure 22.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of the TCR.CCLR[2:0] bits. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count matches the value in TGR, TCNT is set to 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.
Figure 22.6 illustrates periodic counter operation.

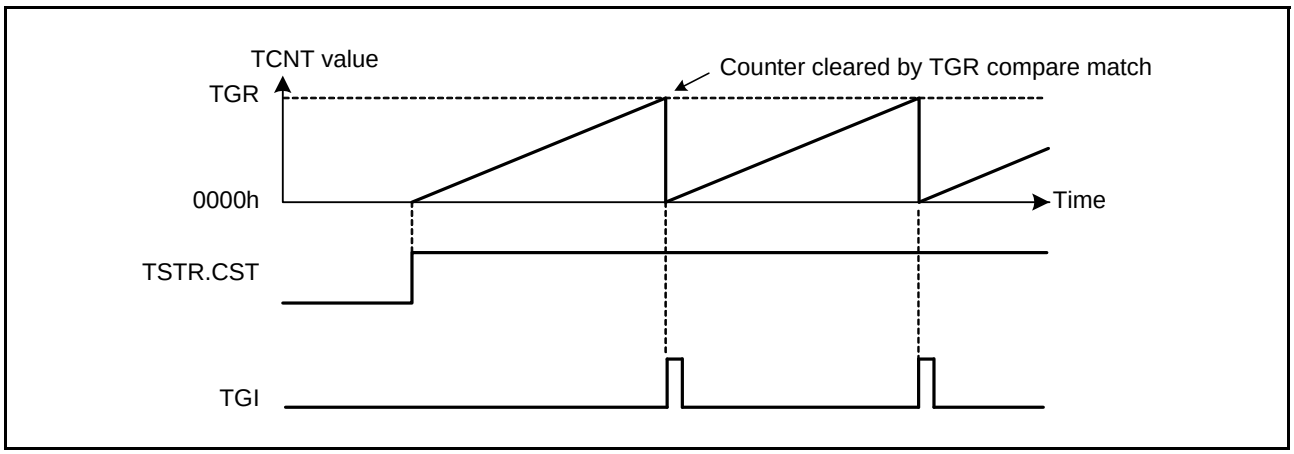


Figure 22.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU can output low or high or toggle output from the corresponding output pin using compare match.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 22.7 shows an example of the procedure for setting waveform output by compare match.

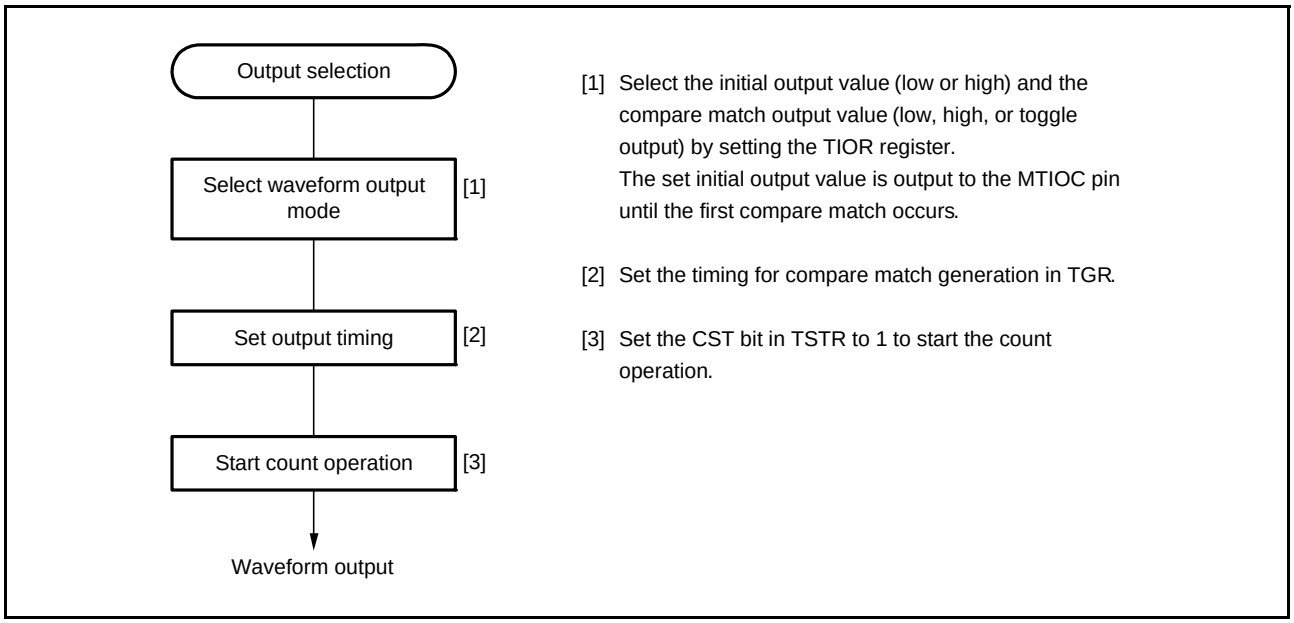


Figure 22.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 22.8 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.

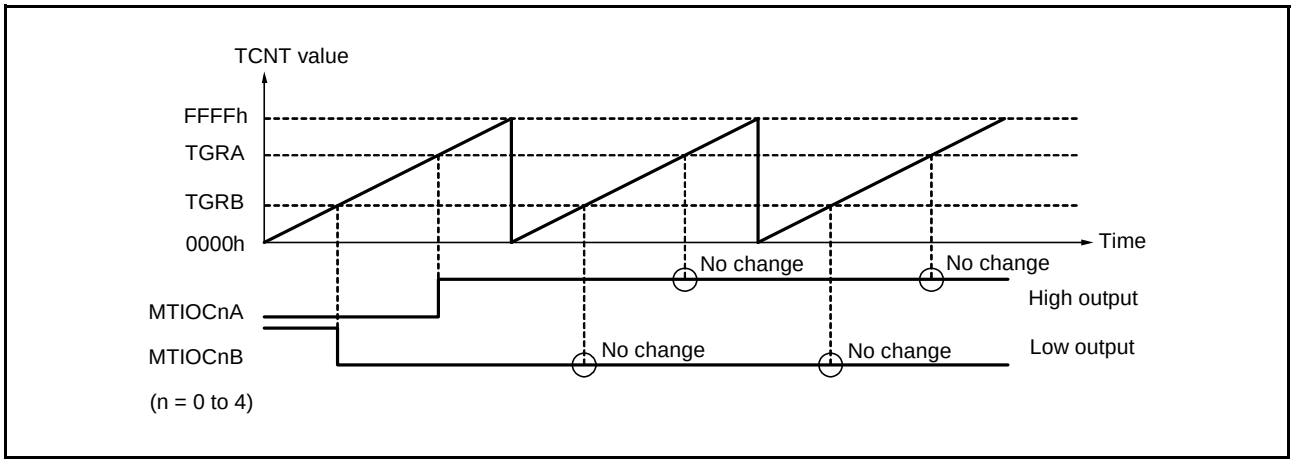


Figure 22.8 Example of Low Output and High Output Operation

Figure 22.9 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.

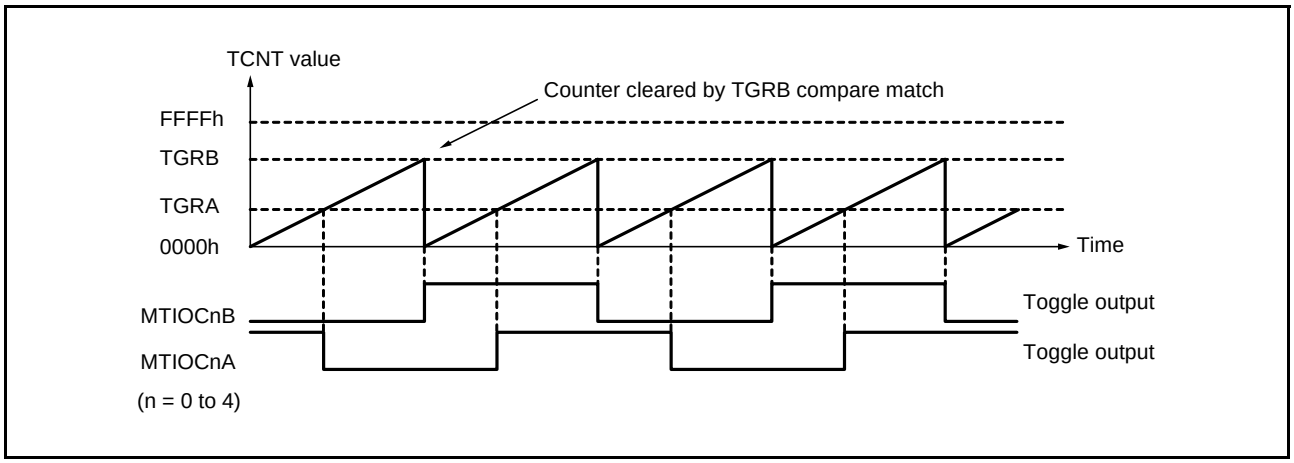


Figure 22.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the input edge of the MTIOCnm (n = 0 to 4; m = A to D) pin and MTIC5m (m = W, V, U) pin.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's counter input clock or compare match signal can also be specified as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for MTU0 and MTU1, PCLK/1 clock should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 clock is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 22.10 shows an example of the input capture operation setting procedure.

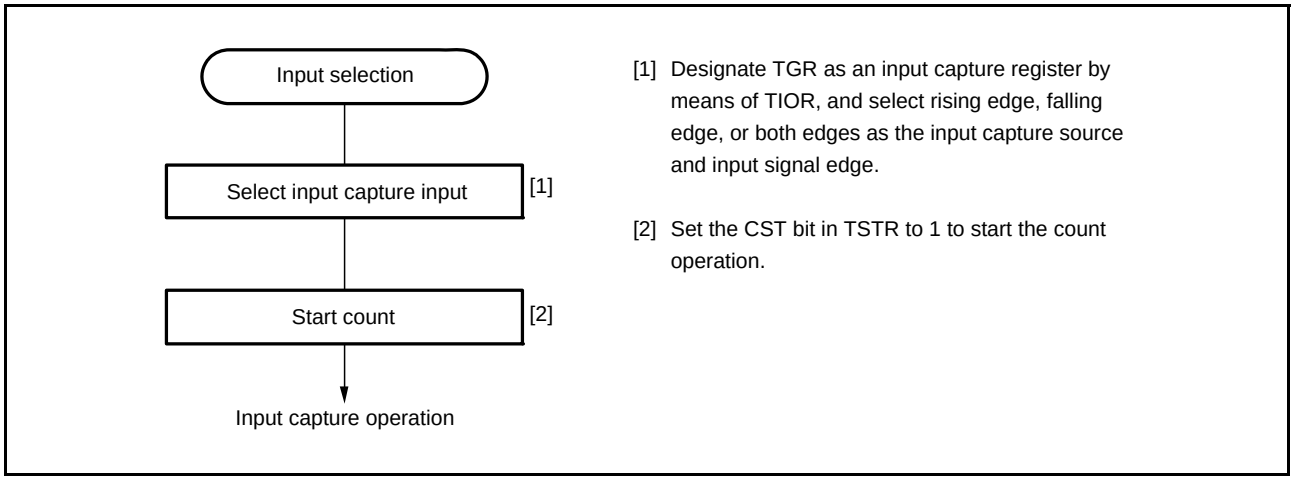


Figure 22.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 22.11 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.

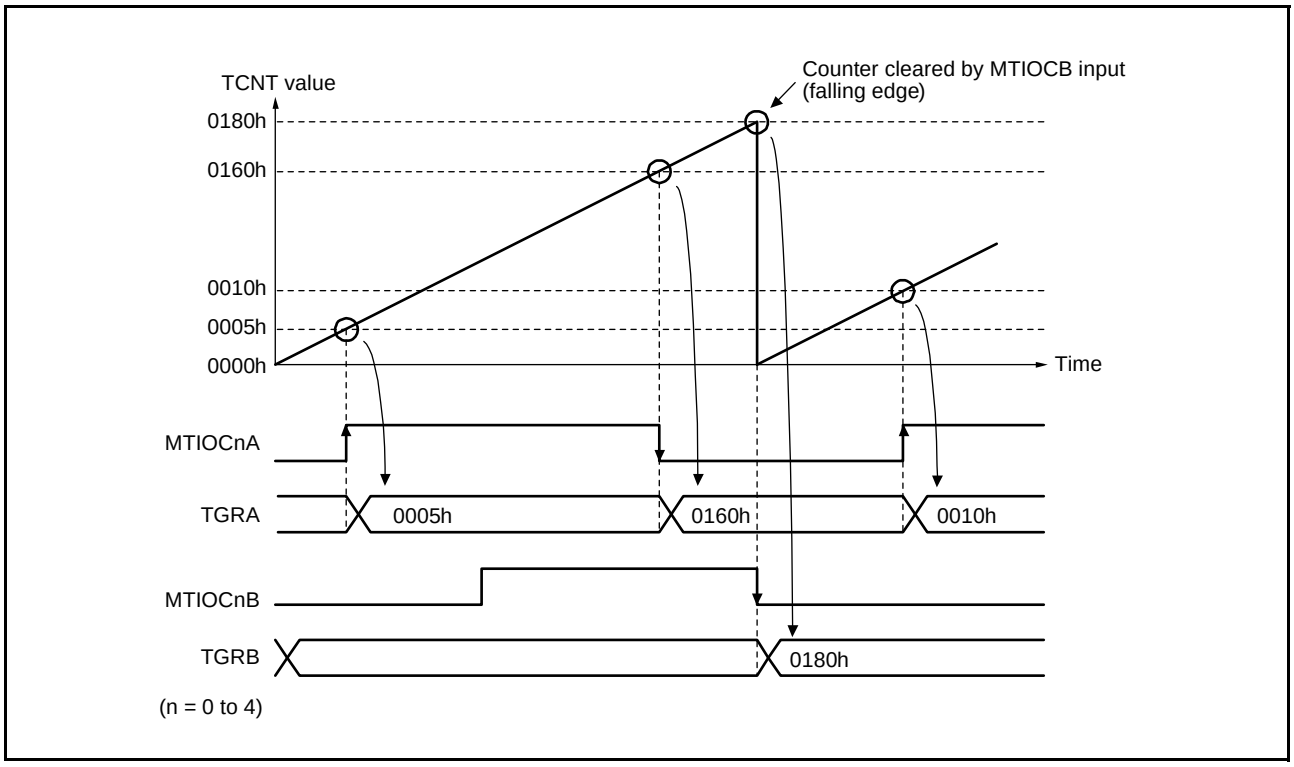


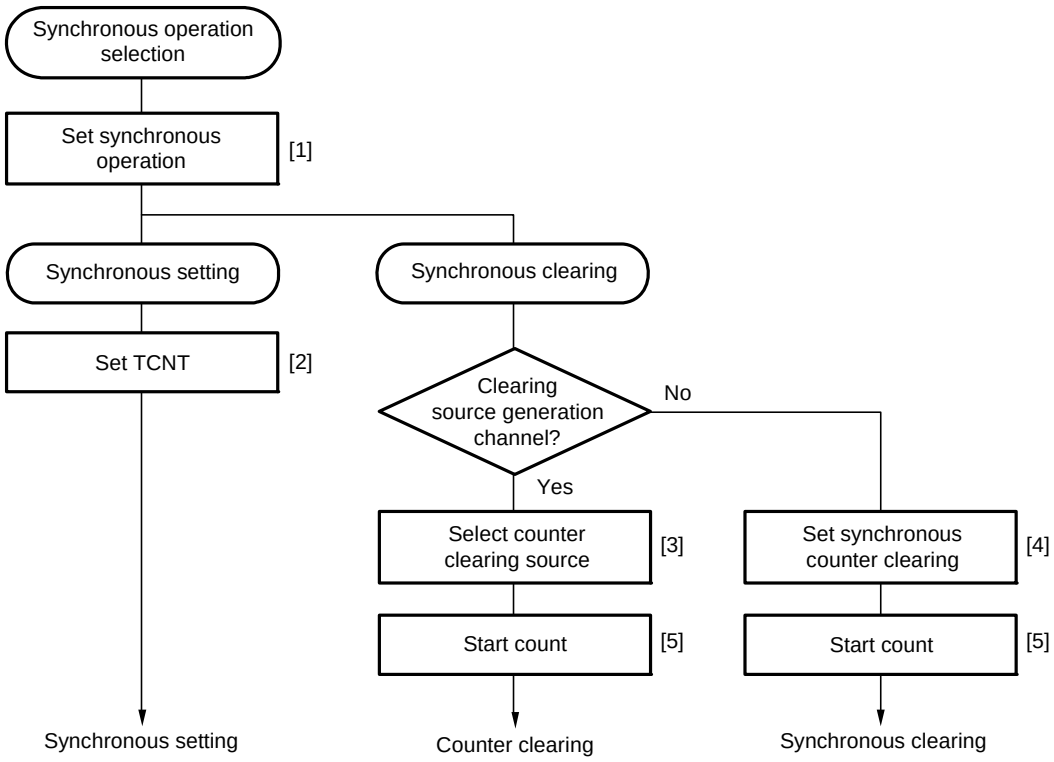
Figure 22.11 Example of Input Capture Operation

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 increases the number of TGR registers assigned to a single time base.
MTU0 to MTU4 can all be designated for synchronous operation.
MTU5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 22.12 shows an example of the synchronous operation setting procedure.



[1] Set 1 in the SYNCn bit (n = 0 to 4) in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.

[5] Set 1 in the CST bit in TSTR for the relevant channels, to start the count operation.

Figure 22.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 22.13 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, 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 cycle.
For details of PWM modes, refer to section 22.3.5, PWM Modes.

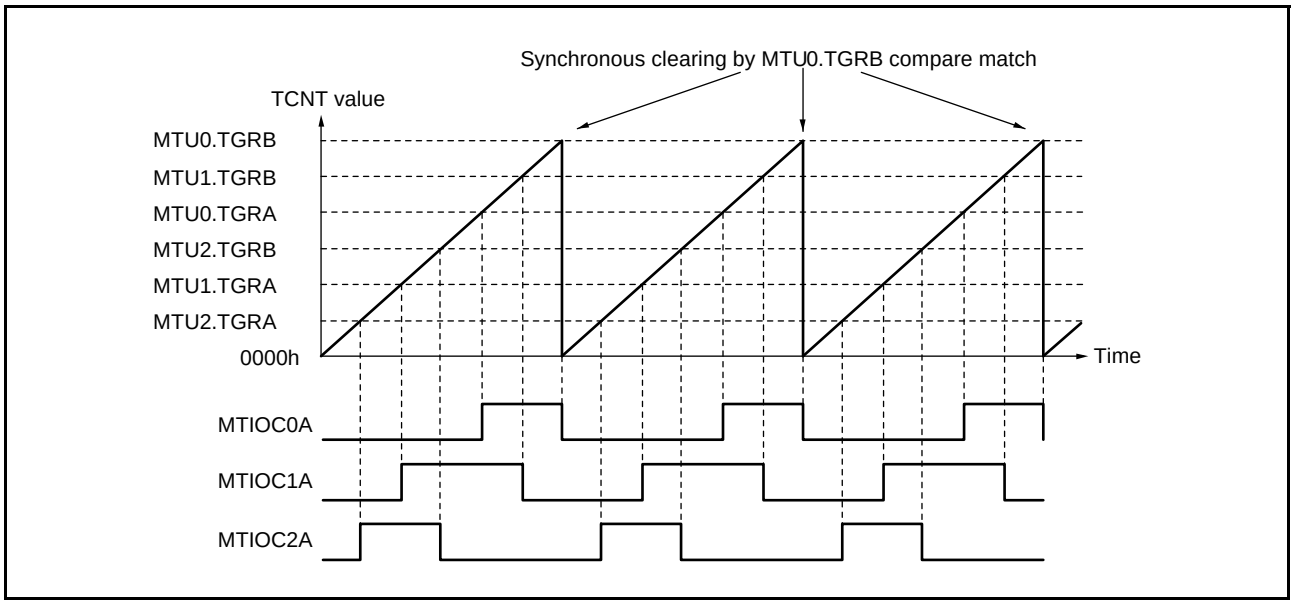


Figure 22.13 Example of Synchronous Operation

22.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, and MTU4, enables TGRC and TGRD to be used as buffer registers. In MTU0, 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 cannot be designated as an input capture register and can only operate as a compare match register.

Table 22.43 shows the register combinations used in buffer operation.

Table 22.43 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

- 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 22.14.

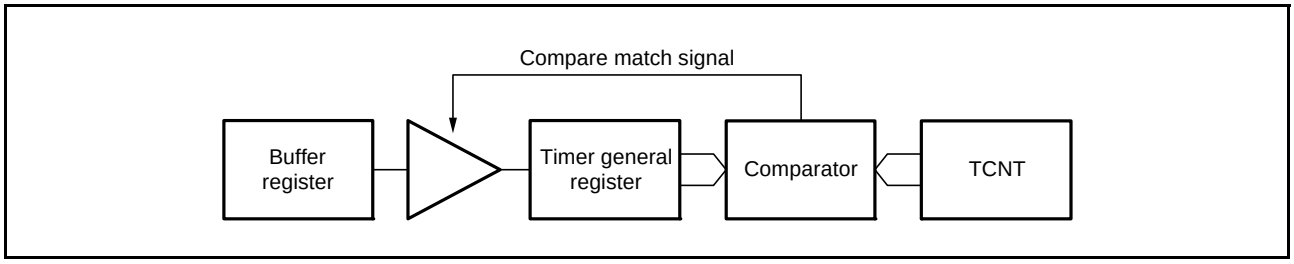


Figure 22.14 Compare Match Buffer Operation

- 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 22.15.

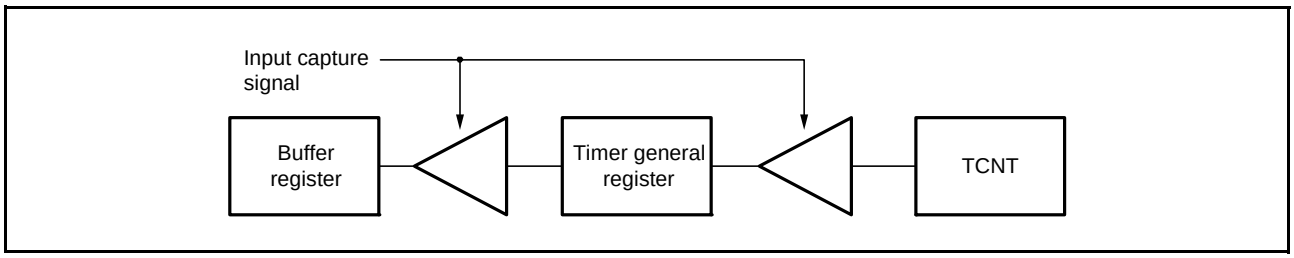


Figure 22.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 22.16 shows an example of the buffer operation setting procedure.

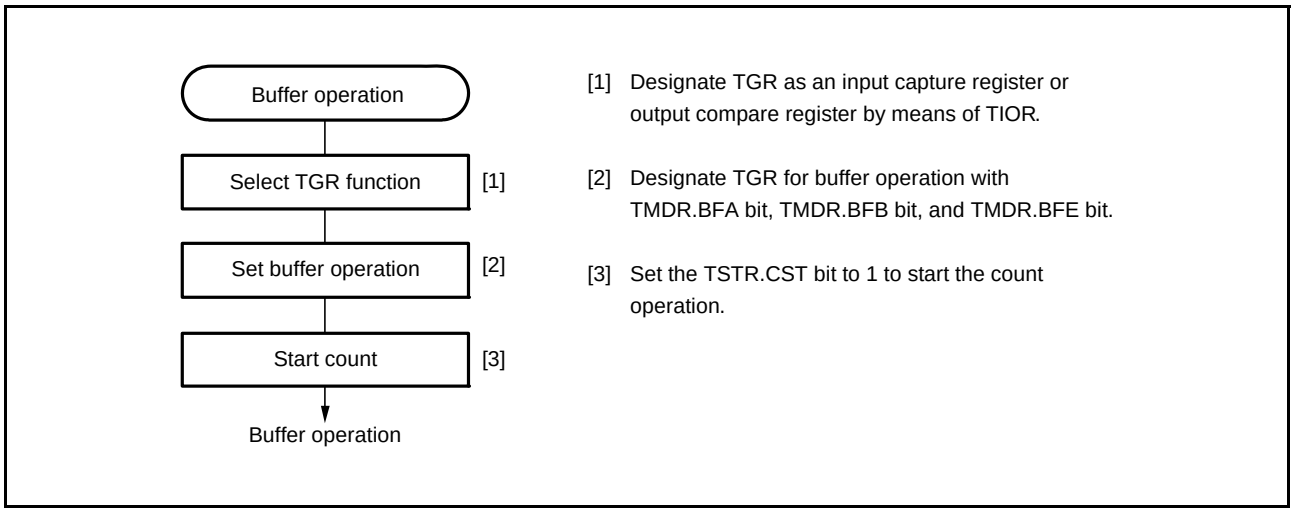


Figure 22.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 22.17 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 TBTM.TTSA bit 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.

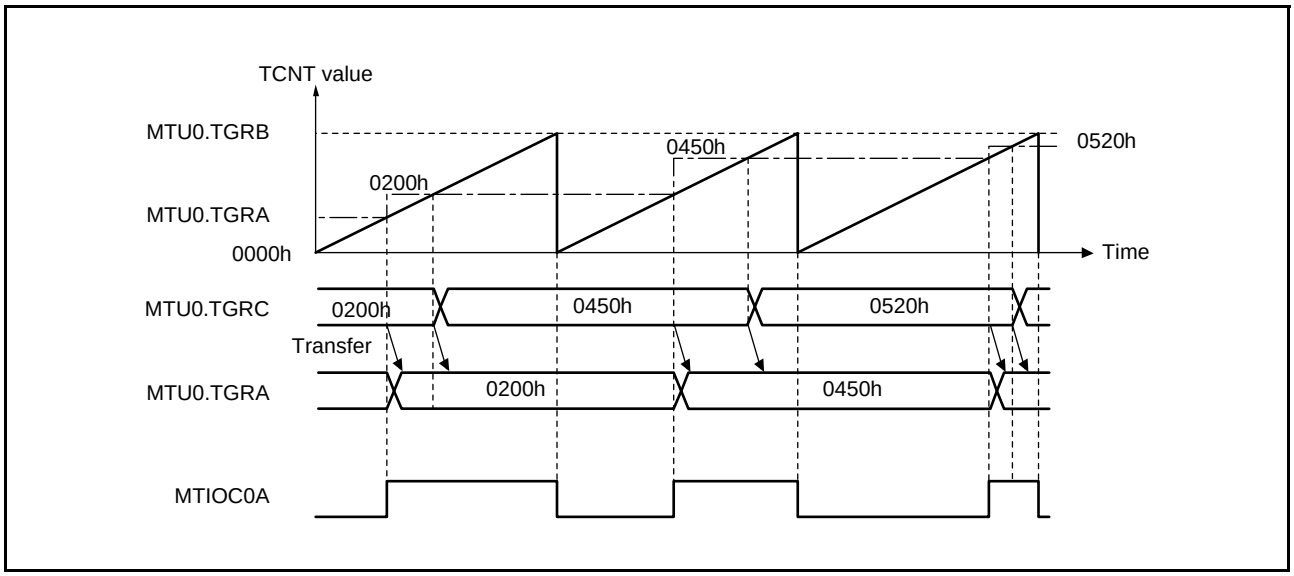


Figure 22.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 22.18 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.
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.

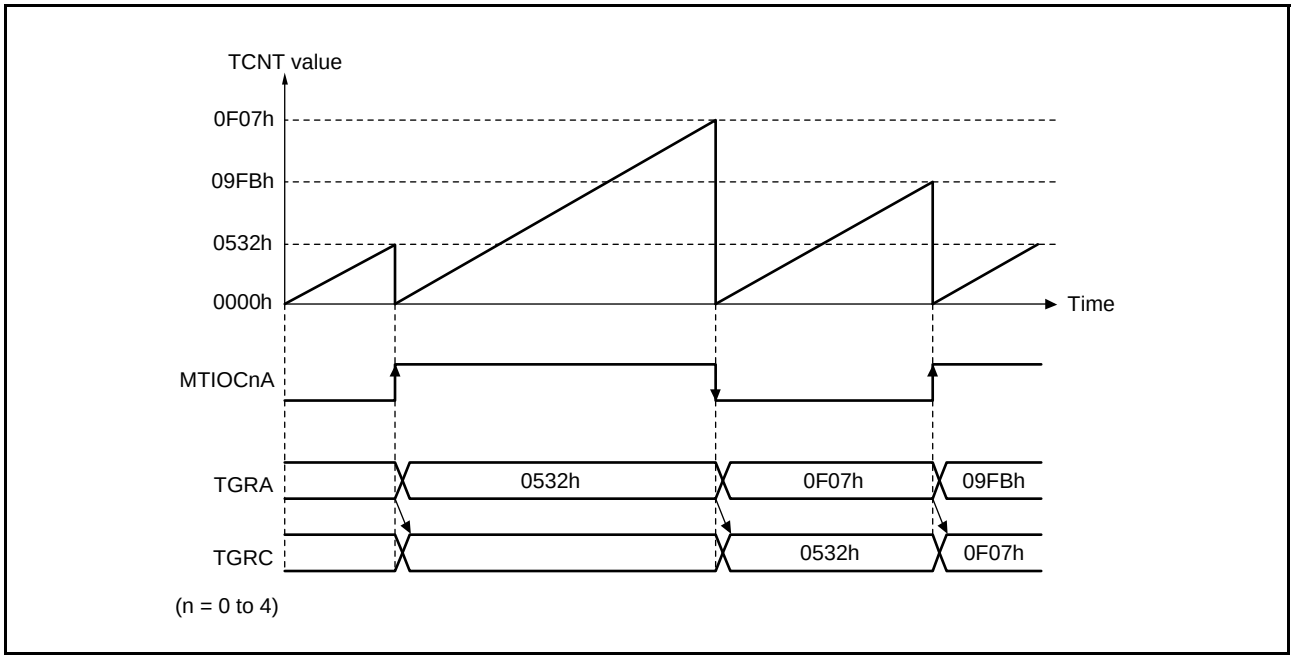


Figure 22.18 Example of Buffer Operation (2)

(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 or in PWM mode 1 for MTU3 and MTU4 by setting the timer buffer operation transfer mode registers (MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM). Either compare match (initial setting) 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 → 0000h)
- When 0000h is written to TCNT during counting
- When TCNT is set to 0000h under the condition specified in the TCR.CCLR[2:0] bits

Note: TBTM must be modified only while TCNT stops.

Figure 22.19 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 MTU0.TBTM.TTSA bit is set to 1.

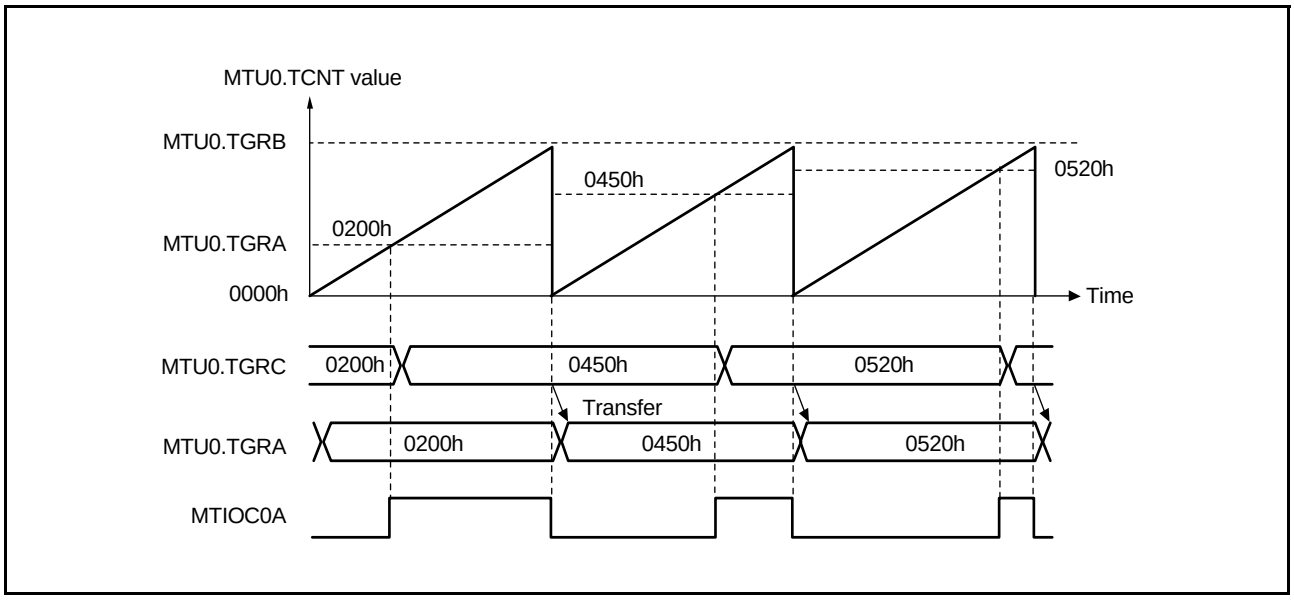


Figure 22.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing

22.3.4 Cascaded Operation

In cascaded operation, 16-bit counters in different two channels are used together as a 32-bit counter.
This function works when overflow/underflow of MTU2.TCNT is selected as the count clock for MTU1 through the TCR.TPSC[2:0] bits.
Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.
Table 22.44 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1 or MTU2, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 22.44 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 TICCR register. 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, a change in the level of the other will not produce an edge for detection. For details, see (4) Cascaded Operation Example (c). For input capture in cascade connection, refer to section 22.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 22.45 lists the TICCR setting and input capture input pins.

Table 22.45 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 22.20 shows an example of the cascaded operation setting procedure.

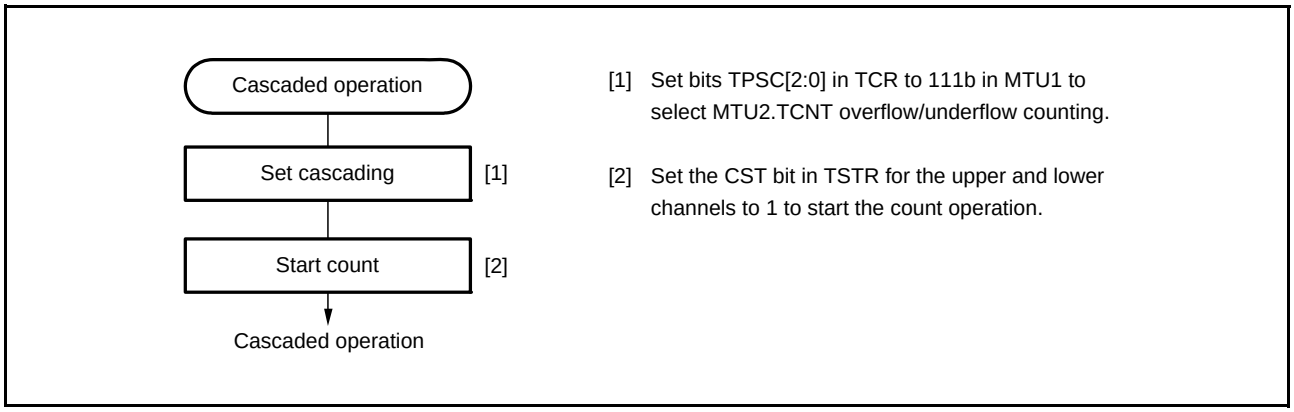


Figure 22.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 22.21 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.

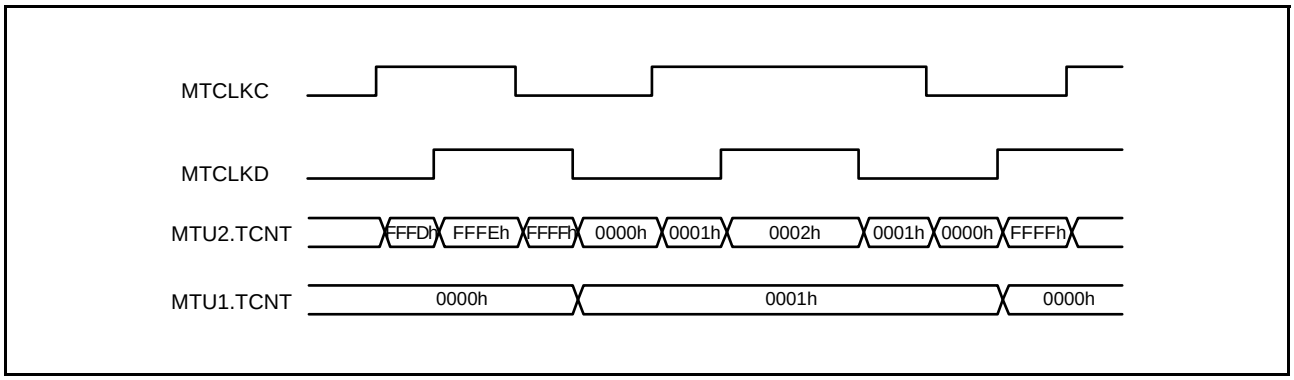


Figure 22.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 22.22 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 the MTIOC1A rising edge 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, 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.

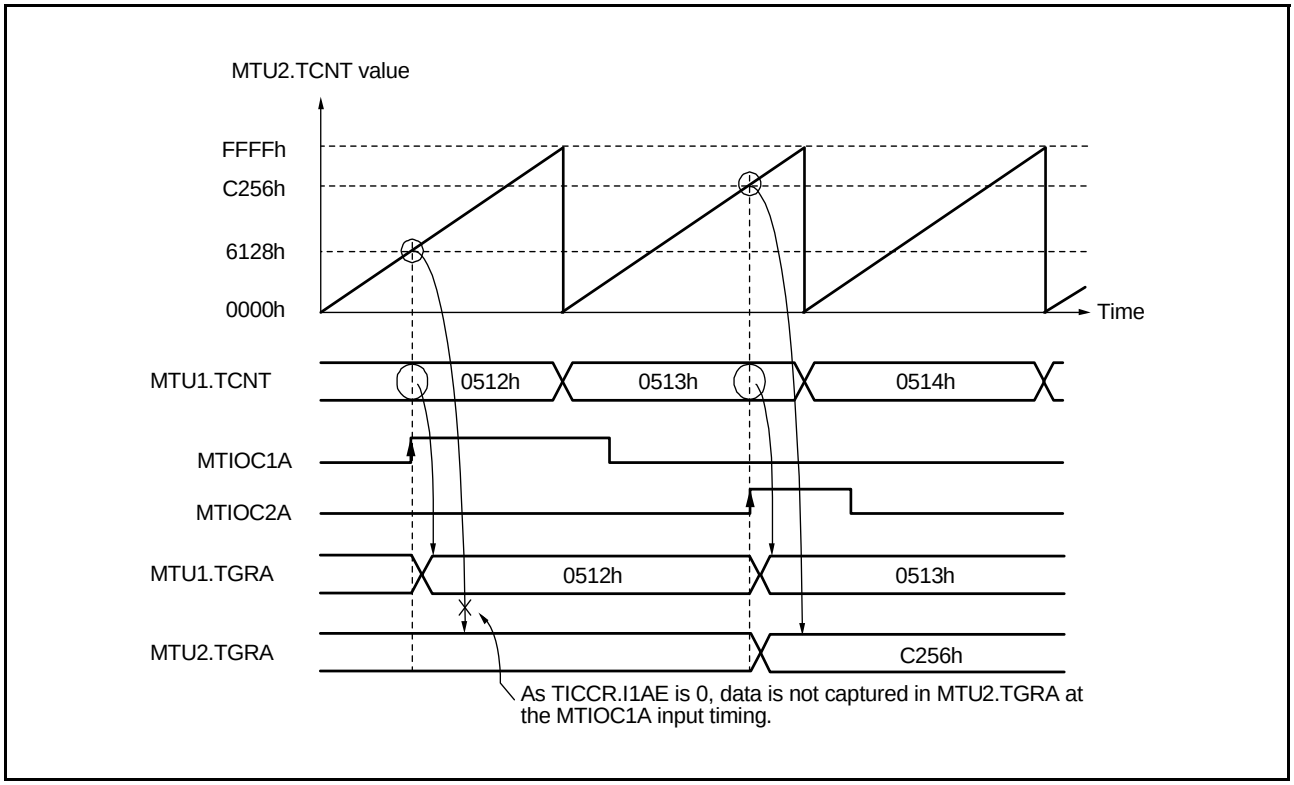
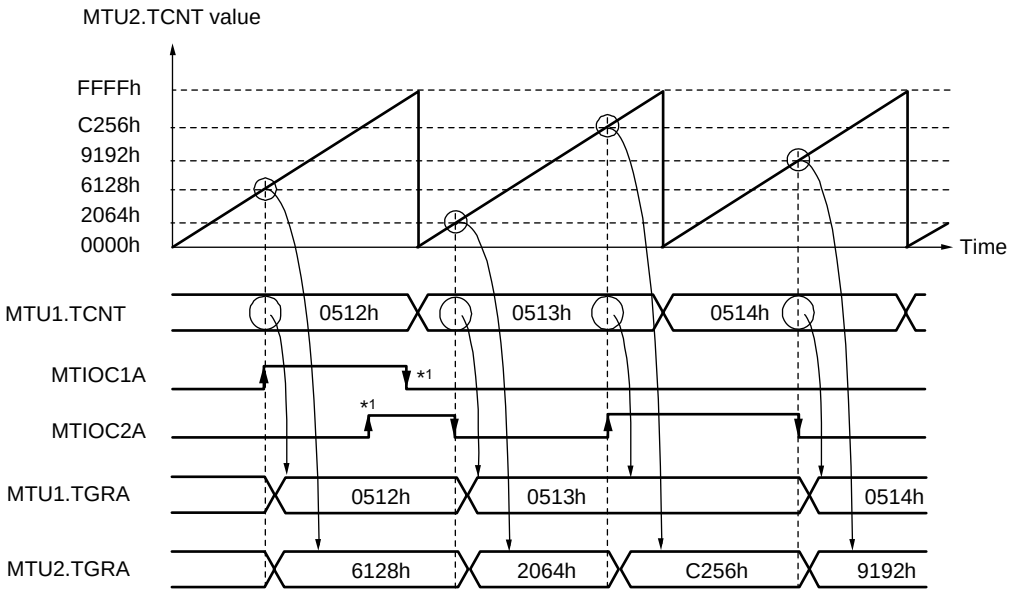


Figure 22.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 22.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCR 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 OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either input signal is at the high level, an edge in the other input signal does not act as the condition for capture.

Figure 22.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 22.24 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 TICCR.I2AE bit has been set to 1.

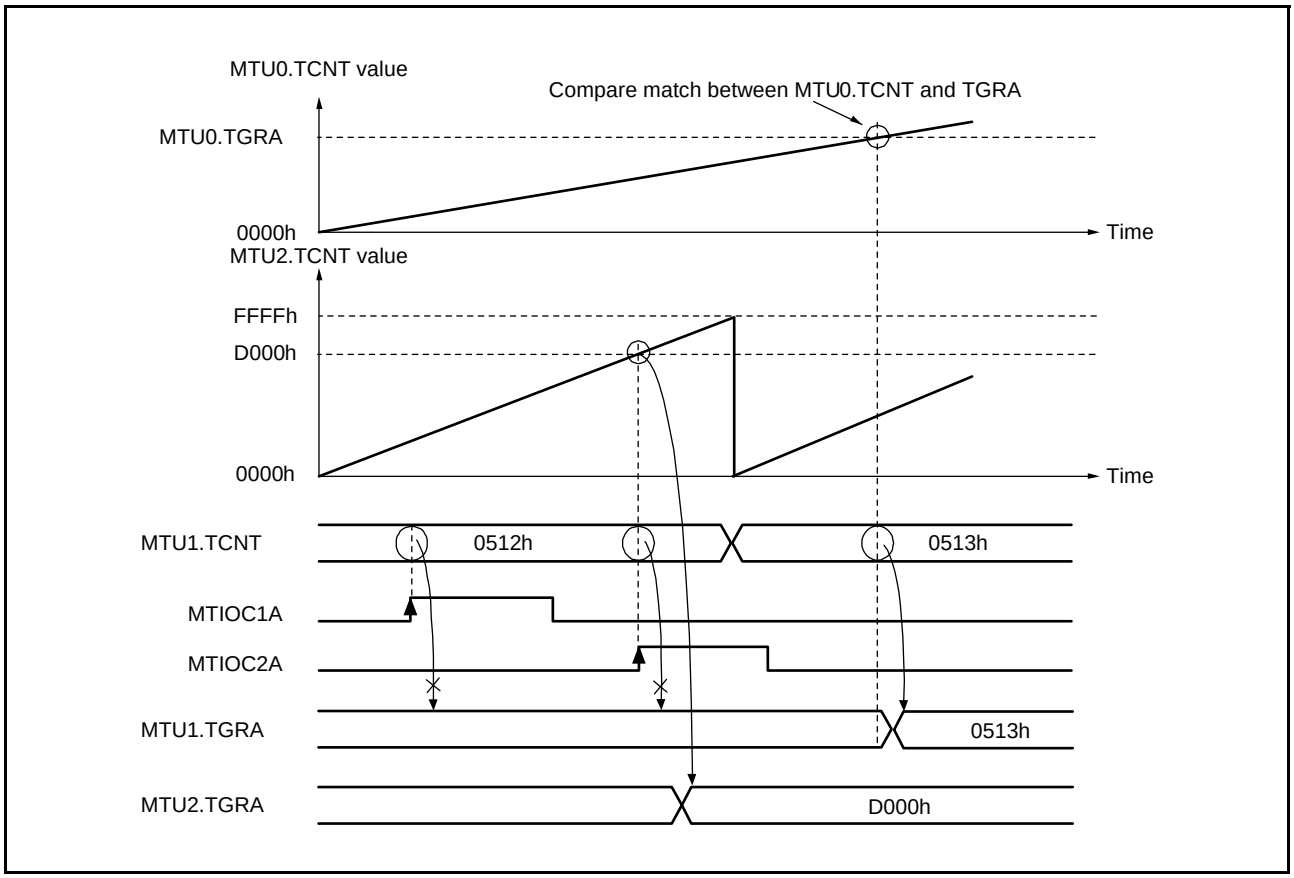


Figure 22.24 Cascaded Operation Example (d)

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 cycle can be specified in that register.
Every channel 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 levels specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D. 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, up to eight phases of PWM waveforms can be output.

(b) PWM Mode 2

PWM output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.
In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.
The correspondence between PWM output pins and registers is listed in Table 22.46.

Table 22.46 PWM Output Registers and Output Pins

Channel	Register	Output Pins	
		PWM Mode 1	PWM Mode 2
MTU0	MTU0.TGRA	MTIOC0A	MTIOC0A
	MTU0.TGRB		MTIOC0B
	MTU0.TGRC	MTIOC0C	MTIOC0C
	MTU0.TGRD		MTIOC0D
MTU1	MTU1.TGRA	MTIOC1A	MTIOC1A
	MTU1.TGRB		MTIOC1B
MTU2	MTU2.TGRA	MTIOC2A	MTIOC2A
	MTU2.TGRB		MTIOC2B
MTU3	MTU3.TGRA	MTIOC3A	Setting prohibited
	MTU3.TGRB		
	MTU3.TGRC	MTIOC3C	
	MTU3.TGRD		
MTU4	MTU4.TGRA	MTIOC4A	
	MTU4.TGRB		
	MTU4.TGRC	MTIOC4C	
	MTU4.TGRD		

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the PWM cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 22.25 shows an example of the PWM mode setting procedure.

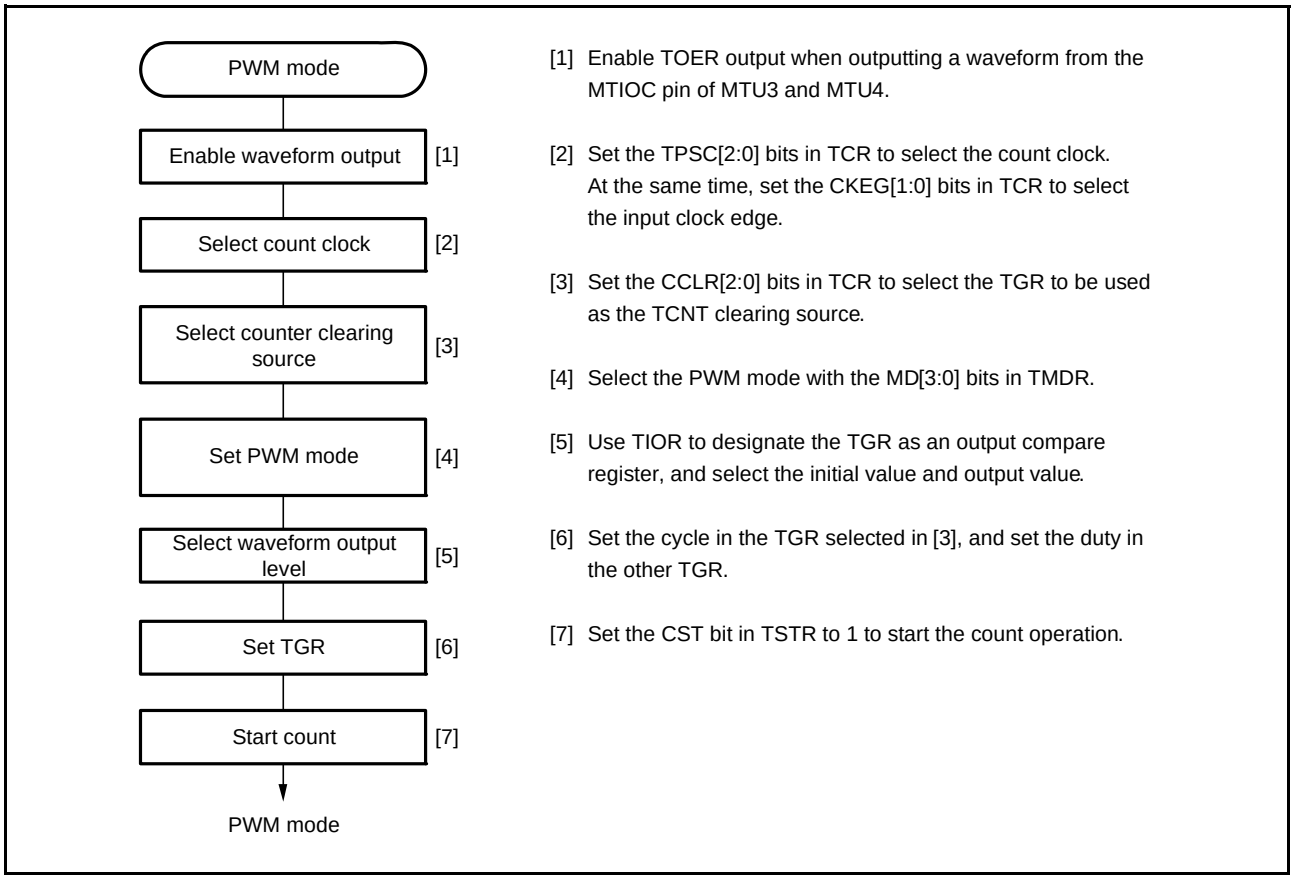


Figure 22.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 22.26 shows an example of operation 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.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty.

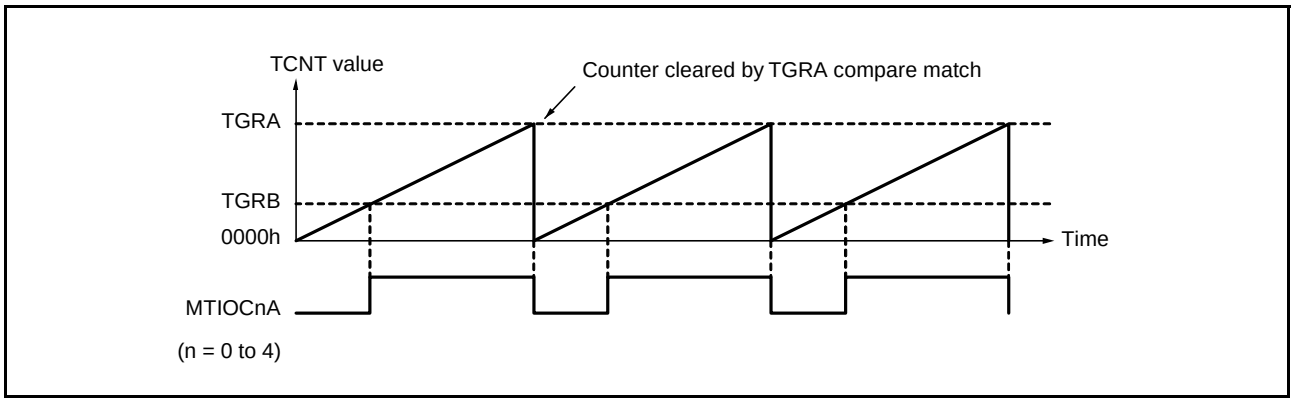


Figure 22.26 Example of PWM Mode Operation

Figure 22.27 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 a low level is set as the initial output value and a high level 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 cycle, and the values set in the other TGRs are used as the duty.

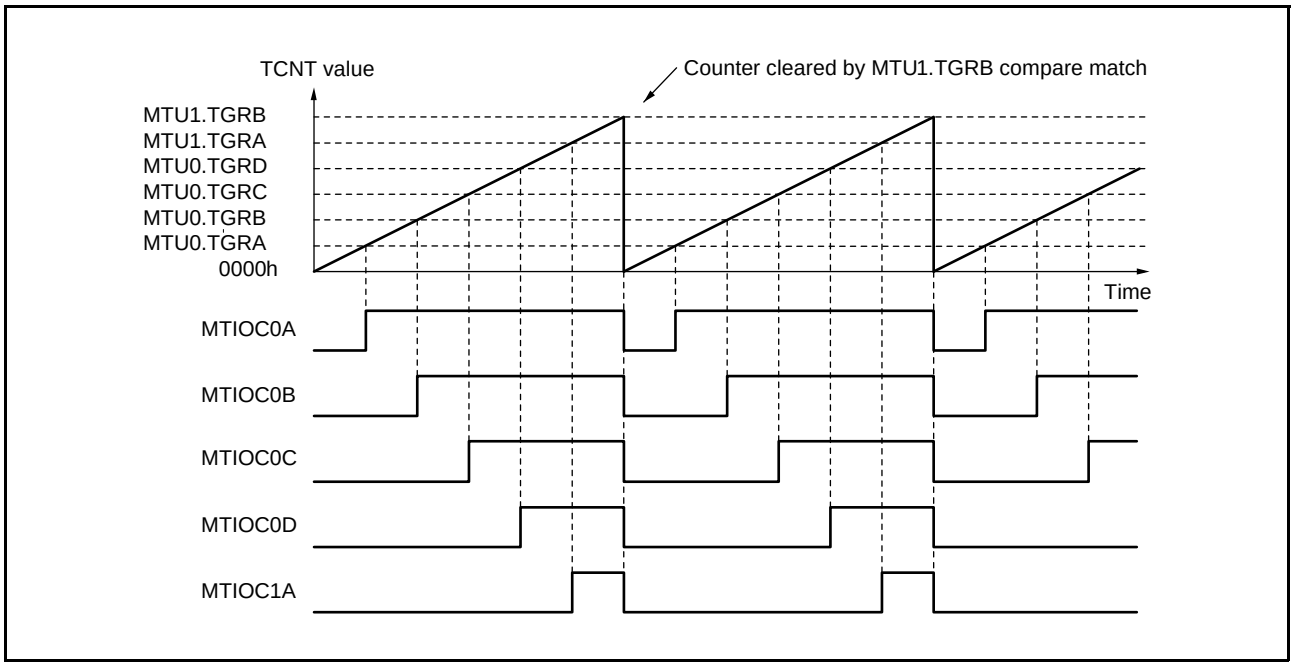


Figure 22.27 Example of PWM Mode Operation

Figure 22.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

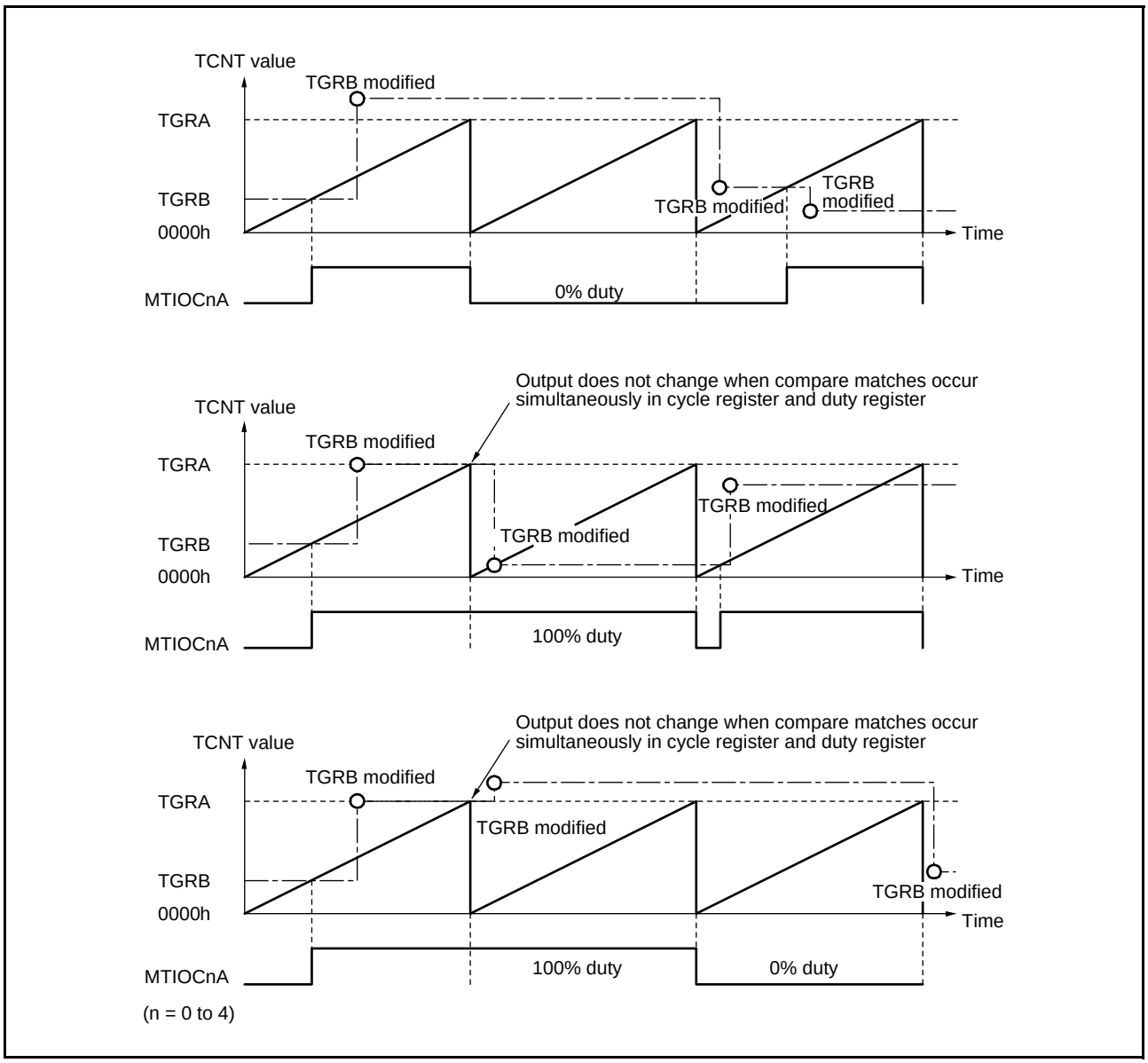


Figure 22.28 Examples of PWM Mode Operation

22.3.6 Phase Counting Mode

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up-counter/down-counter regardless of the setting of the TCR.TPSC[2:0] bits and TCR.CKEG[1:0] bits. However, the functions of the TCR.CCLR[2:0] bits and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.
This can be used for 2-phase encoder pulse input.
If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the corresponding TIER.TCIEV bit is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the corresponding TIER.TCIEU bit is 1.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.
In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD can be used as 2-phase encoder pulse input pins. Table 22.47 lists the correspondence between external clock pins and channels.

Table 22.47 Clock Input Pins in Phase Counting Mode

	External Clock Input Pins	
Channel	**A-Phase**	**B-Phase**
MTU1	MTCLKA	MTCLKB
MTU2	MTCLKC	MTCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 22.29 shows an example of the phase counting mode setting procedure.

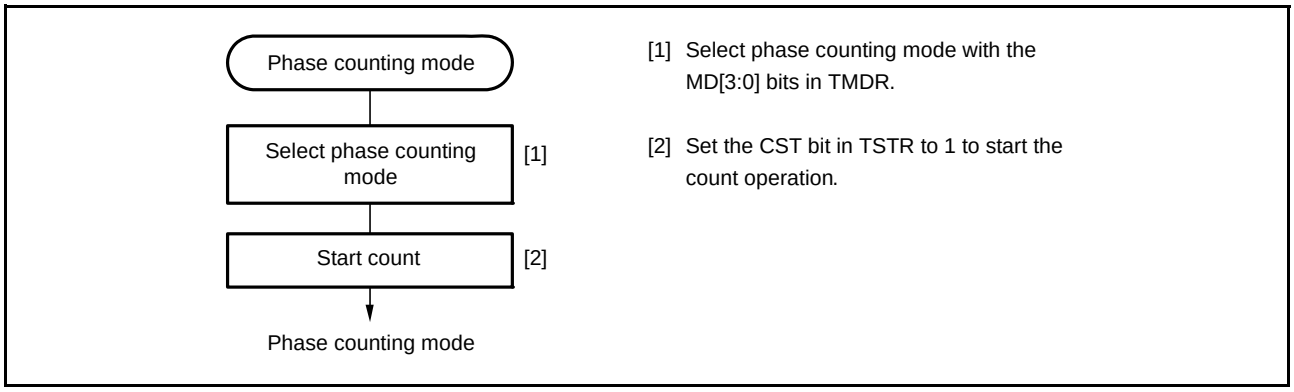


Figure 22.29 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase Counting Mode 1

Figure 22.30 shows an example of operation in phase counting mode 1, and Table 22.48 lists the TCNT up-counting and down-counting conditions.

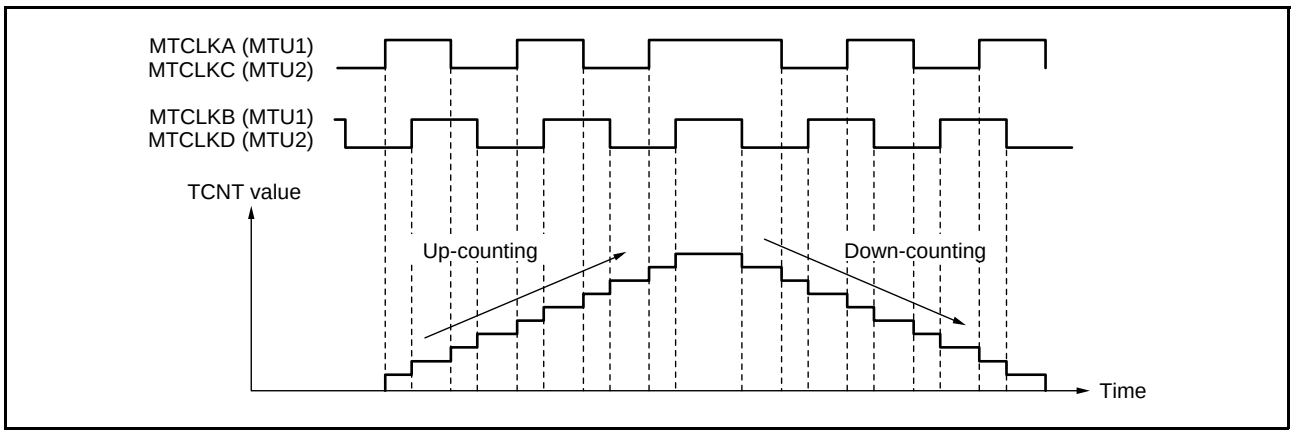


Figure 22.30 Example of Operation in Phase Counting Mode 1

Table 22.48 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 22.31 shows an example of operation in phase counting mode 2, and Table 22.49 lists the TCNT up-counting and down-counting conditions.

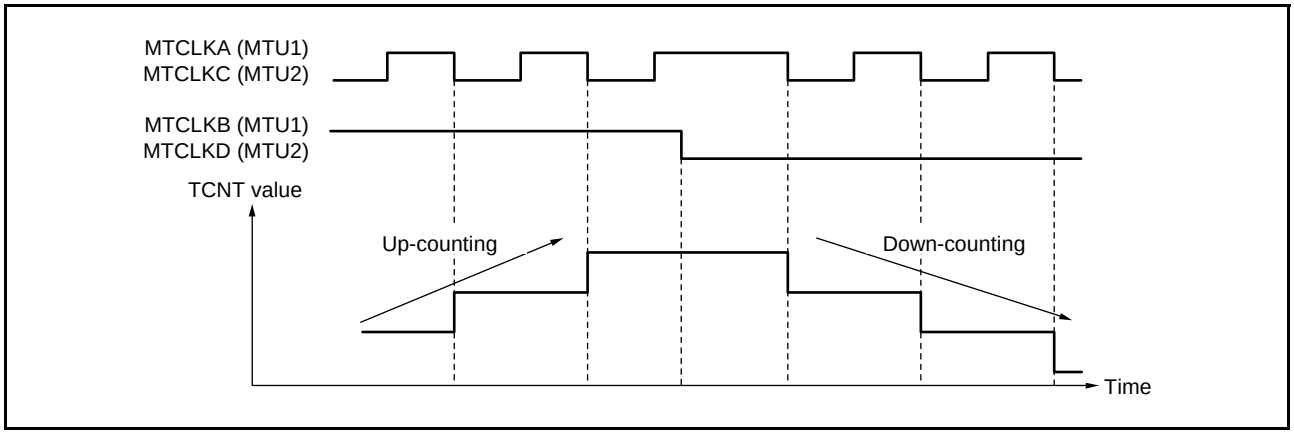


Figure 22.31 Example of Operation in Phase Counting Mode 2

Table 22.49 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	None (Don't care)
Low	↓	None (Don't care)
↑	Low	None (Don't care)
↓	High	Up-counting
High	↓	None (Don't care)
Low	↑	None (Don't care)
↑	High	None (Don't care)
↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 22.32 shows an example of operation in phase counting mode 3, and Table 22.50 lists the TCNT up-counting and down-counting conditions.

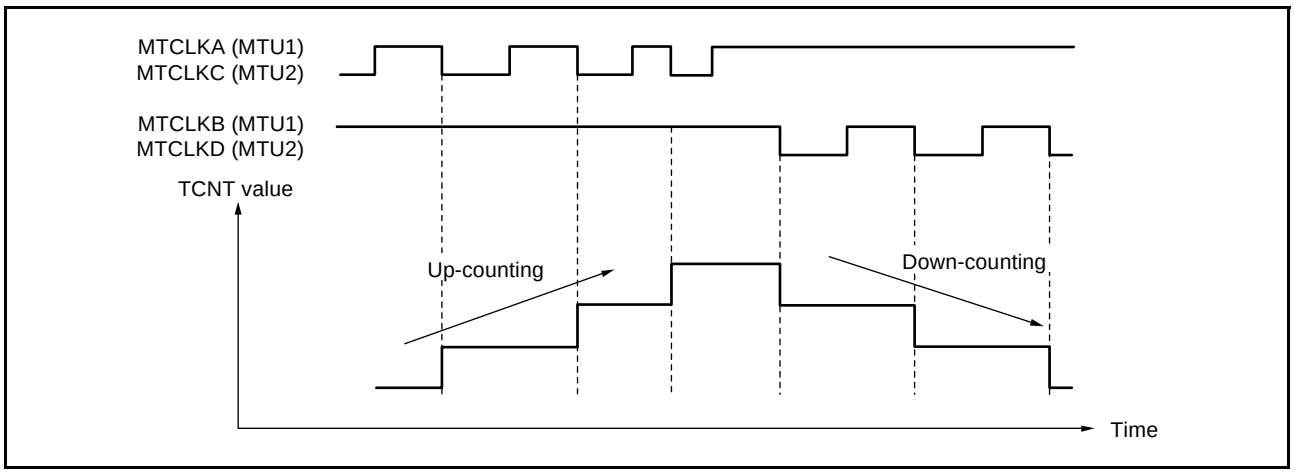


Figure 22.32 Example of Operation in Phase Counting Mode 3

Table 22.50 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	None (Don't care)
Low	↓	None (Don't care)
↑	Low	None (Don't care)
↓	High	Up-counting
High	↓	Down-counting
Low	↑	None (Don't care)
↑	High	None (Don't care)
↓	Low	None (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 22.33 shows an example of operation in phase counting mode 4, and Table 22.51 lists the TCNT up-counting and down-counting conditions.

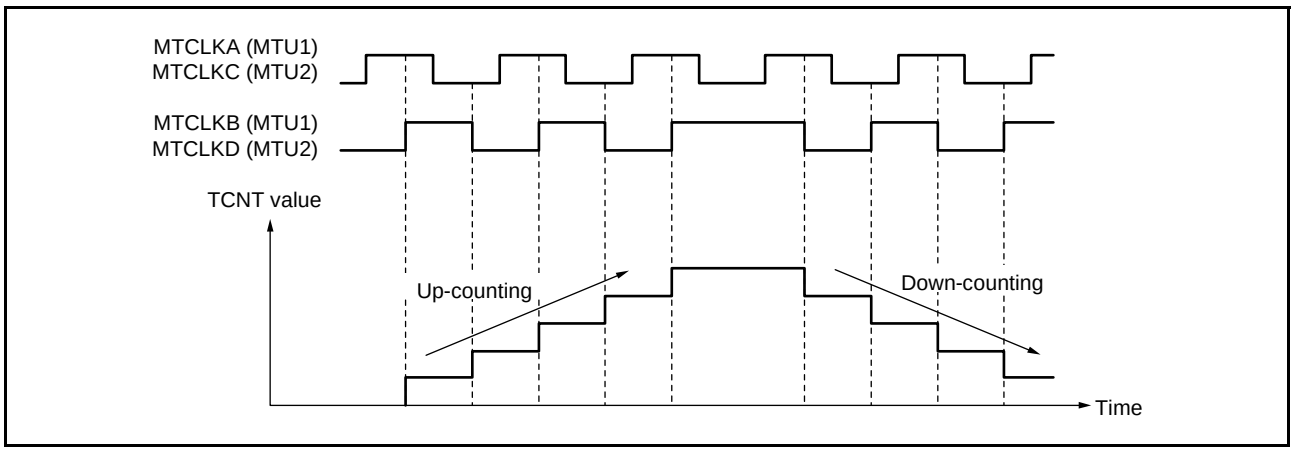


Figure 22.33 Example of Operation in Phase Counting Mode 4

Table 22.51 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	None (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	None (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

(3) Phase Counting Mode Application Example

Figure 22.34 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.

MTU0.TGRC compare match is specified as the MTU0.TCNT clearing source and MTU0.TGRA and TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 counter input 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-counter/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

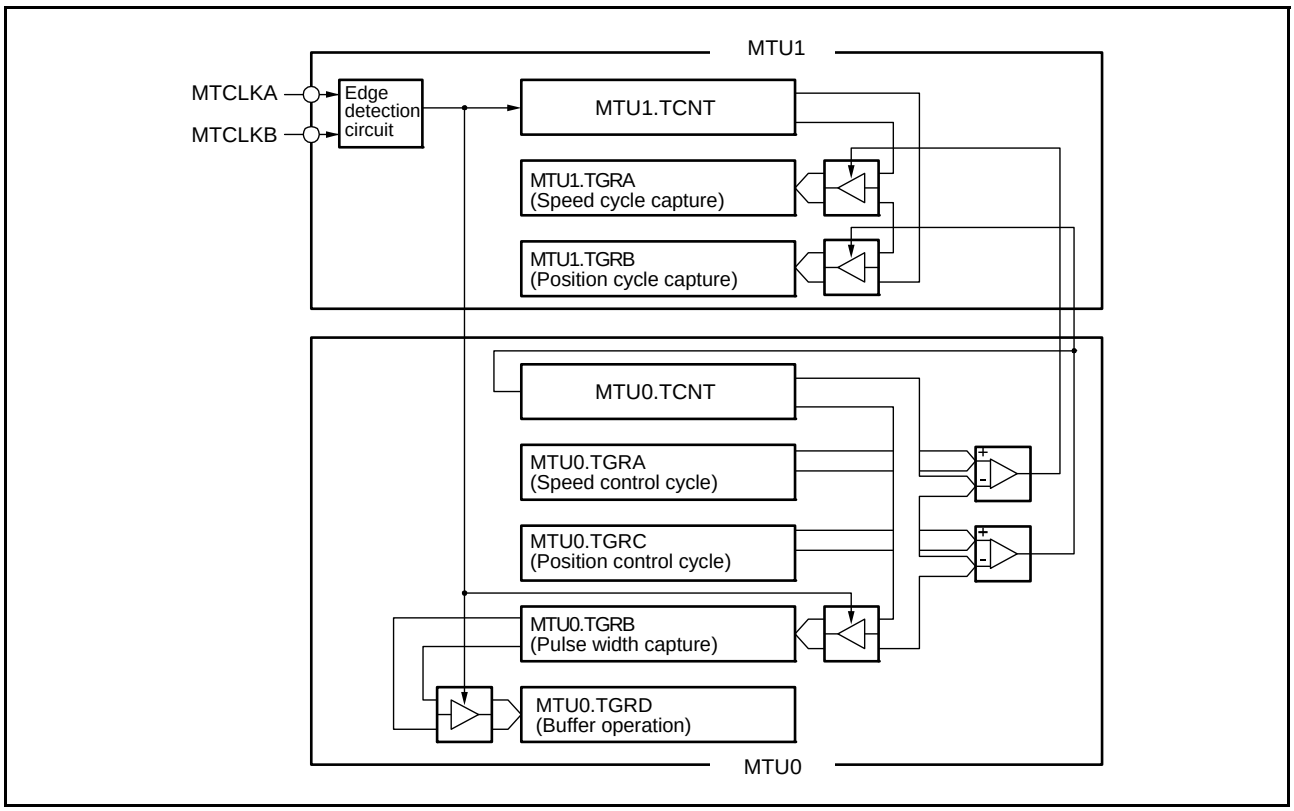


Figure 22.34 Phase Counting Mode Application Example

22.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms that share a common wave transition point can be output by combining MTU3 and MTU4.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D pins function as PWM output pins and the MTU3.TCNT counter functions as an up-counter.
Table 22.52 lists the PWM output pins. Table 22.53 lists the settings of the registers.

Table 22.52 Output Pins for Reset-Synchronized PWM Mode

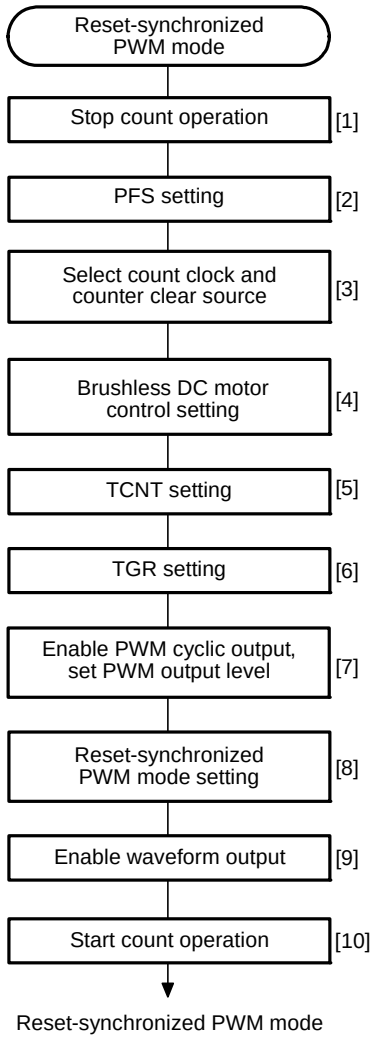
Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 22.53 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count cycle 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

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 22.35 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in TSTR to 0 to stop the TCNT operation. Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set bits TPSC[2:0] and CKEG[1:0] in the MTU3.TCR to select the count clock and clock edge for channel 3. Set the CCLR[2:0] bits in the MTU3.TCR to select TGRA compare-match as a counter clear source.

[4] When performing brushless DC motor control, set the BDC bit in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set MTU3.TCNT and MTU4.TCNT to 0000h.

[6] MTU3.TGRA is the cycle register. Set the waveform cycle value in MTU3.TGRA. Set the transition timing of the PWM output waveforms in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB. Set times within the compare match range of MTU3.TCNT.

X ≤ MTU3.TGRA (X: Setting value)

[7] Enable or disable toggle output synchronized with the PWM cycle using the PSYE bit in TOCR1, and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output Level. In buffer operation of TOCR2 and TOLBR, see Figure 22.3.

[8] Set the MD[3:0] bits in the timer mode register (MTU3.TMDR) to 1000b to select the reset-synchronized PWM mode. Set the BFA bit to 1 for buffer operation of TGRA and TGRC, and the BFB bit to 1 for buffer operation of TGRB and TGRD, if necessary.
Set the initial value in MTU4.TMDR.

[9] Set the enabling/disabling of the PWM waveform output pin in TOER.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA, i.e., cycle = duty cycle.

Figure 22.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 22.36 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT and MTU3.TGRA, and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and the counters are cleared.

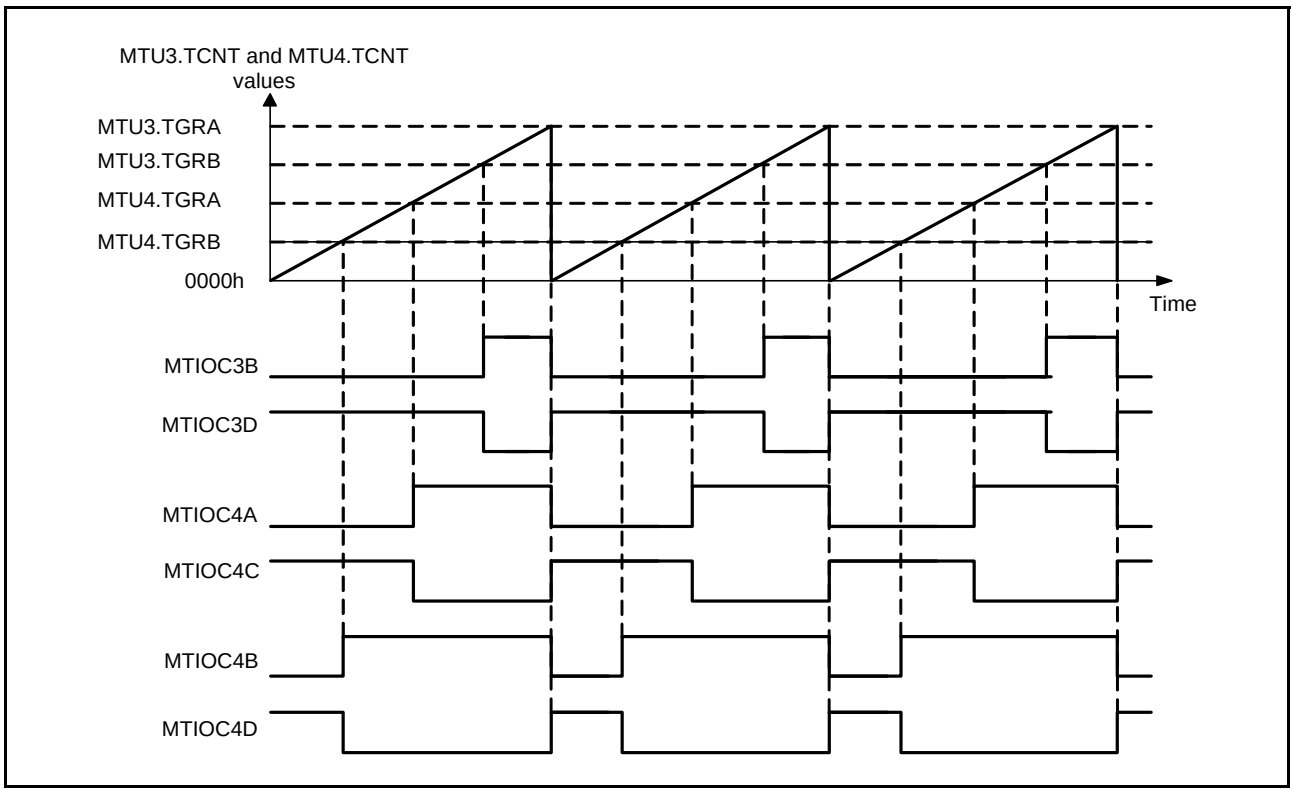


Figure 22.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1.OLSN = 1 and OLSP = 1)

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 phases of positive and negative PWM waveforms with dead time can be output by combining MTU3 and MTU4. PWM waveforms without dead time can also be output.
In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle. MTU3.TCNT and MTU4.TCNT function as up/down-counters.
Table 22.54 lists the PWM output pins used. Table 22.55 lists 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 22.54 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM cycle (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 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 3)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 22.55 Register Settings for Complementary PWM Mode

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	MTU3.TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWER setting*1
	MTU3.TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*1
	MTU3.TGRB	PWM output 1 compare register	Maskable by TRWER setting*1
	MTU3.TGRC	MTU3.TGRA buffer register	Readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
MTU4	MTU4.TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWER setting*1
	MTU4.TGRA	PWM output 2 compare register	Maskable by TRWER setting*1
	MTU4.TGRB	PWM output 3 compare register	Maskable by TRWER setting*1
	MTU4.TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
Timer dead time data register (TDDR)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWER setting*1
Timer cycle data register (TCDR)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*1
Timer cycle buffer register (TCBR)		TCDR buffer register	Readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/MTU3.TGRB temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/MTU4.TGRA temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/MTU4.TGRB temporary register	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWER (timer read/write enable register).

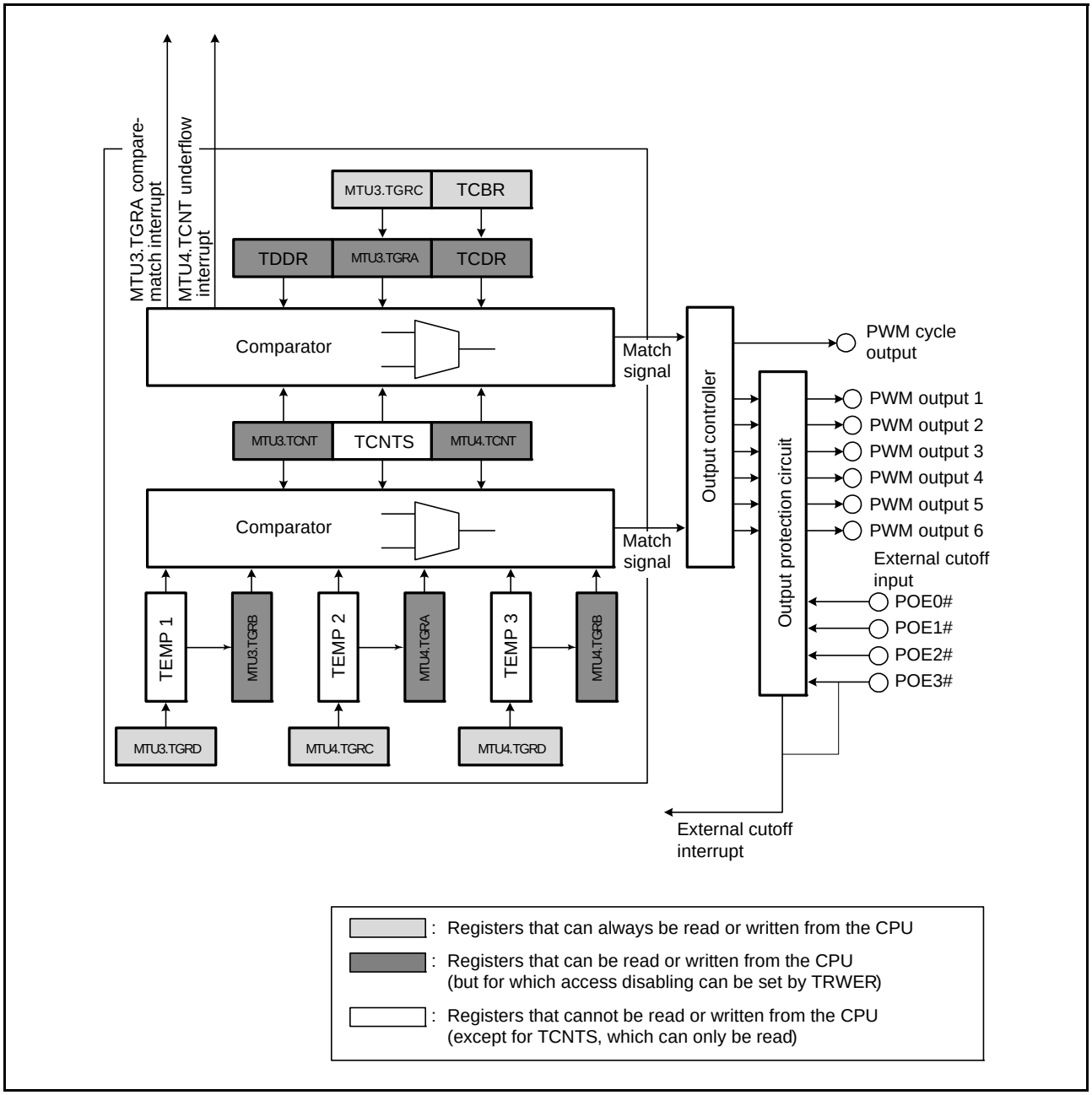
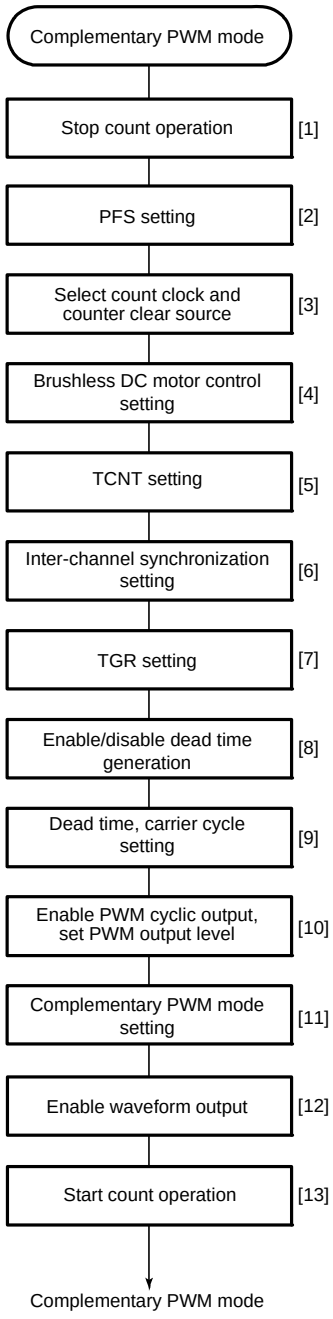


Figure 22.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 22.38 shows an example of the complementary PWM mode setting procedure.



[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0 to stop timer counter (TCNT) operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set the same count clock and clock edge for MTU3 and MTU4 with bits TPSC[2:0] and bits CKEG[1:0] in the timer control register (TCR). Use the CCLR[2:0] bits to set synchronous clearing only when restarting by synchronous clearing with another channel during complementary PWM mode operation.

[4] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0000h.

[6] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the timer synchronous register (TSYR).

[7] Set the output PWM duty cycles in the compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Set the same value in the buffer registers and the compare registers.

[8] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[9] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the timer cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in MTU3.TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[10] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN.
When specifying the PWM output level by using TOLBR as a buffer for MTU.TOCR2, see Figure 22.3, Example of PWM Output Level Setting Procedure in Buffer Operation.

[11] Select complementary PWM mode in the timer mode register (MTU3.TMDR). Set the BFA bit in TGRA and TGRC to 1 for buffer operation of TGRA and TGRC, and the BFB bit to 1 for buffer operation of TGRB and TGRD, if necessary. Set nothing in MTU4.TMDR.

[12] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 22.38 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) of PWM waveforms can be output. Figure 22.39 illustrates counter operation in complementary PWM mode, and Figure 22.40 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTS— perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the TSTR.CST bit is 0.

When the CST bit is set to 1, MTU3.TCNT counts up to the value set in MTU3.TGRA, then switches to down-counting when it matches MTU3.TGRA. When the MTU3.TCNT value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h.

When the CST bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when it matches TCDR. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It does not need to be initialized.

When MTU3.TCNT matches TCDR during up-/down-counting of TCNT in MTU3 and MTU4, TCNTS starts down-counting, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches MTU3.TGRA, it is cleared to 0000h.

When MTU4.TCNT matches TDDR during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTS starts up-counting, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches 0000h, it is set with the value in MTU3.TGRA.

TCNTS is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

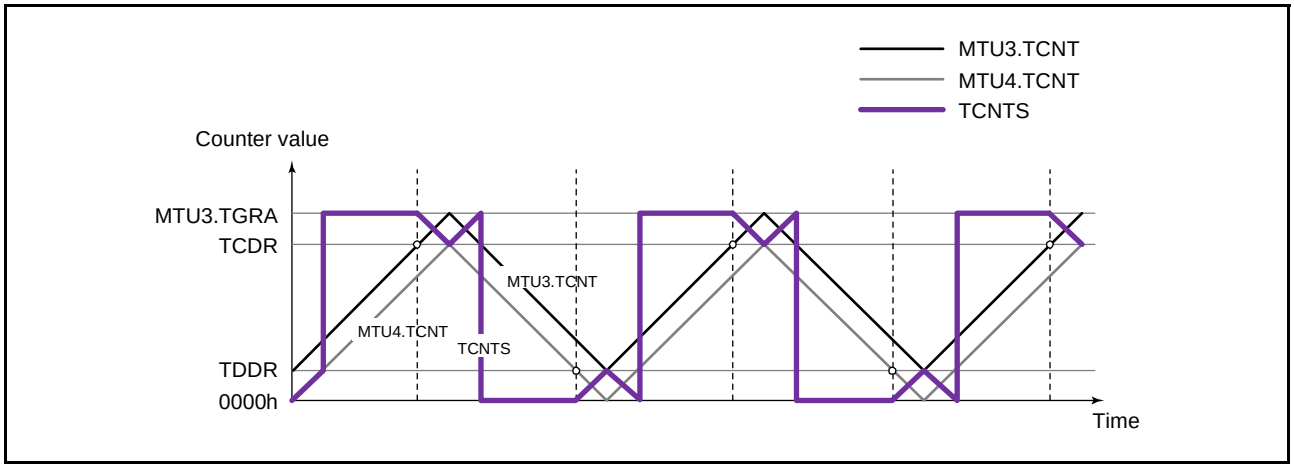


Figure 22.39 Counter Operation in Complementary PWM Mode

(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 22.40 shows an example of operation in complementary PWM mode.
MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the TOCR1.OLSN and OLSP bits is output from the PWM output pin.
MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers. Between a buffer register and a compare register, there is a temporary register. The temporary registers cannot be accessed by the CPU. 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, be sure to write to MTU4.TGRD last and enable data transfer from the buffer register to a temporary register. At this time, transfer from TCBR and MTU3.TGRC, which operate as buffer registers for the timer cycle 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 immediately transferred to the temporary register. 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 while TCNTS is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTS is counting down). The timing for transfer from the temporary register to the compare register can be selected with the TMDR.MD[3:0] bits. Figure 22.40 shows an example in which the trough is selected for the transfer timing.
In the Tb (Tb2 in Figure 22.40) interval in which data is not transferred to the temporary register, 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 TCNTS) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

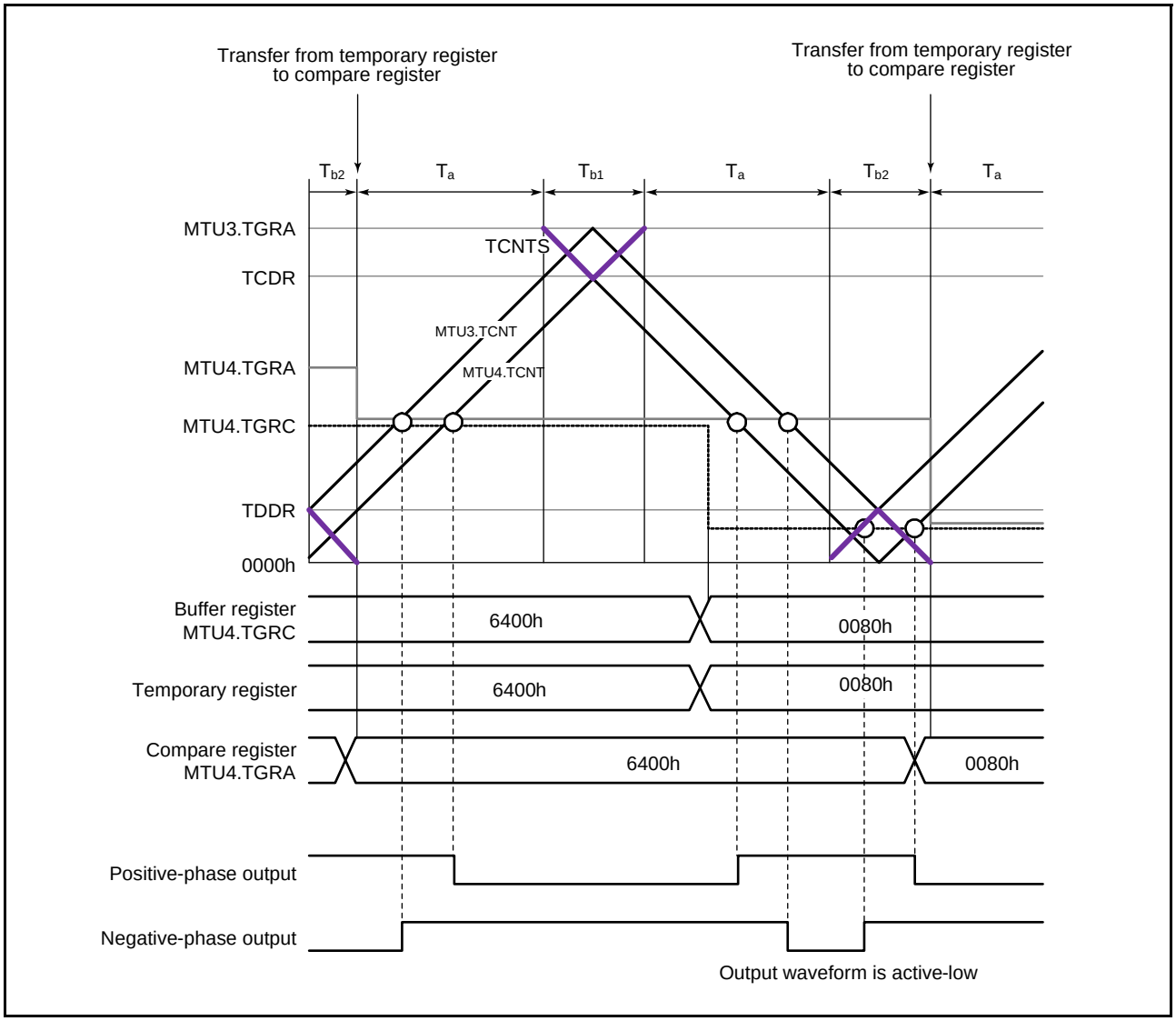


Figure 22.40 Example of Operation in Complementary PWM Mode

(c) Initial Setting

In complementary PWM mode, there are six 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 the TMDR.MD[3:0] bits, initial values should be set in the following registers.
MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM cycle + 1, and TDDR should be set to 1.
Set the respective initial PWM duty values in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD. The values set in the five buffer registers excluding TDDR are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT to 0000h before setting complementary PWM mode.

Table 22.56 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
MTU3.TGRC	1/2 PWM cycle + dead time Td (1/2 PWM cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty value for each phase
MTU4.TCNT	0000h

Note: The value set in MTU3.TGRC should be the sum of 1/2 the PWM cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC should be set to 1/2 the PWM cycle + 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 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).
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 (TDDR). The value set in TDDR is used as the MTU3.TCNT counter start value and creates a dead time between MTU3.TCNT and MTU4.TCNT. Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER.TDER bit to 0. The TDER bit can be cleared to 0 only when 0 is written to it after reading it as 1.
MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM cycle + 1 and the TDDR register should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 22.41 shows an example of operation without dead time.

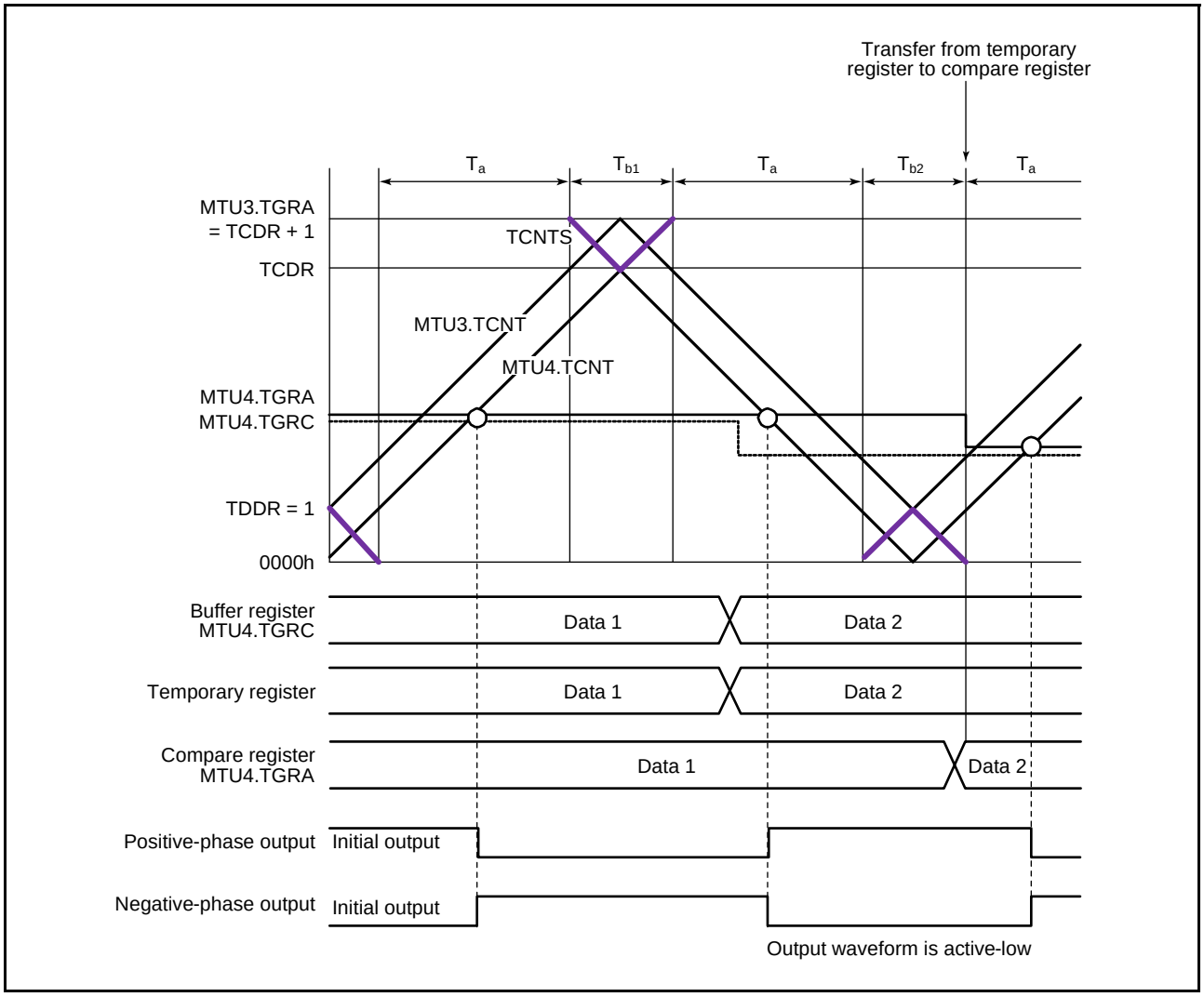


Figure 22.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM cycle is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDR, in which the MTU4.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 setting = TCDR setting + TDDR setting
Without dead time: MTU3.TGRA setting = TCDR setting + 1

The settings should be made so as to achieve the following relationship between registers TCDR and TDDR.
TCDR setting > TDDR setting × 2 + 2

The MTU3.TGRA and TCDR settings are made by setting values in buffer registers MTU3.TGRC and TCBR. When data is written to MTU4.TGRD to enable transfers, the values set in MTU3.TGRC and TCBR are transferred simultaneously to MTU3.TGRA and TCDR with the transfer timing selected with the TMDR.MD[3:0] bits.
The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 22.42 illustrates the operation when the PWM cycle 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.

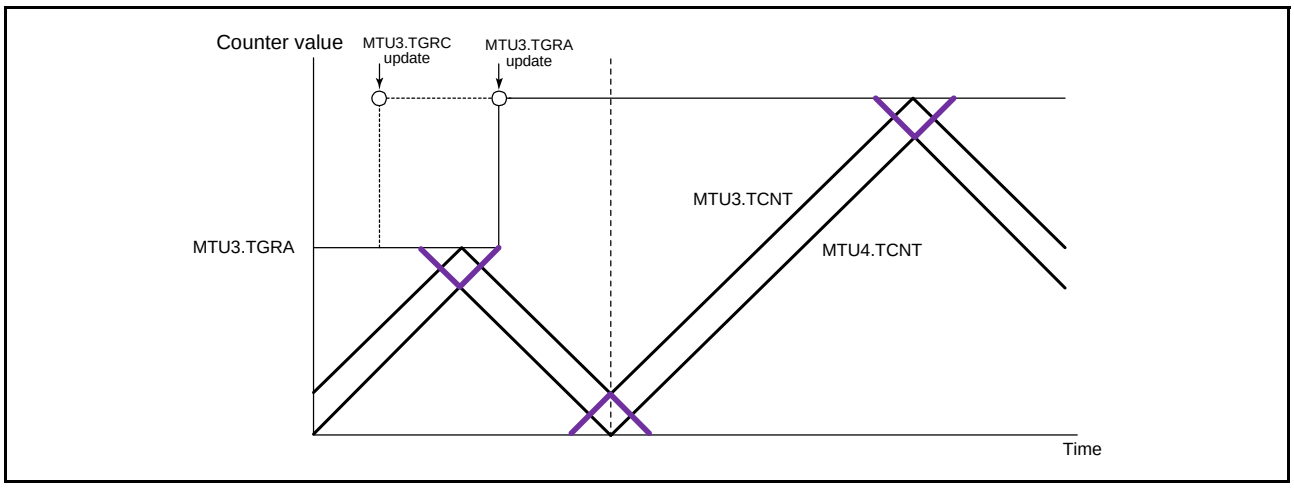


Figure 22.42 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and PMW cycle registers) that have buffer registers and can be updated during operation.
There is a temporary register between each of these registers and its buffer register. While subcounter TCNTS 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 TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.
The temporary register value is transferred to the compare register at the data update timing set with the TMDR.MD[3:0] bits. Figure 22.43 shows an example of data updating in complementary PWM mode. 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 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.
Even when not updating all five registers or when not updating the MTU4.TGRD data, be sure to write to MTU4.TGRD after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD should be the same as the data prior to the write operation.

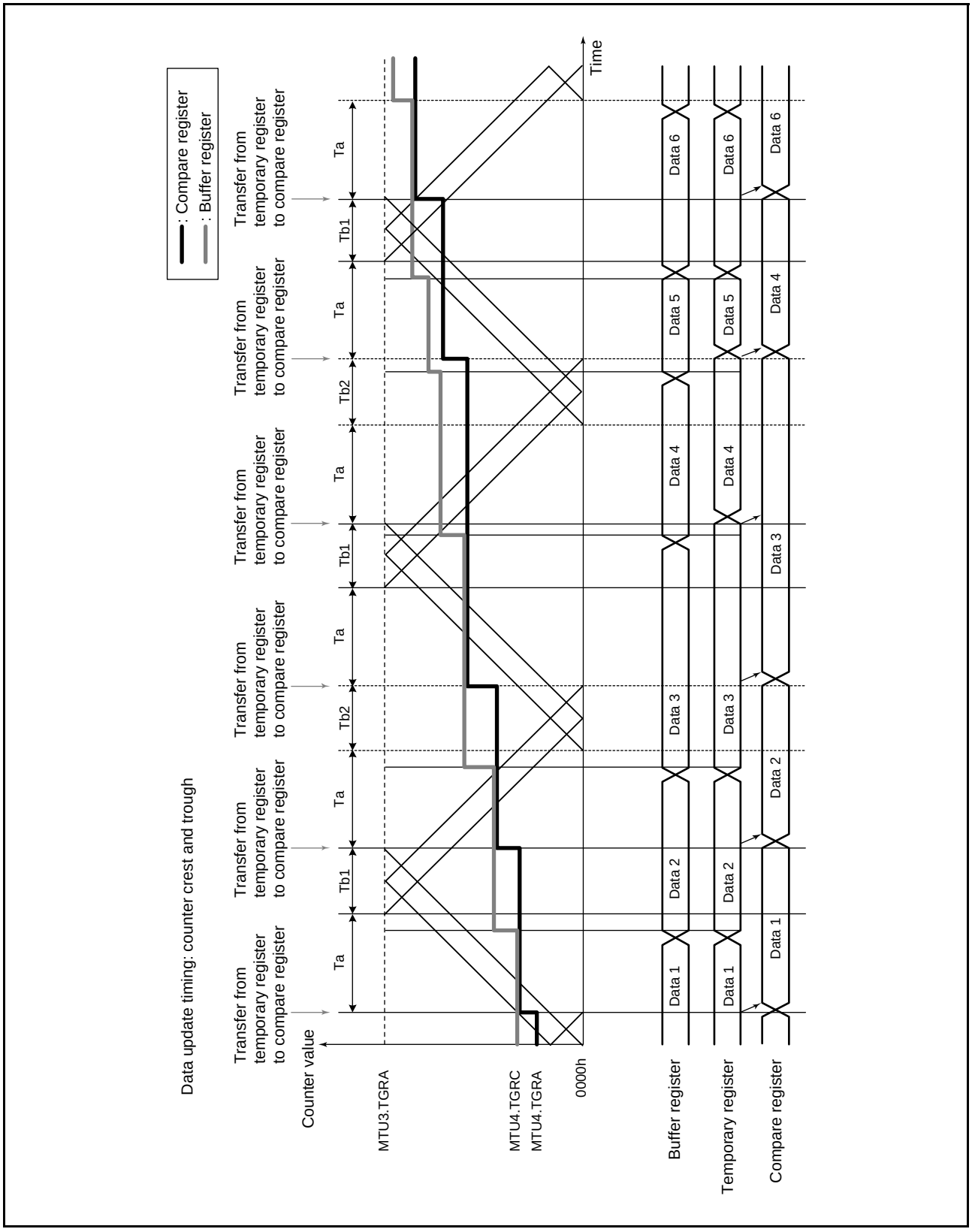


Figure 22.43 Example of Data Updating in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2). This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the timer mode register (TMDR) until MTU4.TCNT exceeds the value set in the dead time register (TDDR). Figure 22.44 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 22.45.

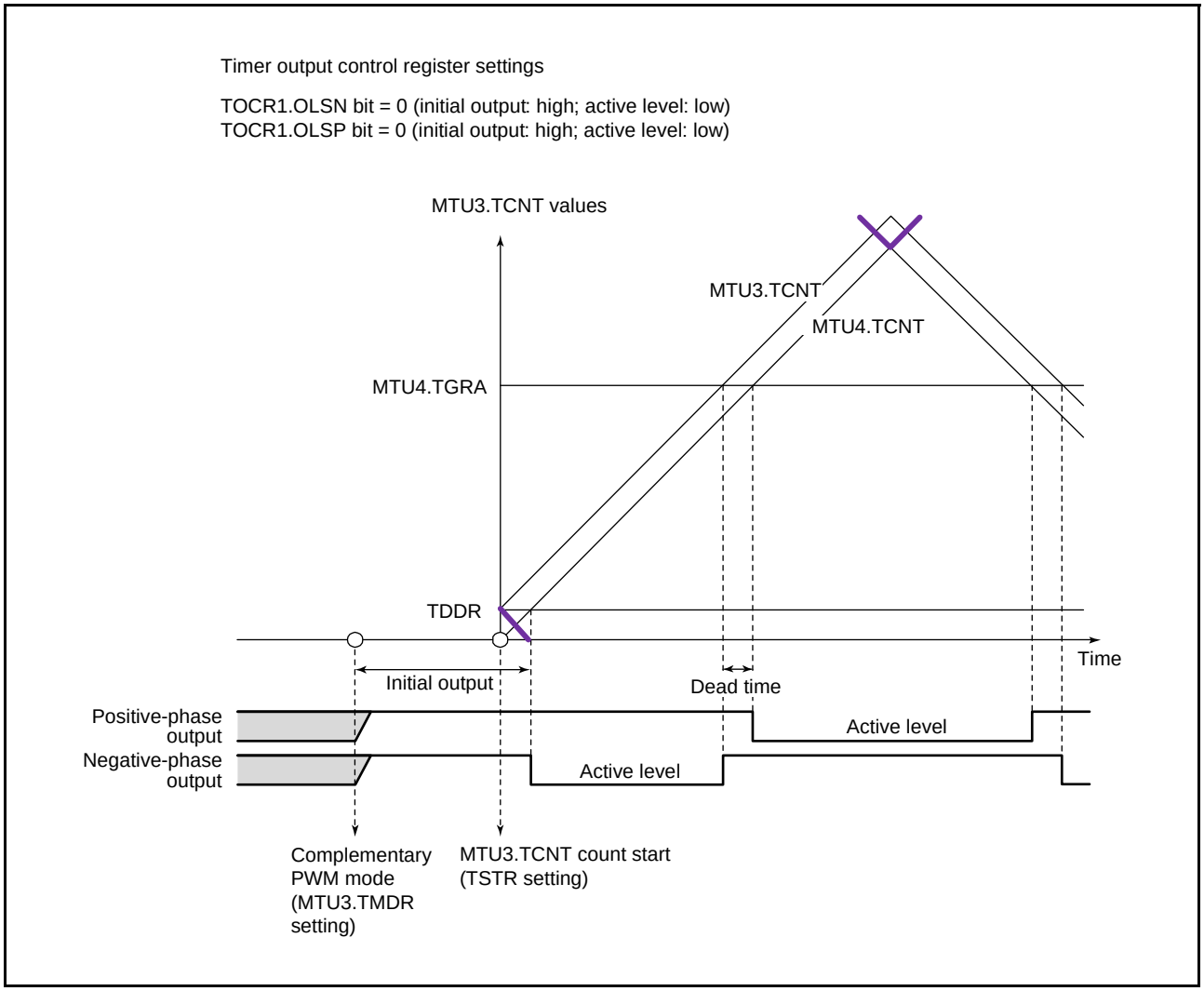


Figure 22.44 Example of Initial Output in Complementary PWM Mode (1)

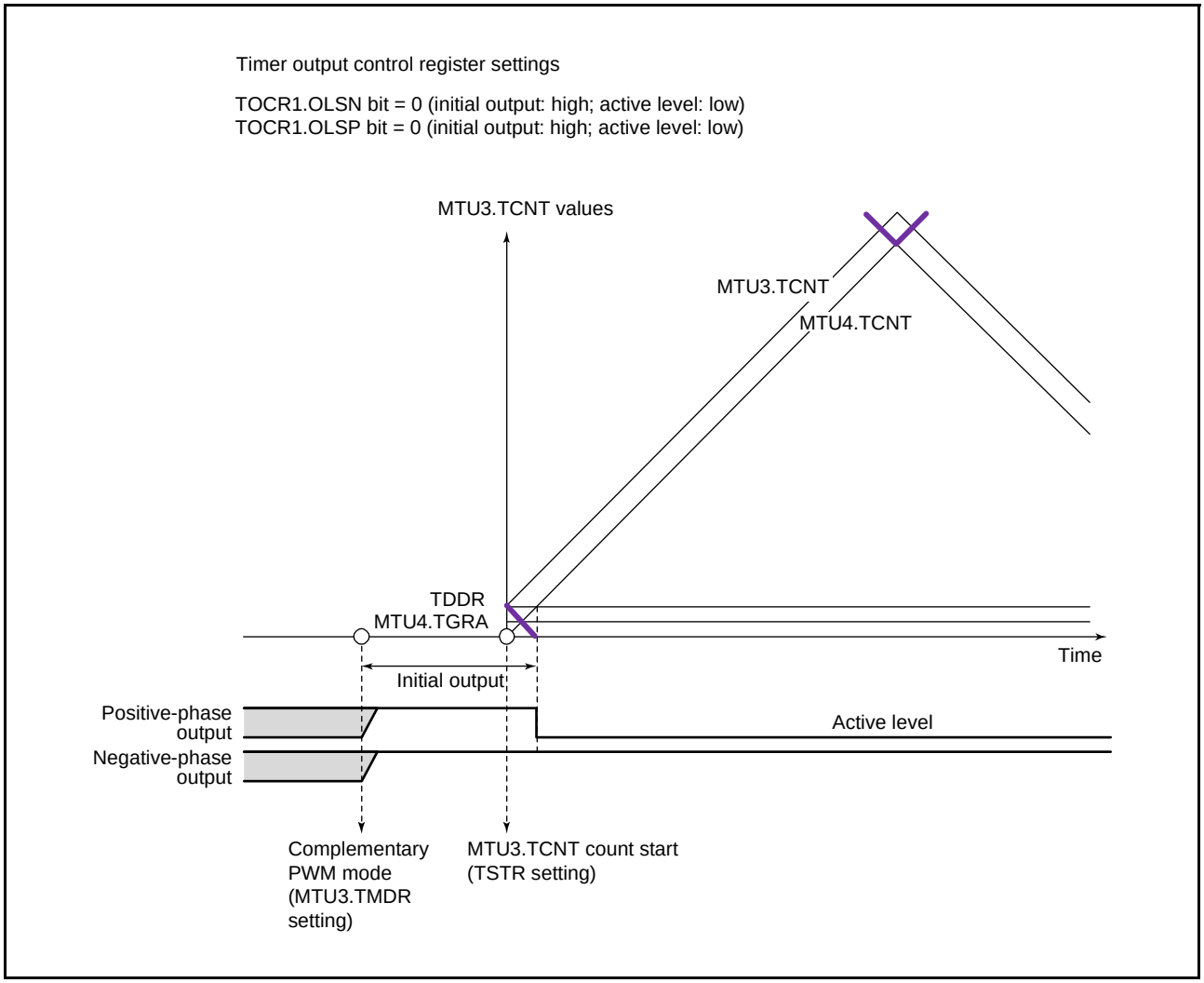


Figure 22.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) of 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 TCNTS is counting, the compare register and temporary register values are simultaneously compared to generate consecutive PWM waveforms from 0 to 100% duty cycle. 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 22.46 to Figure 22.48 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 22.46.

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 22.47, 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 higher 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 22.48, 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.

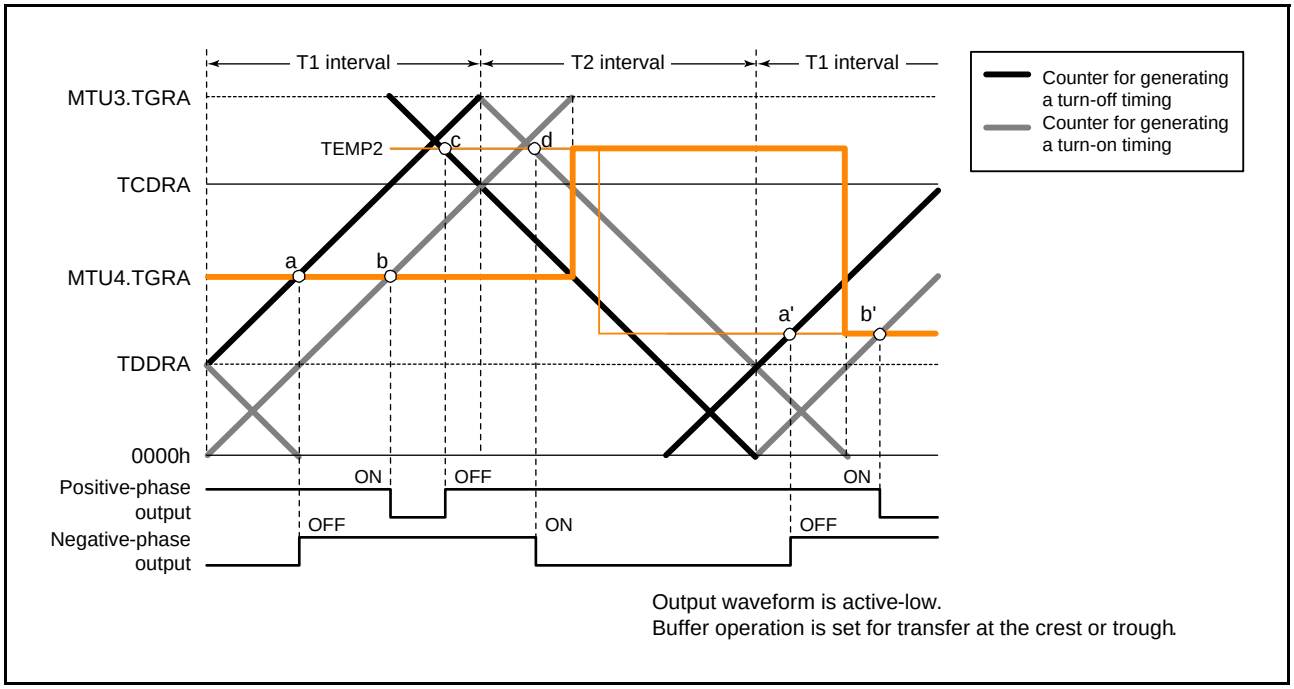


Figure 22.46 Example of Waveform Output in Complementary PWM Mode (1)

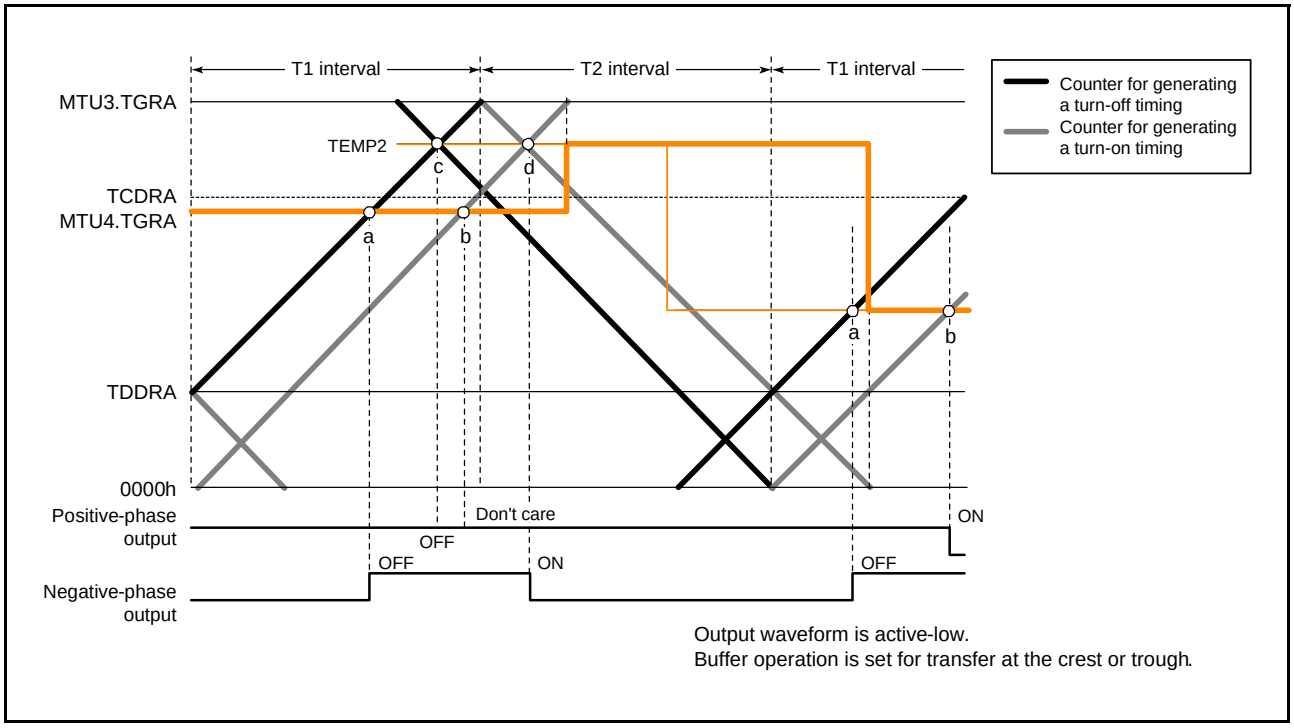


Figure 22.47 Example of Waveform Output in Complementary PWM Mode (2)

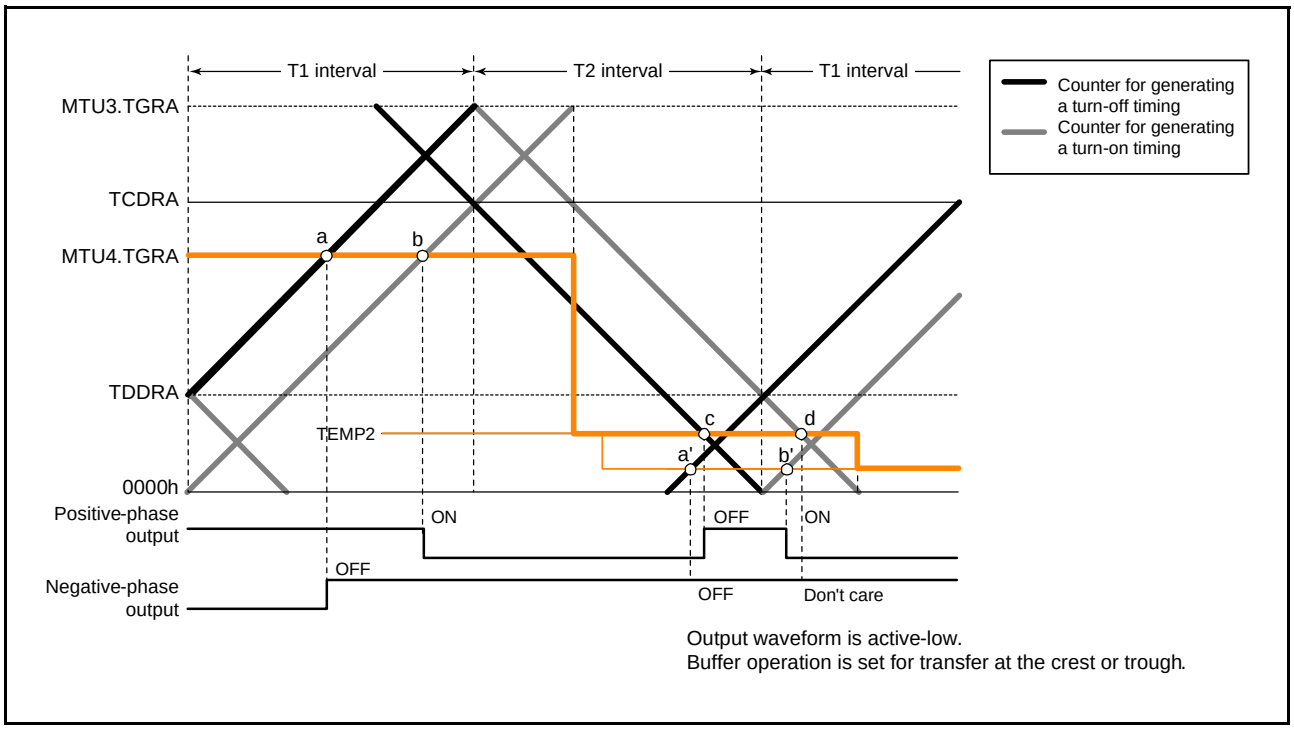


Figure 22.48 Example of Waveform Output in Complementary PWM Mode (3)

(k) 0% and 100% Duty Cycle Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty cycle PWM waveforms can be output as required. Figure 22.49 to Figure 22.53 show output examples.

A 100% duty cycle waveform is output when the data register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty cycle waveform is output when the data register value is set to the same value as MTU3.TGRA. The waveform in this case has a positive phase with a 100% off-state.

On and 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.

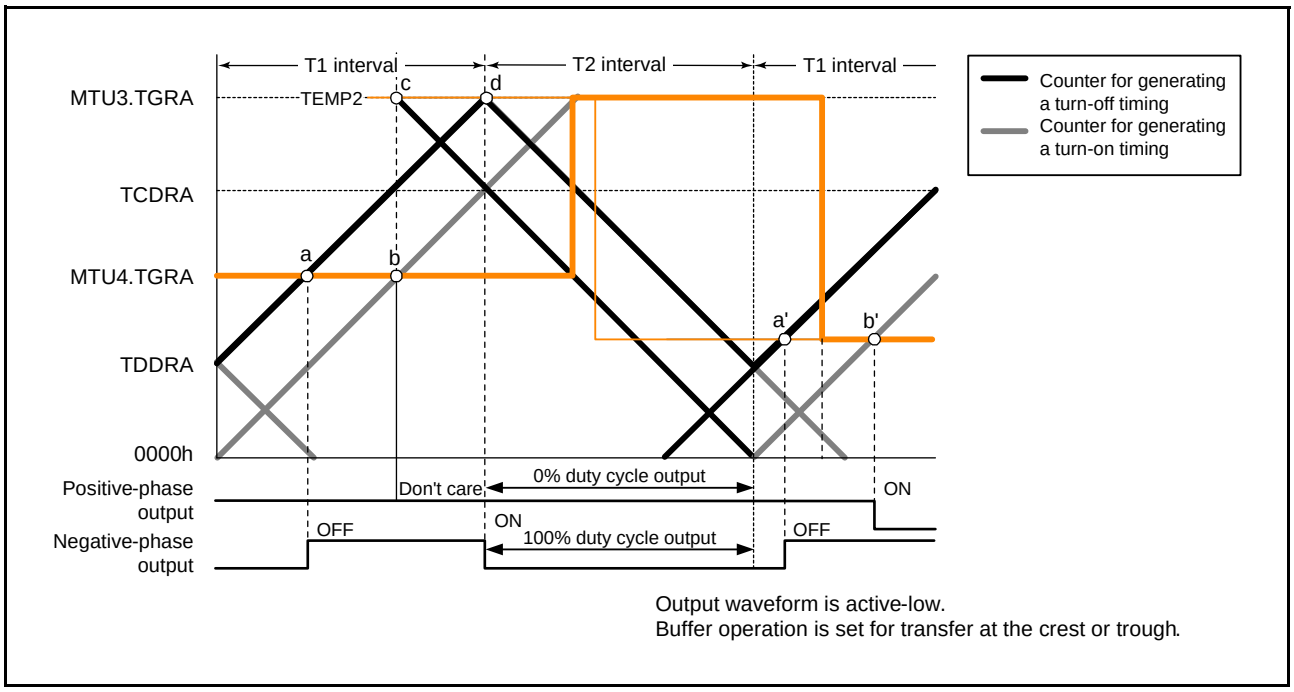


Figure 22.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

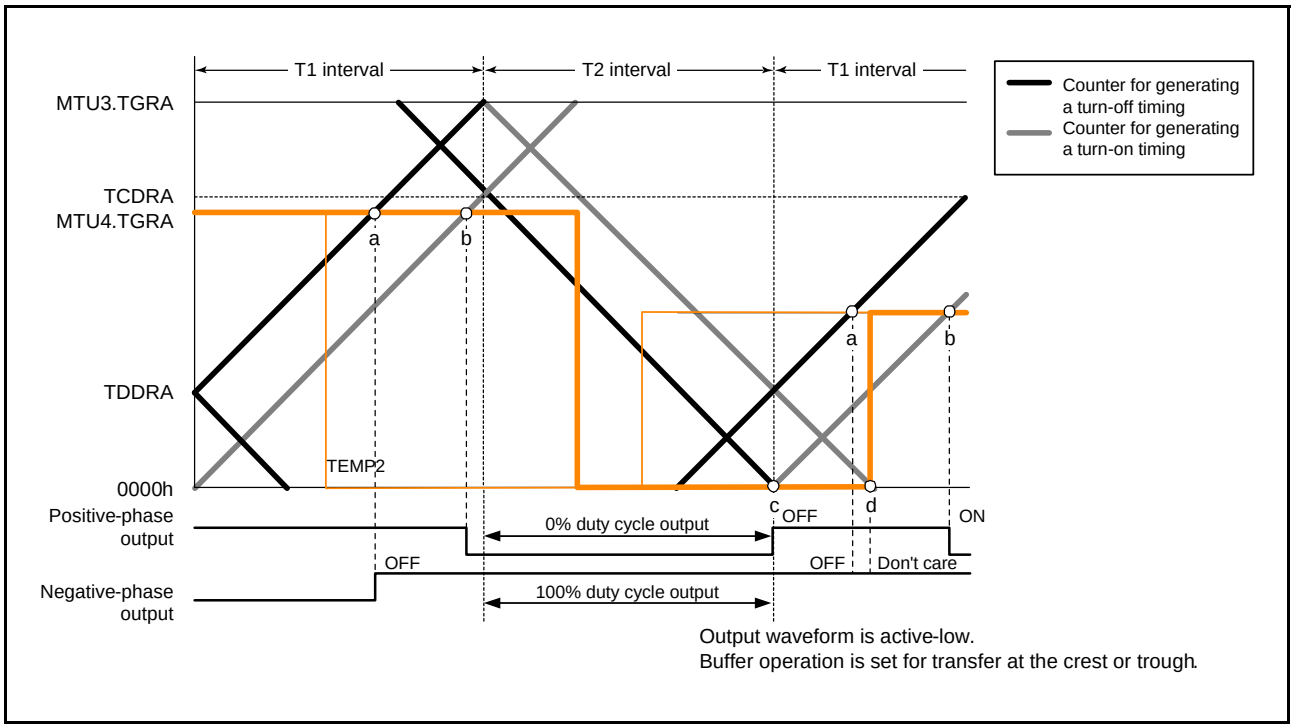


Figure 22.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

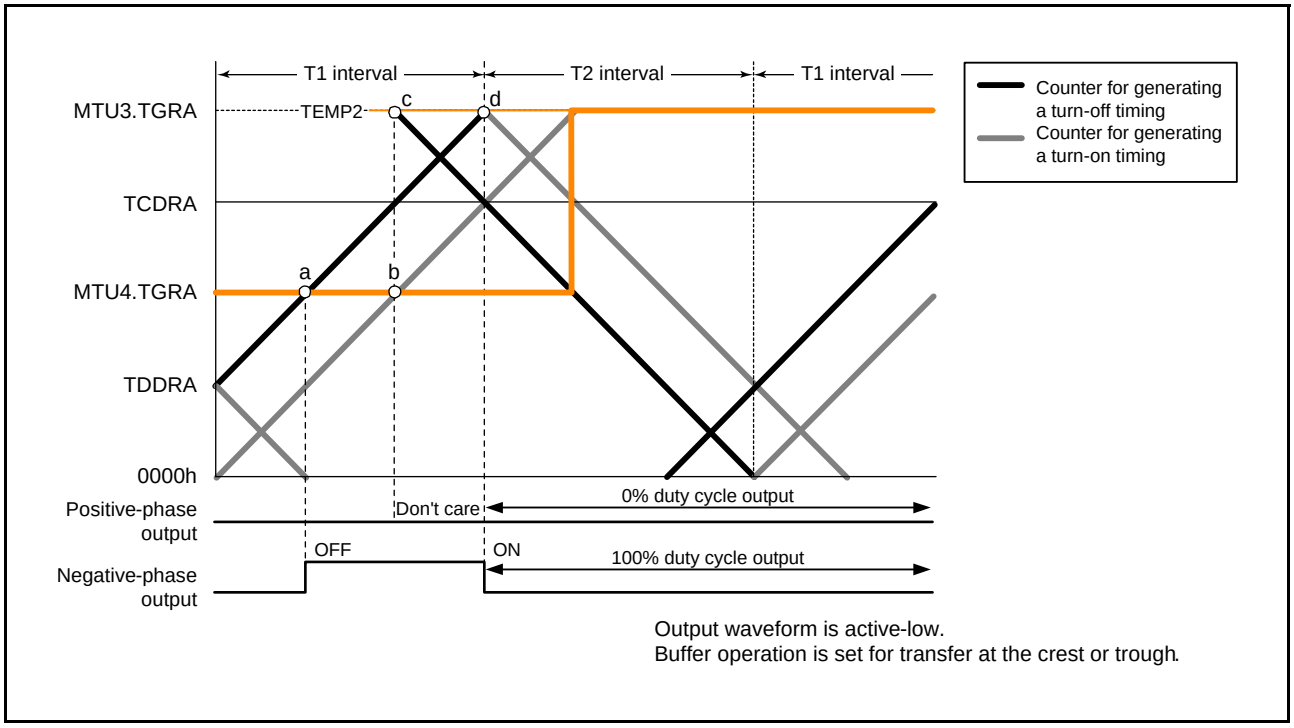


Figure 22.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

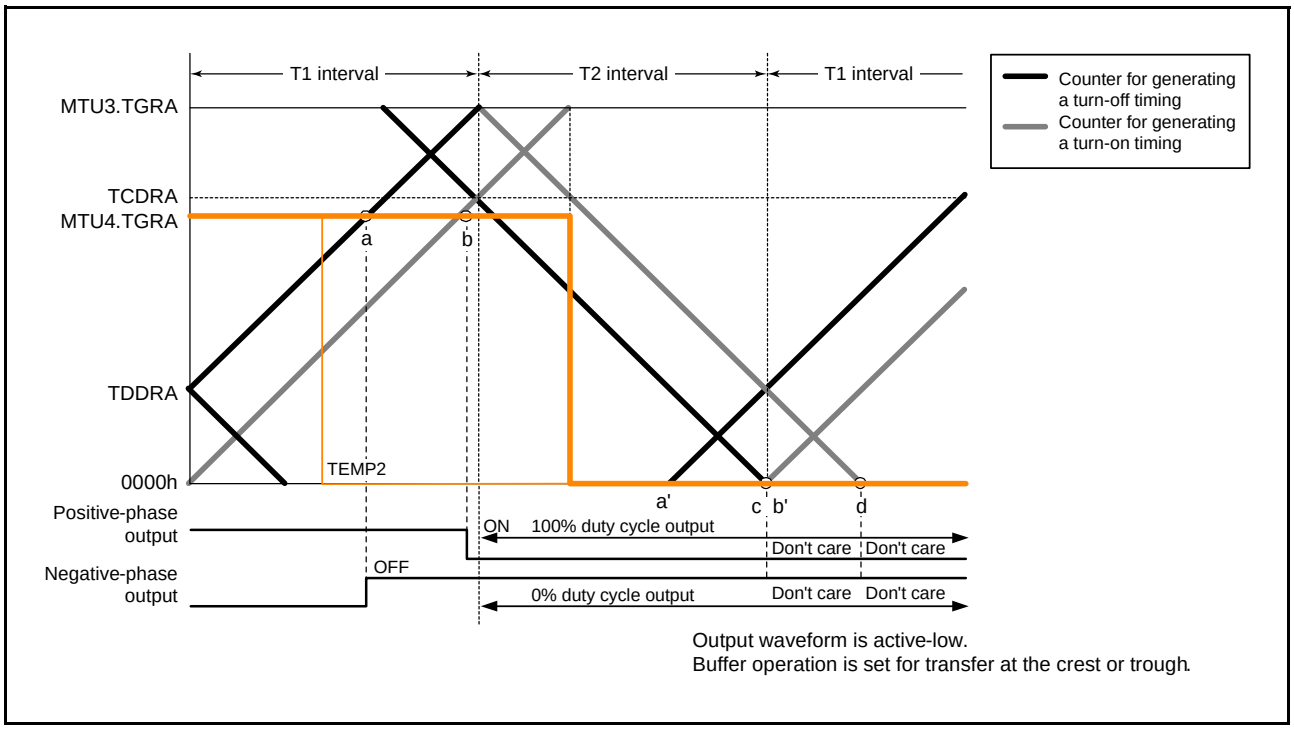


Figure 22.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

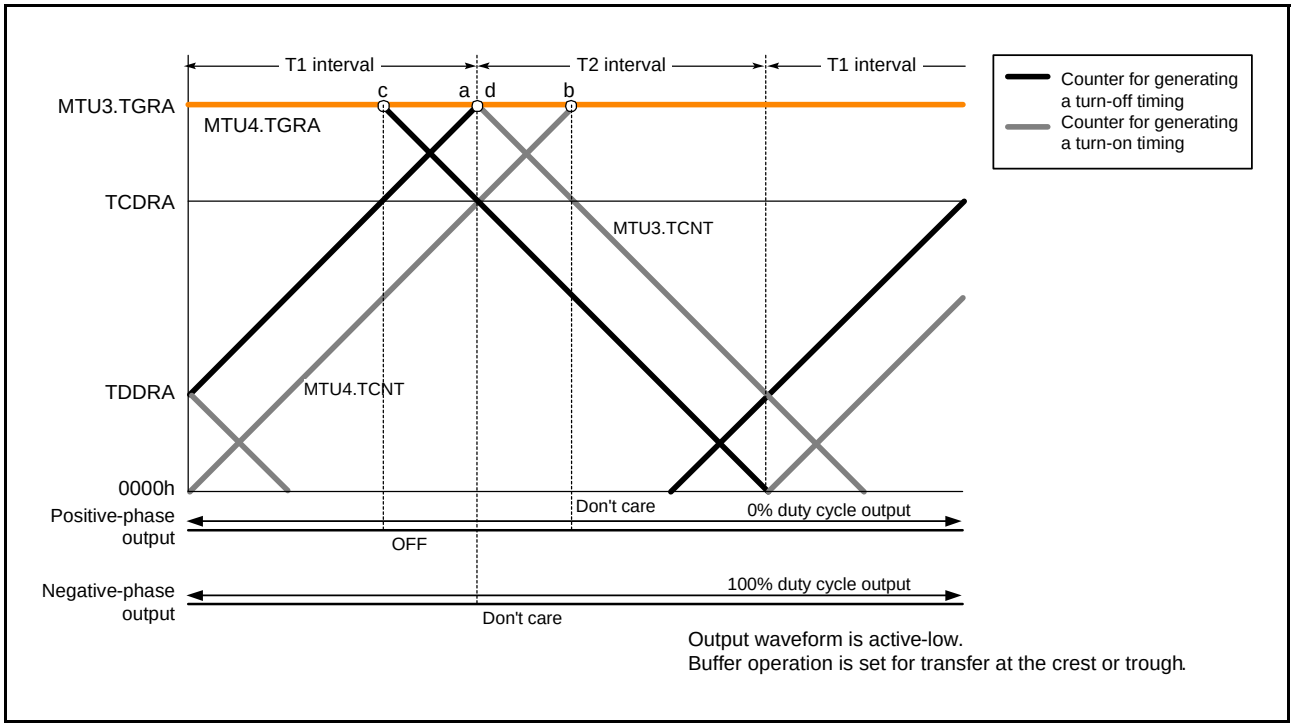


Figure 22.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the TOCR1.PSYE bit to 1. An example of a toggle output waveform is shown in Figure 22.54. This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA and a compare match between MTU4.TCNT and 0000h.

The MTIOC3A pin is assigned for this toggle output. The initial output is a high level.

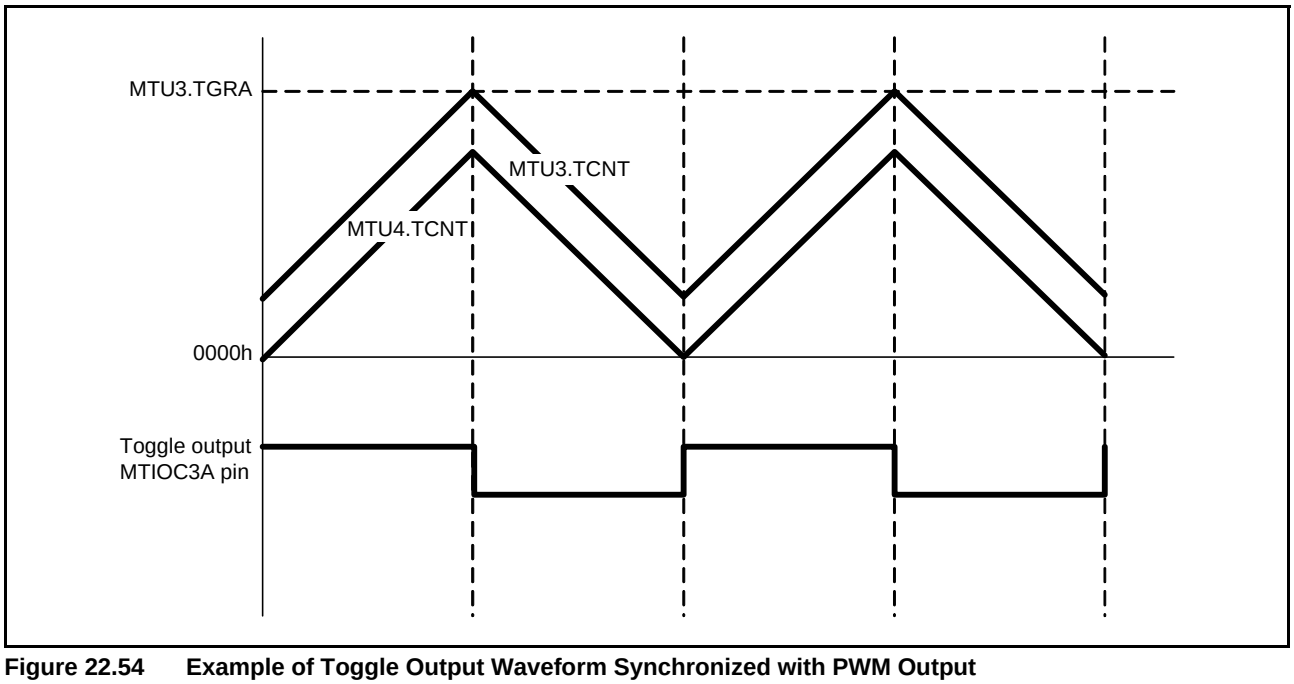


Figure 22.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel source when a mode for synchronization with another channel is specified by TSYR and synchronous clearing is selected with the MTU3.TCR.CCLR[2:0] bits.

Figure 22.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

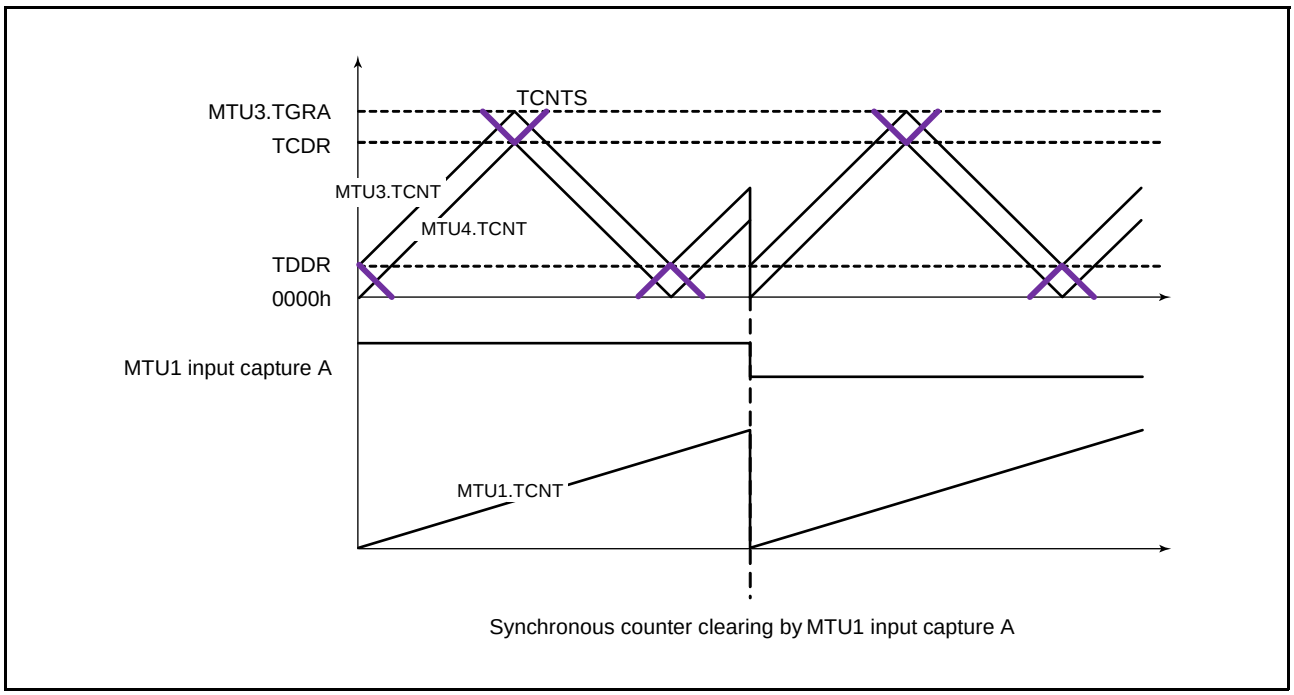


Figure 22.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the TWCR.WRE bit 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 setting TWCR.WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 22.56. When synchronous clearing occurs outside that interval, the initial value specified by the TOCR1.OLSN bit and TOCR1.OLSP bit is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 22.56) immediately after the counters start operation, initial value output is not suppressed.

Synchronous clearing generated in MTU0 to MTU2 can cause counter clearing in the MTU.

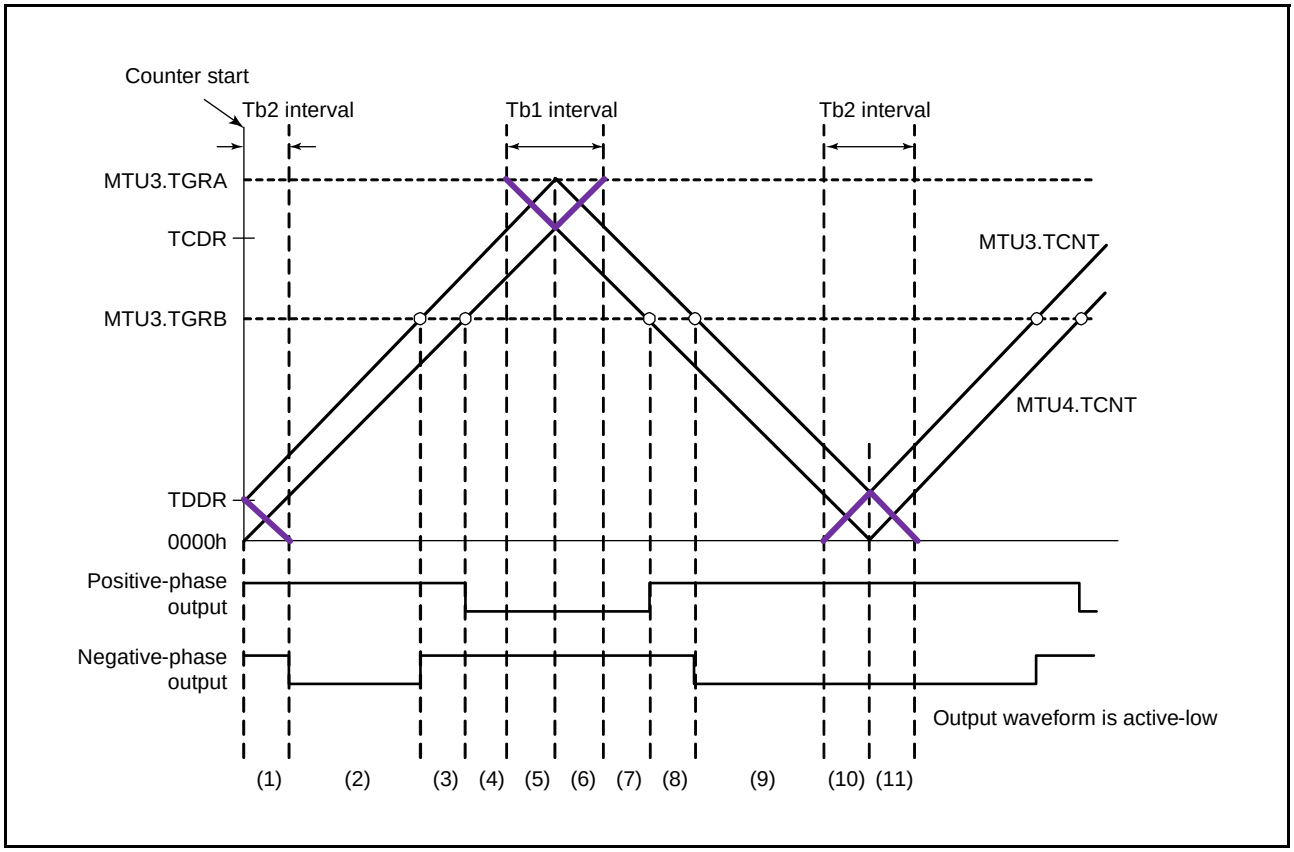


Figure 22.56 Timing for Synchronous Counter Clearing

- 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 22.57.

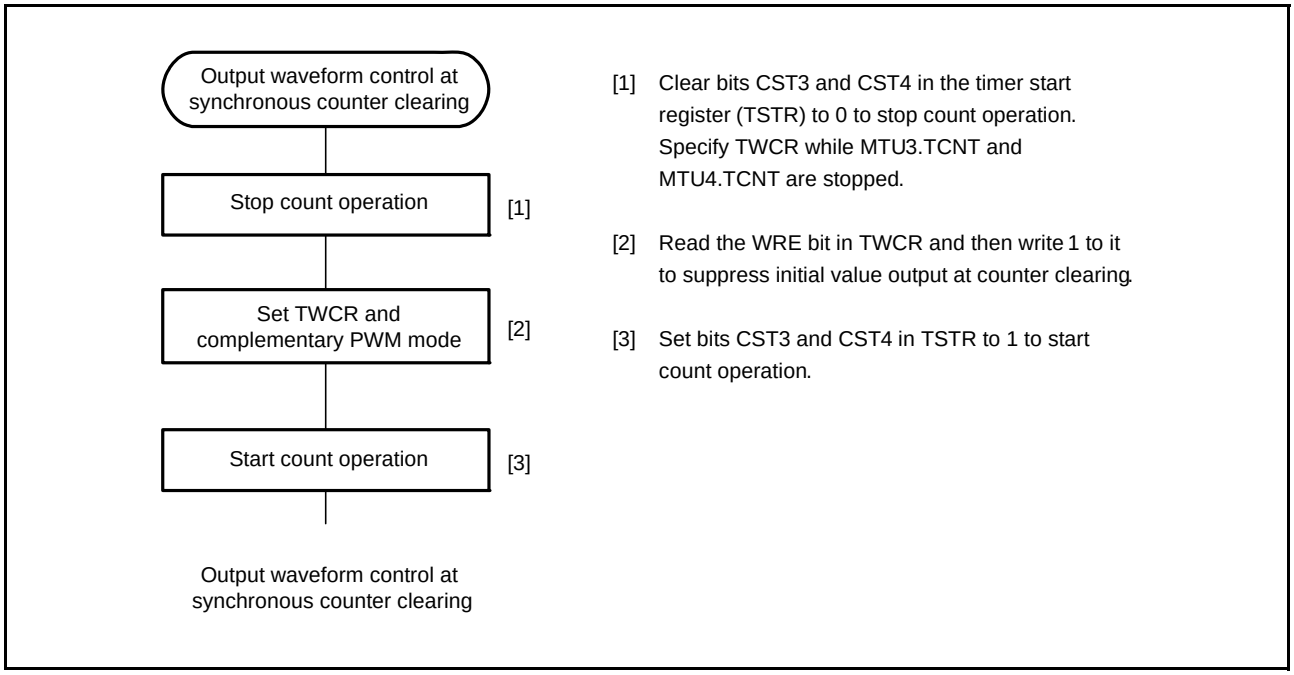


Figure 22.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 22.58 to Figure 22.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the TWCR.WRE bit is set to 1. In the examples shown in Figure 22.58 to Figure 22.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.56, respectively.

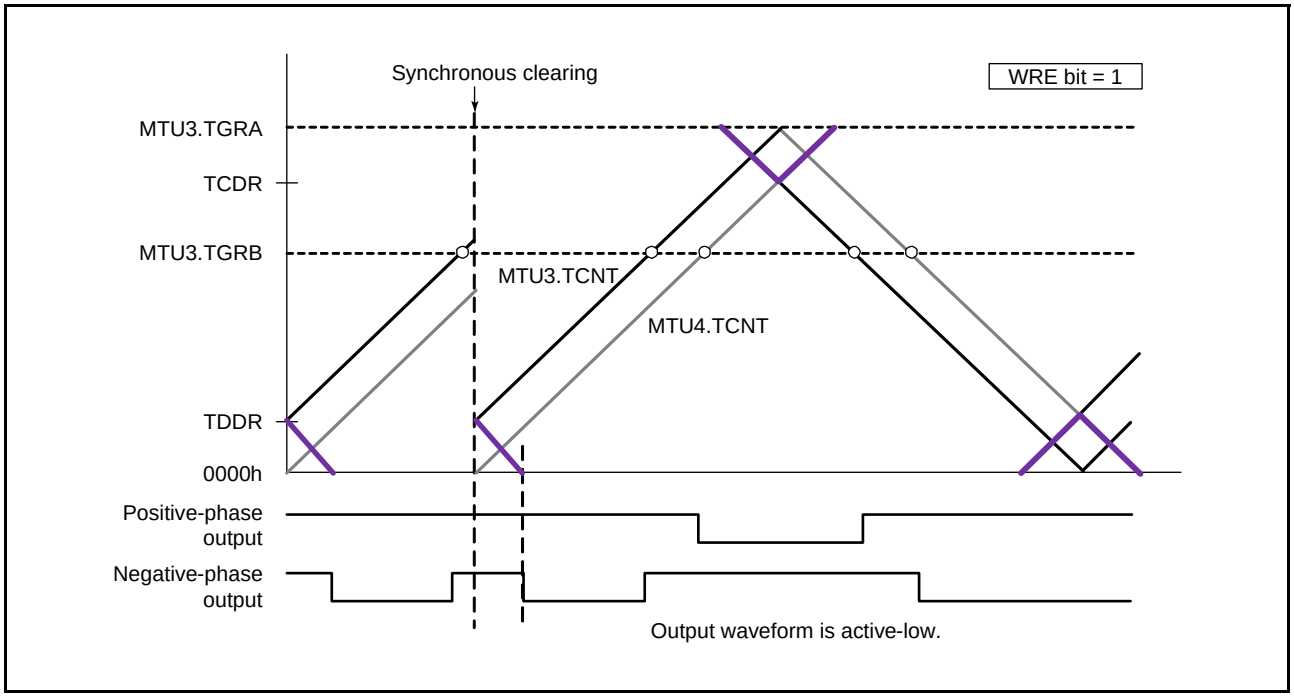


Figure 22.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.56; TWCR.WRE Bit is 1)

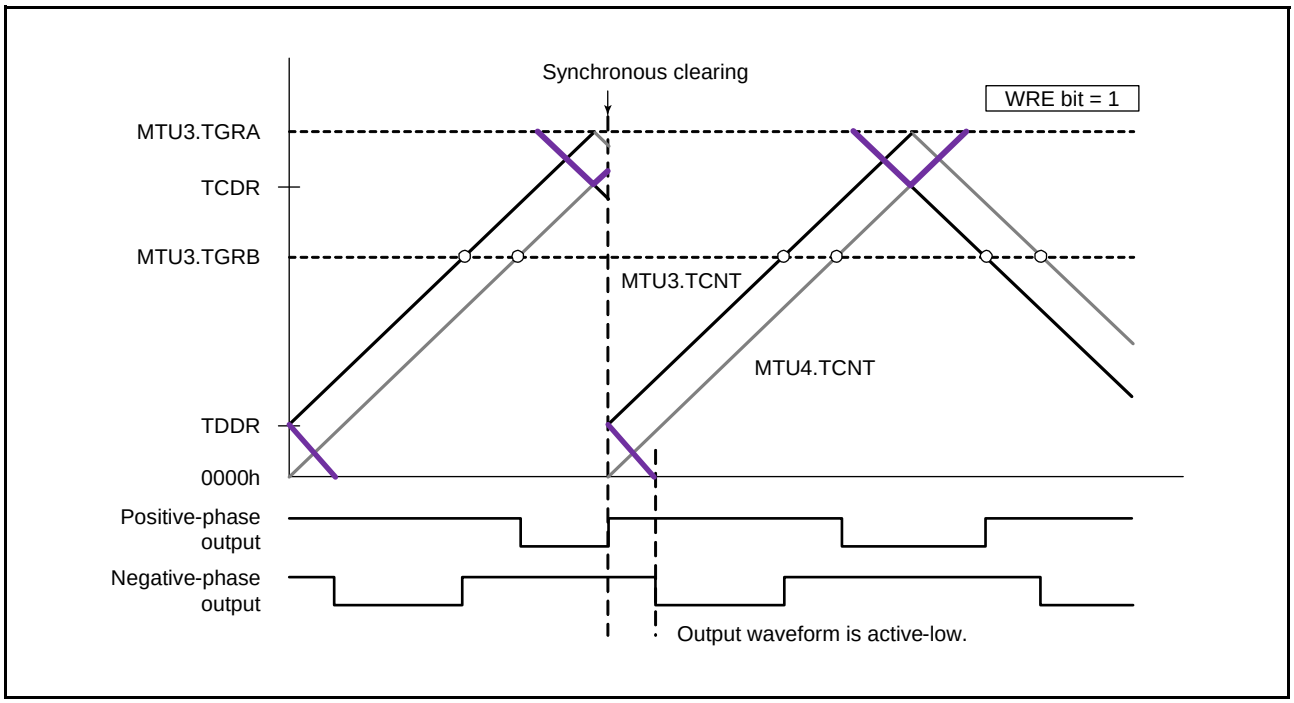


Figure 22.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 22.56; TWCR.WRE Bit is 1)

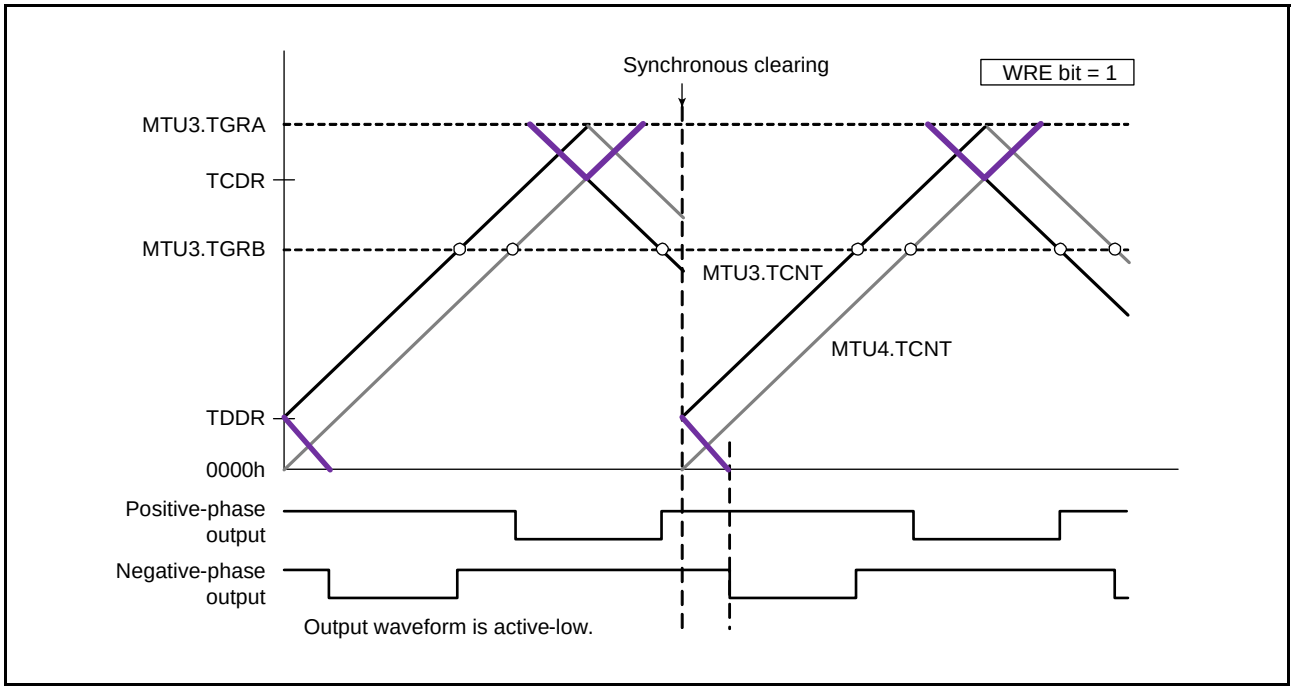


Figure 22.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.56; TWCR.WRE Bit is 1)

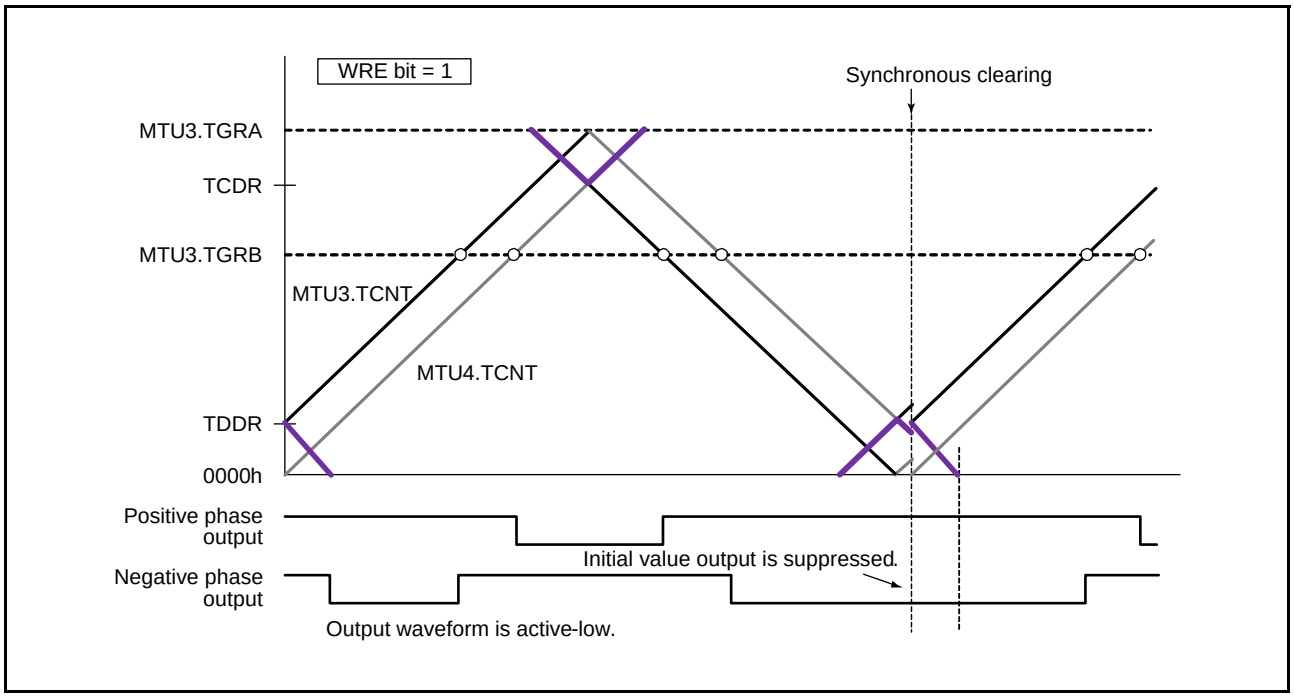


Figure 22.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 22.56; TWCR.WRE Bit is 1)

(o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by MTU3.TGRA compare match when the TWCR.CCE bit is set.
Figure 22.62 shows an operation example.

Note: Use this function only in complementary PWM mode 1 (transfer at crest).
Note: Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1).
Note: Do not set the PWM duty cycle value to 0000h.
Note: Do not set the TOCR1.PSYE bit to 1.

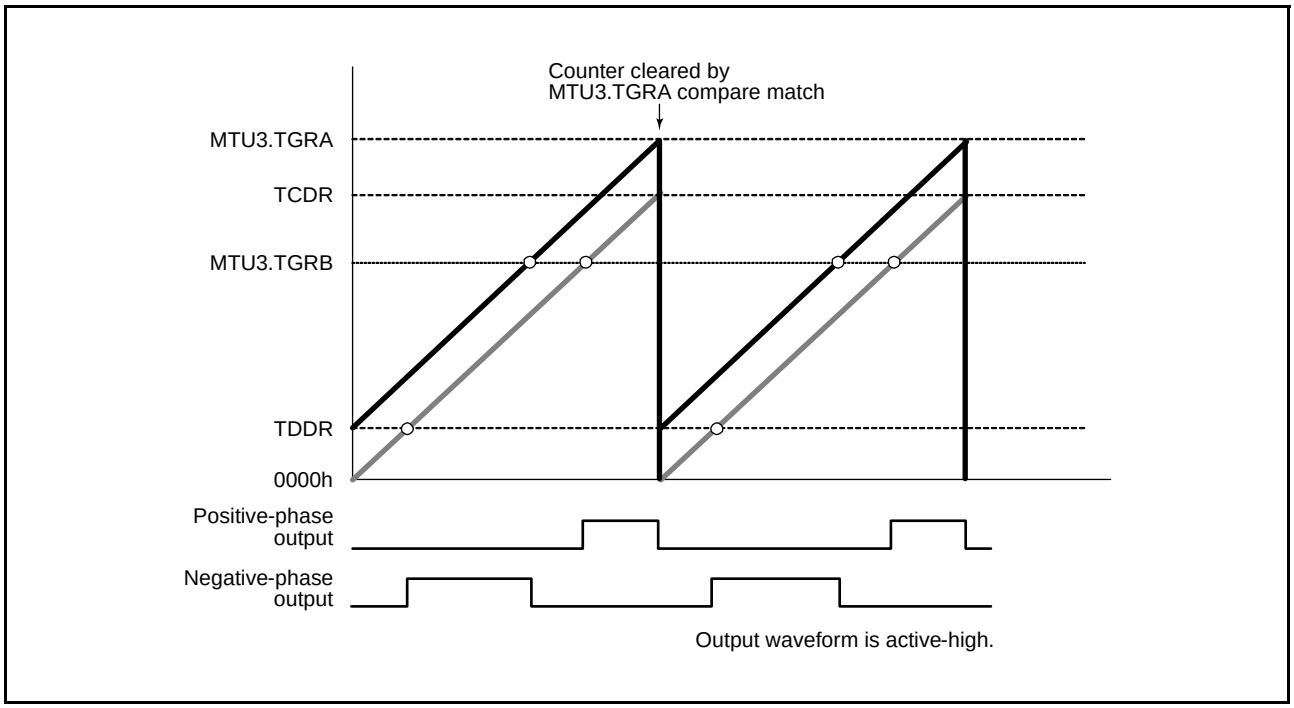


Figure 22.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(p) 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 timer gate control register (TGCR). Figure 22.63 to Figure 22.66 show examples of brushless DC motor driving waveforms created using TGCR.
To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the TGCR.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 in MTU0. When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.
When the TGCR.FB bit is 1, the output on/off state is switched when the TGCR.UF bit, TGCR.VF bit, or TGCR.WF bit is cleared to 0 or set to 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 TGCR.N bit or TGCR.P bit to 1. When the TGCR.N bit or TGCR.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 TOCR1.OLSN bit and TOCR1.OLSP bit regardless of the setting of the TGCR.N bit and TGCR.P bit.

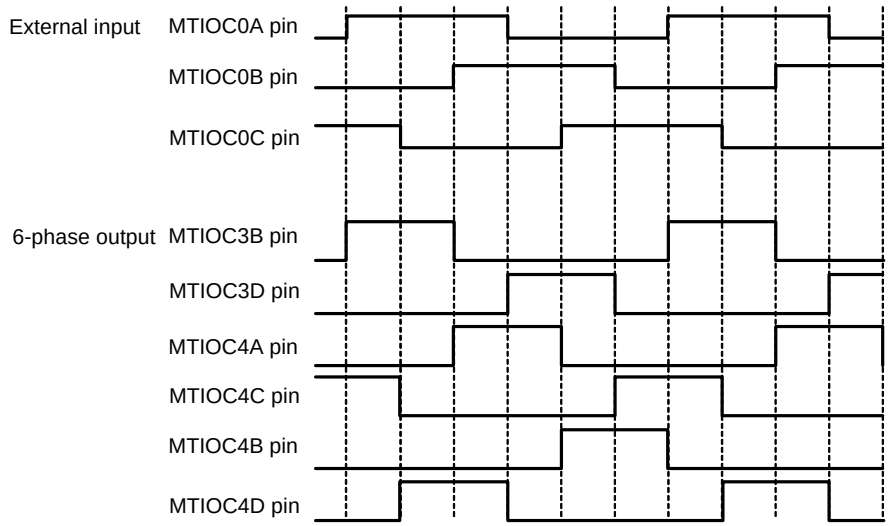


Figure 22.63 Example of Output Phase Switching by External Input (1)

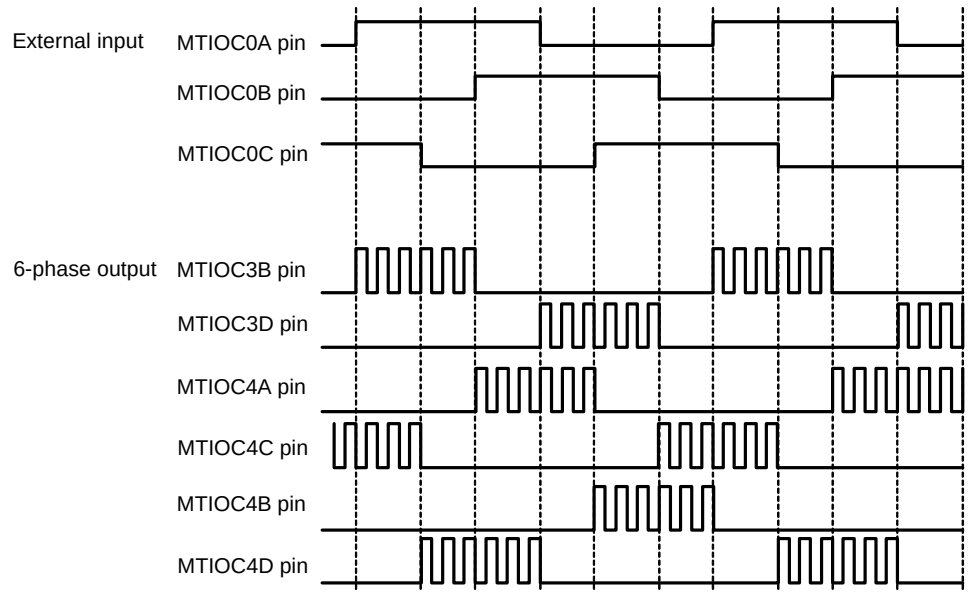


Figure 22.64 Example of Output Phase Switching by External Input (2)

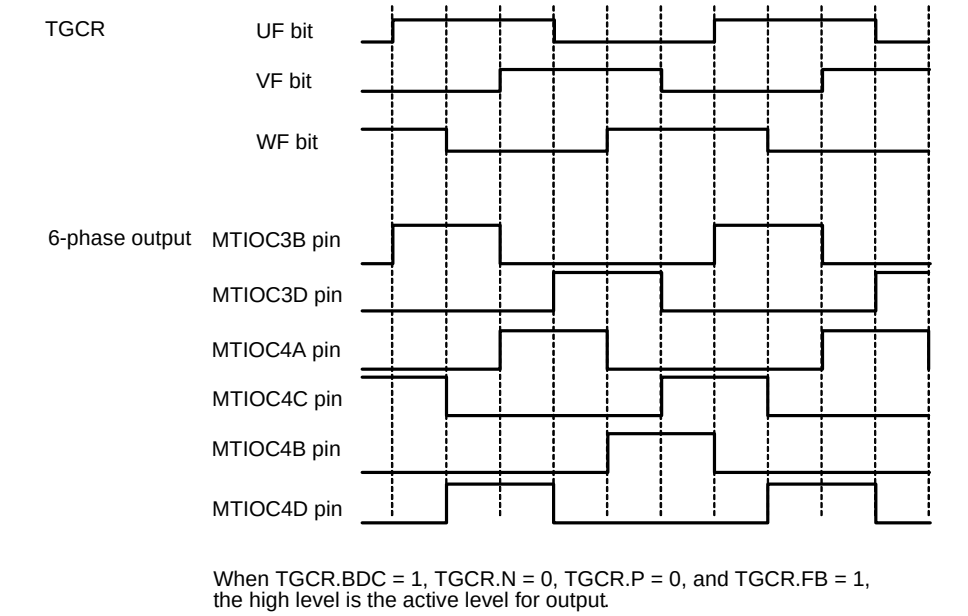


Figure 22.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

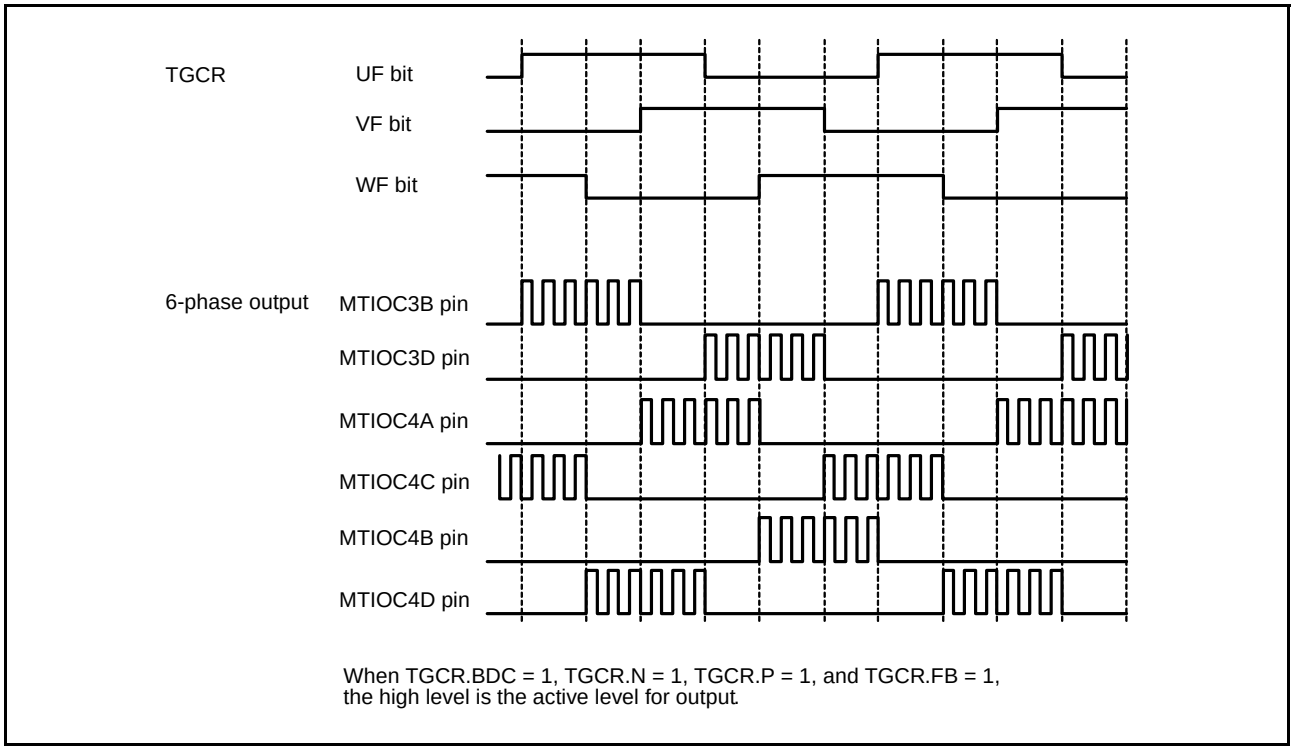


Figure 22.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA compare match, MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.

When start requests using MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.

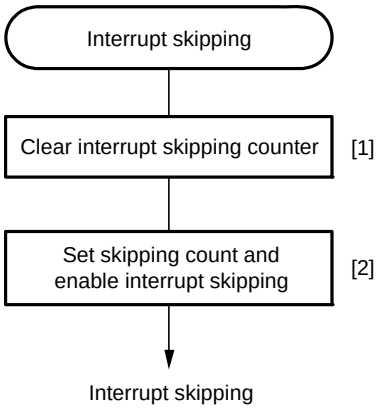
A/D converter start requests can be specified by setting the TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2 bit to 1.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA3 (at the crest) and TCIV4 (at the trough) in MTU3 and MTU4 can be skipped up to seven times by setting the timer interrupt skipping set register (TITCR).
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 timer buffer transfer set register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.
A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 22.3.9, A/D Converter Start Request Delaying Function.
The timer interrupt skipping set register (TITCR) should be set while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which compare match never occur and TGIA3 and TGIA4 interrupt requests by compare match are never generated. Before changing the skipping count, be sure to clear the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 22.67 shows an example of the interrupt skipping operation setting procedure. Figure 22.68 shows the periods during which interrupt skipping count can be changed.



[1] Set bits T3AEN and T4VEN in the timer interrupt skipping set register (TITCR) to 0 to clear the skipping counter.

[2] Specify the interrupt skipping count within the range from 0 to 7 times in bits T3ACOR2 to T3ACOR0 and T4VCOR2 to T4VCOR0 in TITCR, and enable interrupt skipping through bits T3AEN and T4VEN.

Note: The setting of TITCR must be done while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of registers MTU3.TIER and MTU4.TIER under the condition in which compare match never occur.
Before changing the skipping count, be sure to set the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter.

Figure 22.67 Example of Interrupt Skipping Operation Setting Procedure

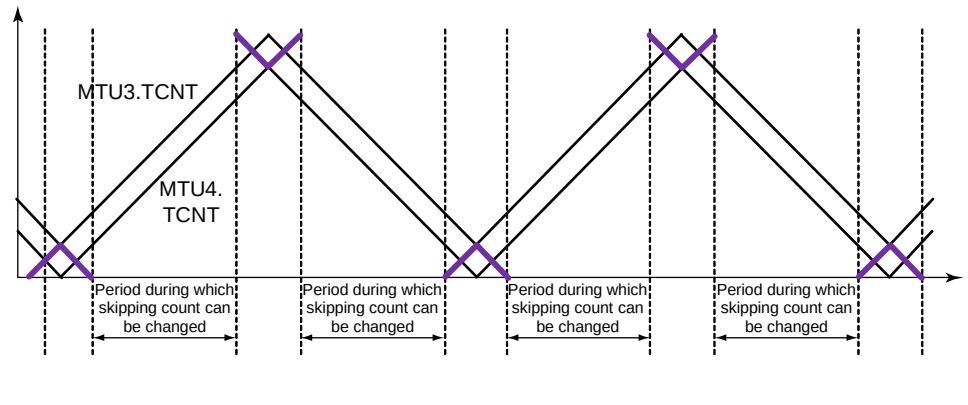


Figure 22.68 Periods during which Interrupt Skipping Count Can be Changed

(b) Example of Interrupt Skipping Operation

Figure 22.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the TITCR.T3ACOR bit and the TITCR.T3AEN bit is set to 1.

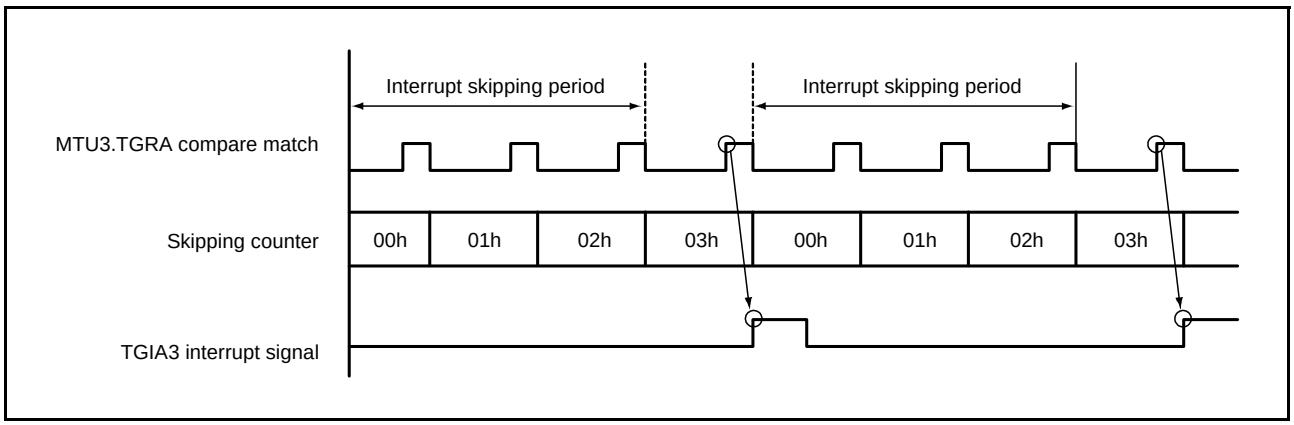


Figure 22.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

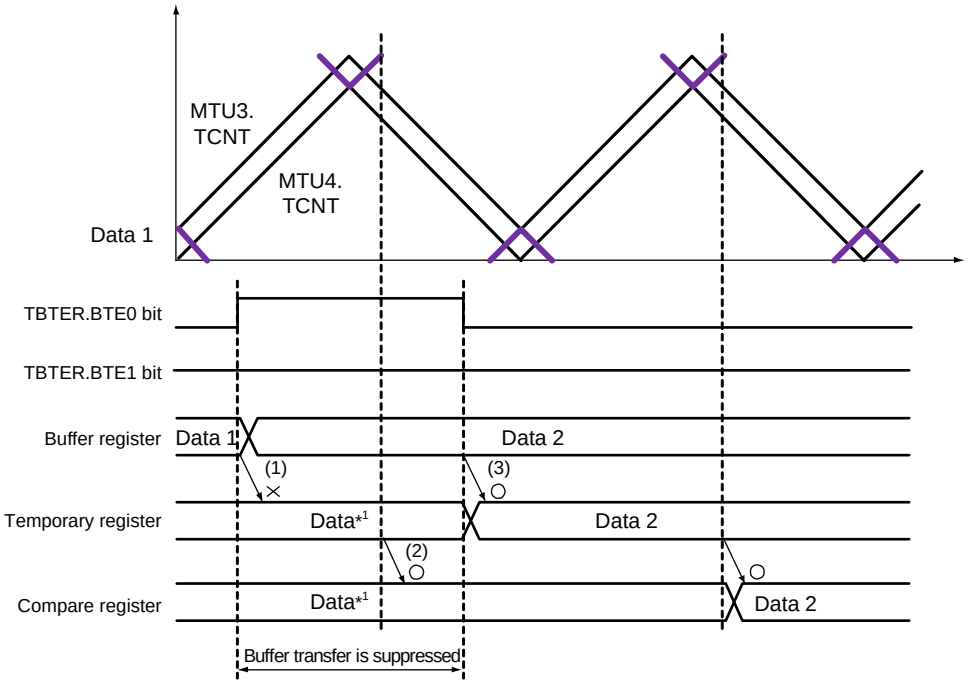
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the timer buffer transfer set register (TBTER).

Figure 22.70 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 22.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, if data is written to the buffer register within the buffer transfer-enabled period, the data is transferred immediately from the buffer register to the temporary register. If data is written to the buffer register outside the buffer transfer-enabled period, the data is transferred from the buffer register to the temporary register at the timing when the next buffer transfer-enabled period starts.

Note that the buffer transfer-enabled period depends on the TITCR.T3AEN bit and TITCR.T4VEN bit settings. Figure 22.72 shows the relationship between the TITCR.T3AEN bit and TITCR.T4VEN bit settings and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the TBTER.BTE1 bit 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 (setting TBTER.BTE[1:0] to 01b).
(2) Data is transferred from the temporary register to the compare register even in the compare 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 22.70 Example of Operation When Buffer Transfer is Disabled (TBTER.BTE[1:0] = 01b)

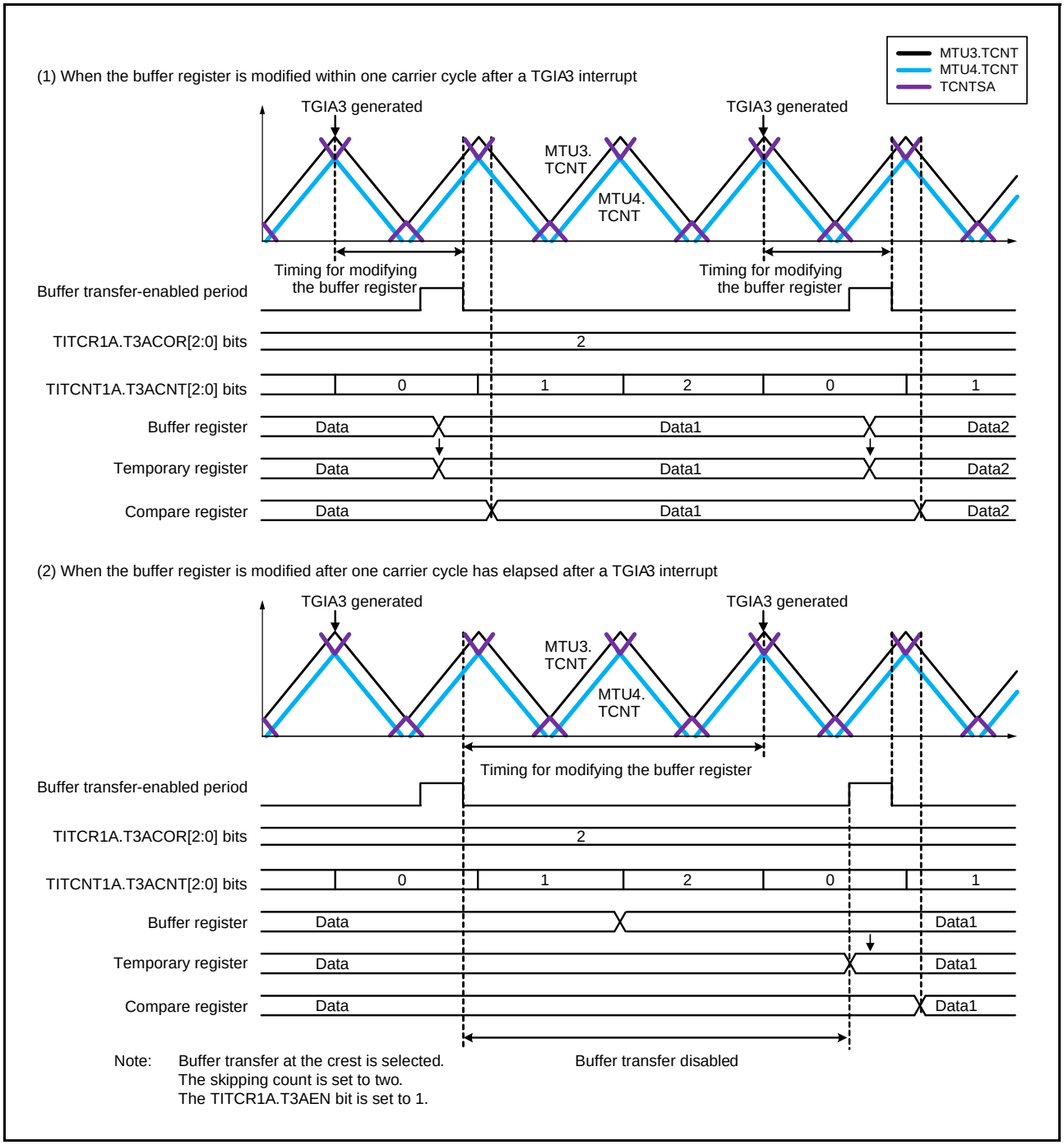


Figure 22.71 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (TBTER.BTE[1:0] = 10b)

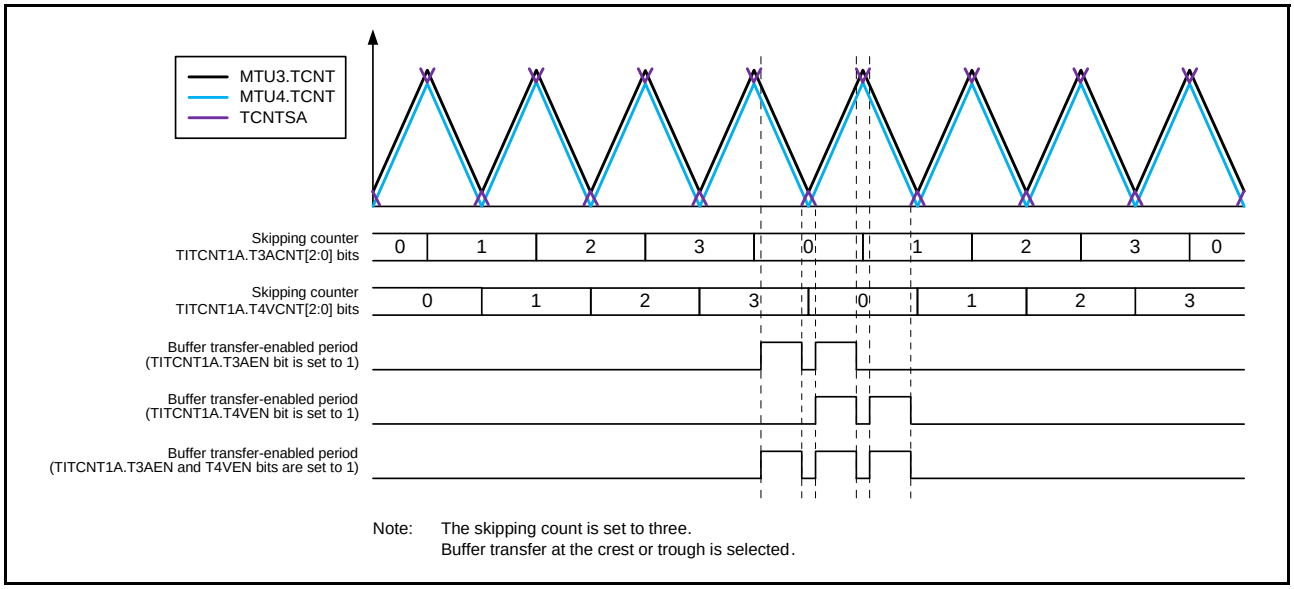


Figure 22.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The MTU provides the following protection functions for complementary PWM mode output.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers can be enabled or disabled by setting the TRWER.RWE bit. The applicable registers are some of the registers in MTU3 and MTU4 shown below:

22 registers in total

MTU3.TCR and MTU4.TCR, MTU3.TMDR and MTU4.TMDR, MTU3.TIORH and MTU4.TIORH, MTU3.TIORL and MTU4.TIORL, MTU3.TIER and MTU4.TIER, MTU3.TCNT and MTU4.TCNT, MTU3.TGRA and MTU4.TGRA, MTU3.TGRB and MTU4.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, and TDDR

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

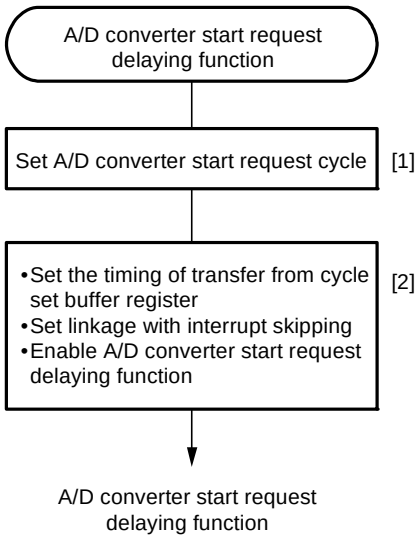
The PWM output pins of MTU0, MTU3, and MTU4 can be set to the high-impedance state automatically.
Refer to section 23, Port Output Enable 2 (POE2a), for details.

22.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB).
The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).
A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and ITB4VE bit.

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 22.73 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle set buffer registers A and B (MTU4.TADCOBRA/TADCOBRB) and timer A/D converter start request cycles set registers A and B (MTU4.TADCORA/TADCORB). (The same initial value must be specified in the cycle set buffer register and cycle set register.)

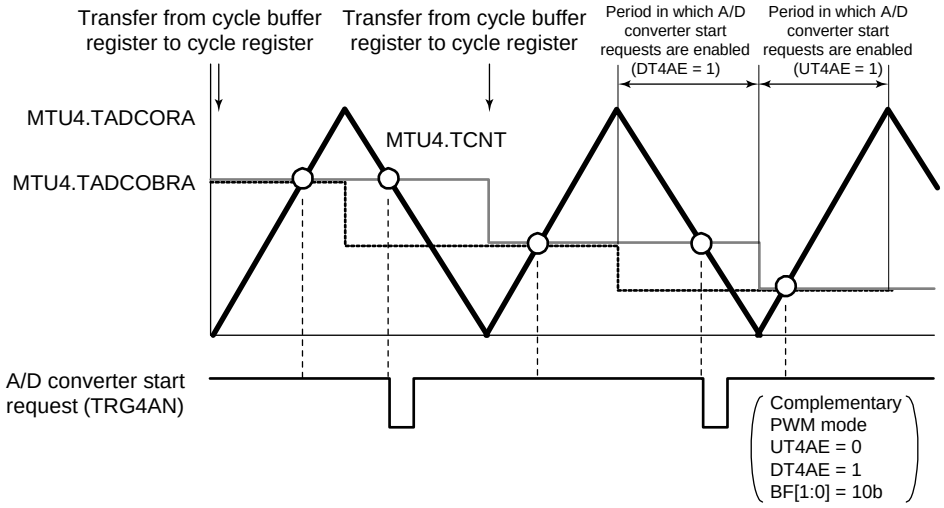
[2] Use the BF[1:0] bits in the timer A/D converter start request control register (TADCR) to specify the timing of buffer transfer from the timer A/D converter start request cycle set buffer register to A/D converter start request cycle set register.
- Specify whether to link with interrupt skipping through TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit.
- Use TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, and TADCR.DT4BE bit to enable A/D converter start requests (TRG4AN or TRG4BN).

Note: Perform TADCR setting while MTU4.TCNT is stopped.
Set the TADCR.BF1, ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, and DT4BE bits to 0 when complementary PWM mode is not selected.
Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits to 0 when interrupt skipping is disabled.

Figure 22.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 22.74 shows a basic example of A/D converter start request signal (TRG4AN) operation when the trough of MTU4.TCNT is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT down-counting.



Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR - 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 22.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Converter Start Requests are Enabled

When the MTU4.TCNT counter and the MTU4.TADCORA or MTU4.TADCORB register match within the period enabled by the UT4AE and UT4BE bits, the corresponding A/D converter start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1 in complementary PWM mode, A/D converter start requests are enabled during the MTU4.TCNT up-counting (0 ≤ MTU4.TCNT ≤ TCDR - 1). When the DT4AE and DT4BE bits in the MTU4.TADCR register are set to 1, A/D converter start requests are enabled during MTU4.TCNT down-counting (TCDR ≥ MTU4.TCNT ≥ 1). See Figure 22.74.

(4) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the MTU4.TADCR.BF[1:0] bits.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode. For details, section 22.6.27, Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR register to 0.

(5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

In complementary PWM mode, A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit. Figure 22.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

Figure 22.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D converter start requests are linked with interrupt skipping.

In modes other than complementary PWM mode, do not use the A/D converter start request delaying function linked with interrupt skipping.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register to 0.

Note: This function should be used in combination with interrupt skipping.
When interrupt skipping is disabled (the TITCR.T3AEN bit and TITCR.T4VEN bit are cleared to 0 or the skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit to 0).
Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.
When this function is used, MTU4.TADCORA and MTU4.TADCORB should be set with the value ranging 0002h to the value set in TCDRA minus 2.

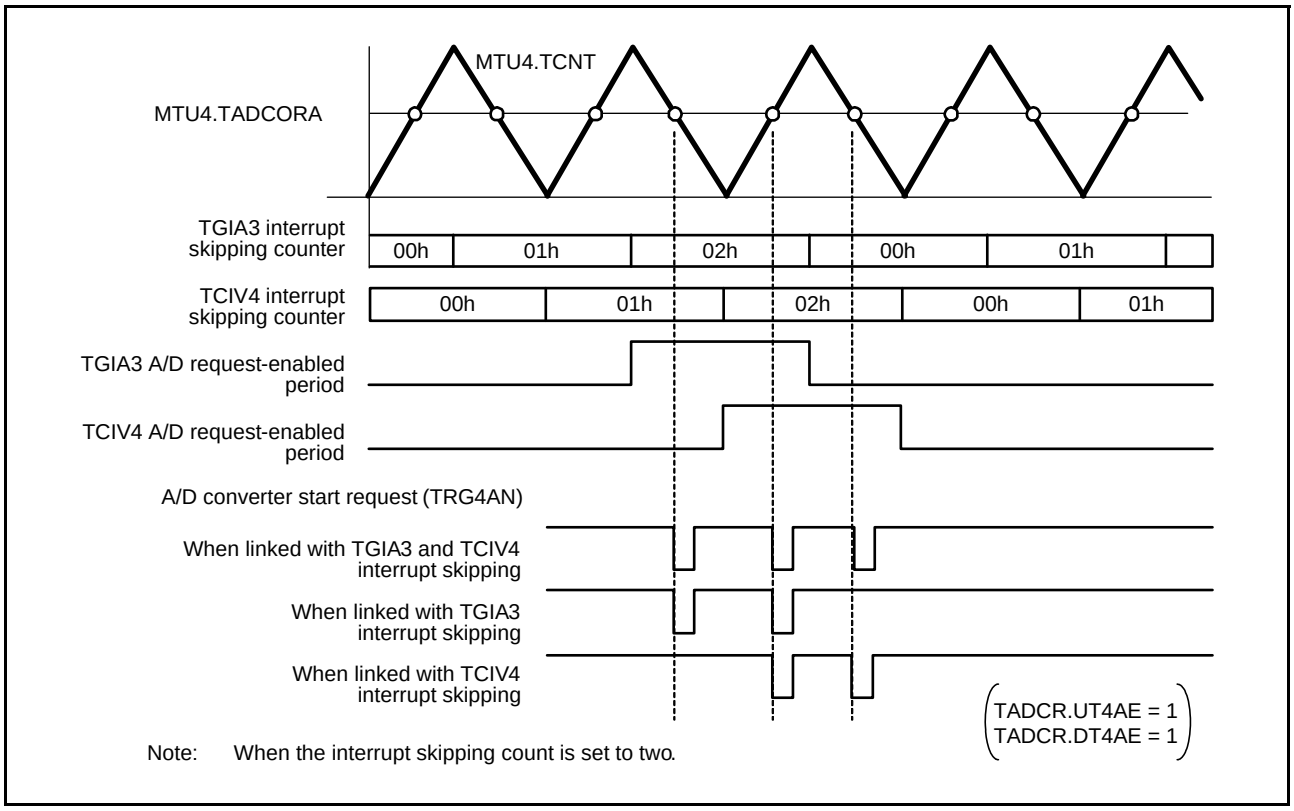


Figure 22.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)

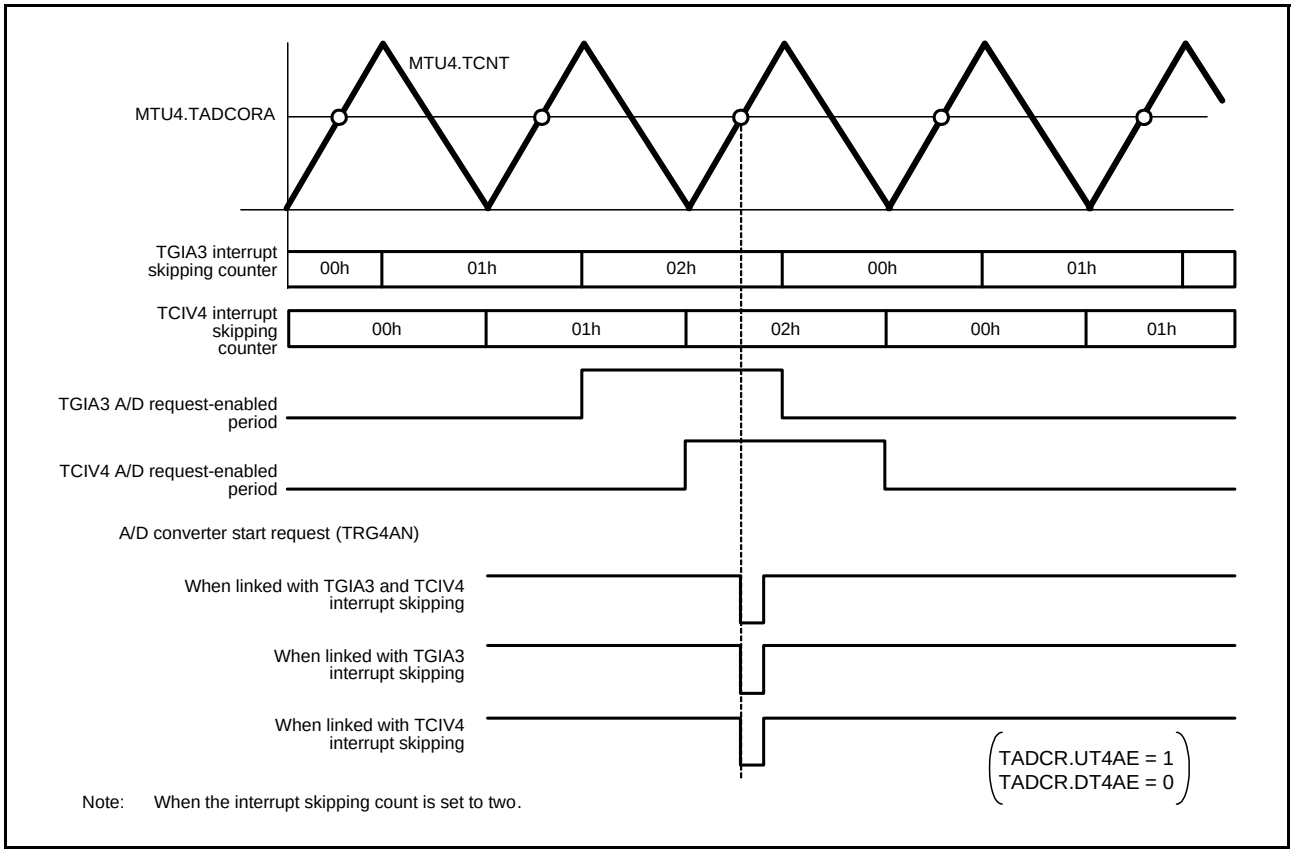


Figure 22.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)

22.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.
When the IOC[4:0] bits in MTU5.TIORU, TIORV, and TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins is measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.
Figure 22.77 shows an example of setting external pulse width measurement, and Figure 22.78 an example of external pulse width measurement.

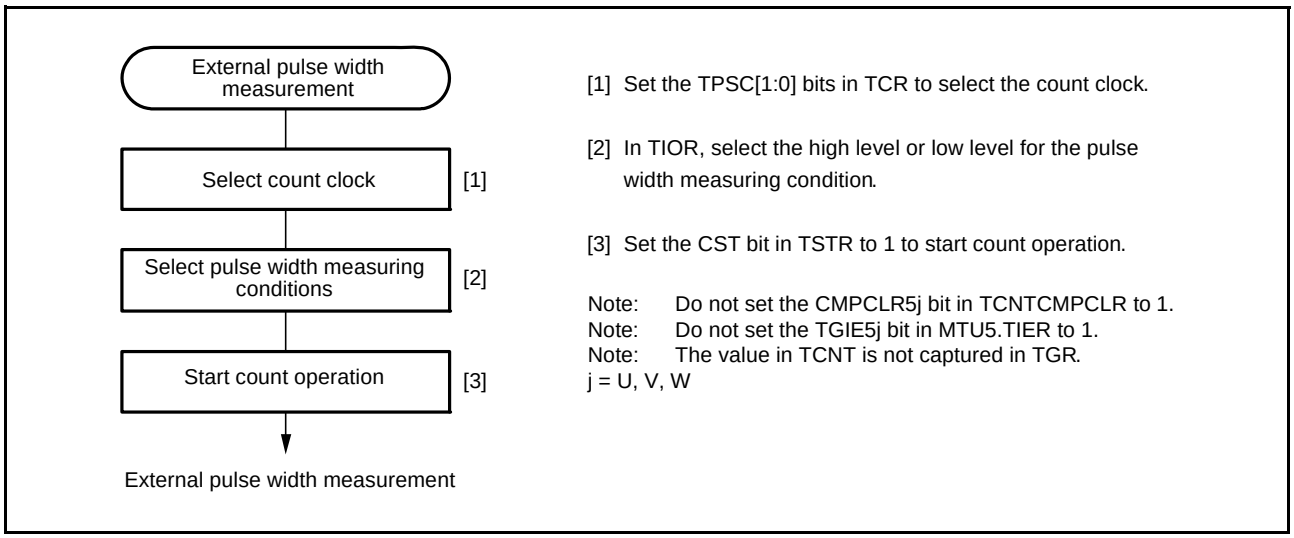


Figure 22.77 Example of External Pulse Width Measurement Setting Procedure

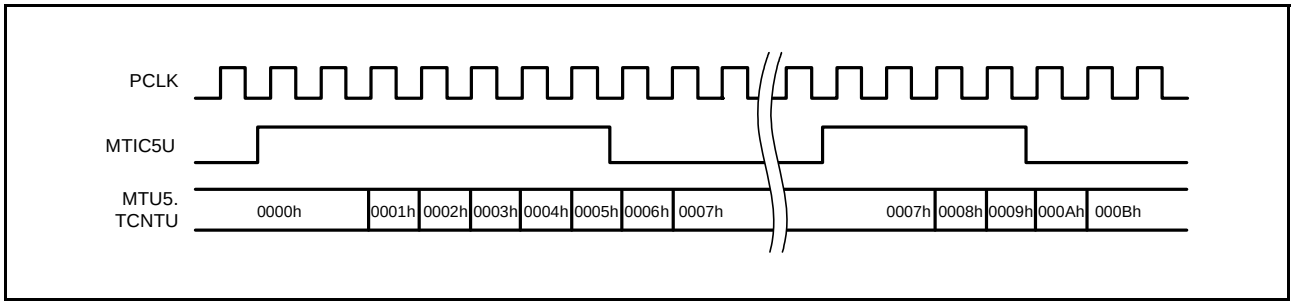


Figure 22.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

22.3.11 Dead Time Compensation

The motor control circuit is configured so that a delay in the dead time (delay between complementary PWM output and inverter output) is fed back to MTU5 (Figure 22.79). The MTU5 external pulse measurement function allows the delay in the dead time to be measured and reflected in the duty ratio, which can be used as dead time compensation for the PWM output waveform in complementary PWM operation when MTU and MTU7 are used (Figure 22.80). Figure 22.81 shows the procedure for setting dead time compensation using MTU5. For details on MTU5 operation at this time, refer to section (2), TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode Operation.

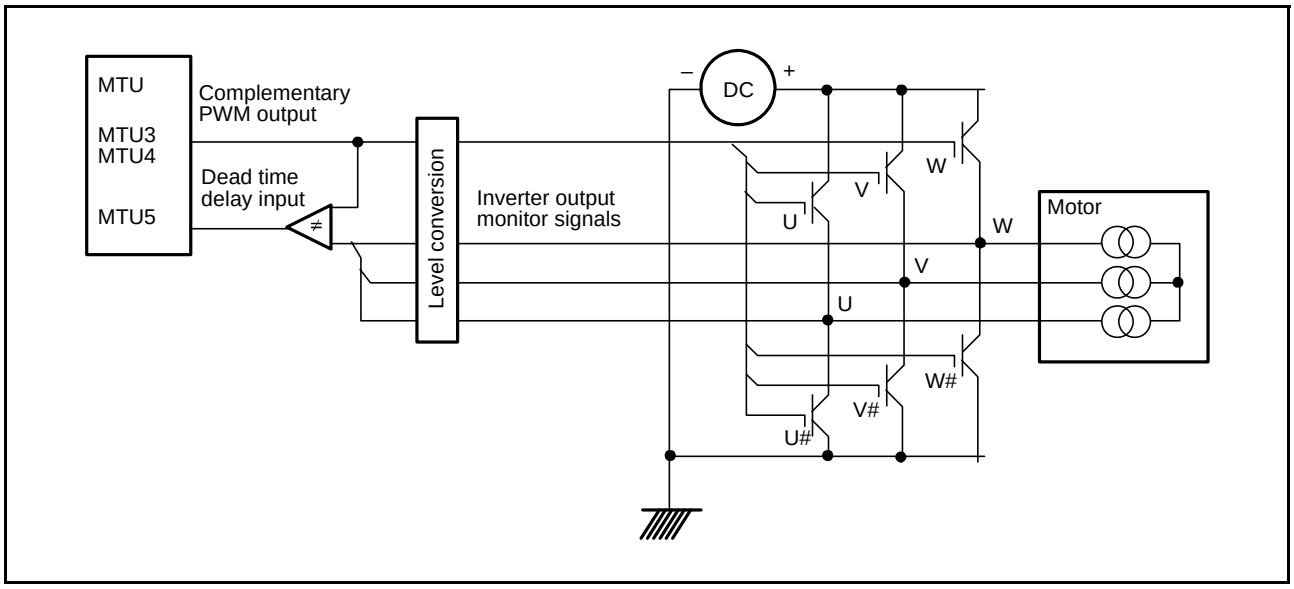


Figure 22.79 Example of Motor Control Circuit Configuration

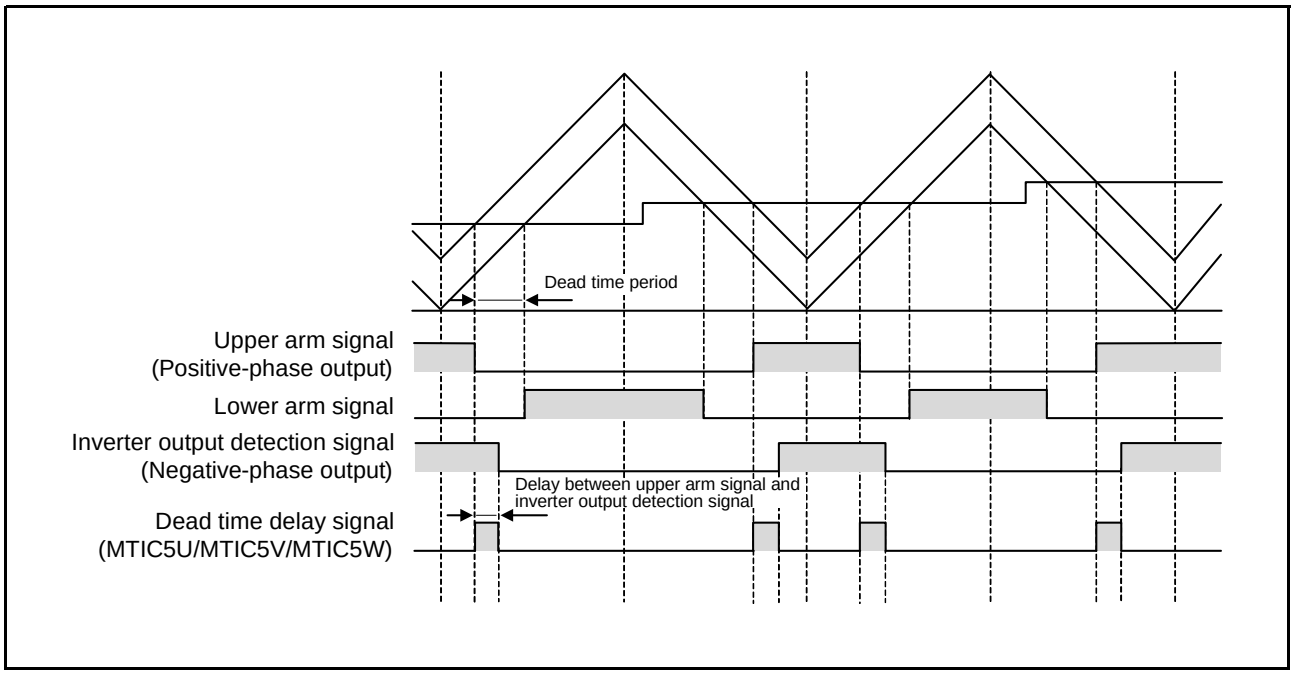
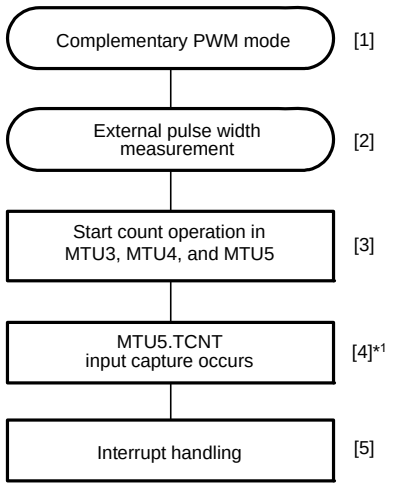


Figure 22.80 Delay in Dead Time in Complementary PWM Mode Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 22.81 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Place MTU3 and MTU4 in complementary PWM mode. For details, refer to the Complementary PWM Mode section.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to the External Pulse Width Measurement section.

[3] Set bits CST3 and CST4 in MTU.TSTR 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.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA3) or trough (TCIV4) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU5.TGRU and MTU3.TGRB, and write the corrected value to MTU3.TGRD in the interrupt handling.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU4.TGRC and MTU4.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 1. As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 22.81 Example of Dead Time Compensation Setting Procedure

(2) TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode Operation

The MTU5 external pulse width measurement function can be used to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, trough, or crest and trough during complementary PWM mode operation. The transfer timing should be set in TIORU, TIORV, and TIORW. When the TCNTCNPCLR.CMPCLR5U, CMPCLR5V, and CMPCLR5W bits are set to 1, TCNTU, TCNTV, and TCNTW are cleared to 0 at the transfer timing for TGRU, TGRV, and TGRW.

Figure 22.82 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 or though in complementary PWM mode.

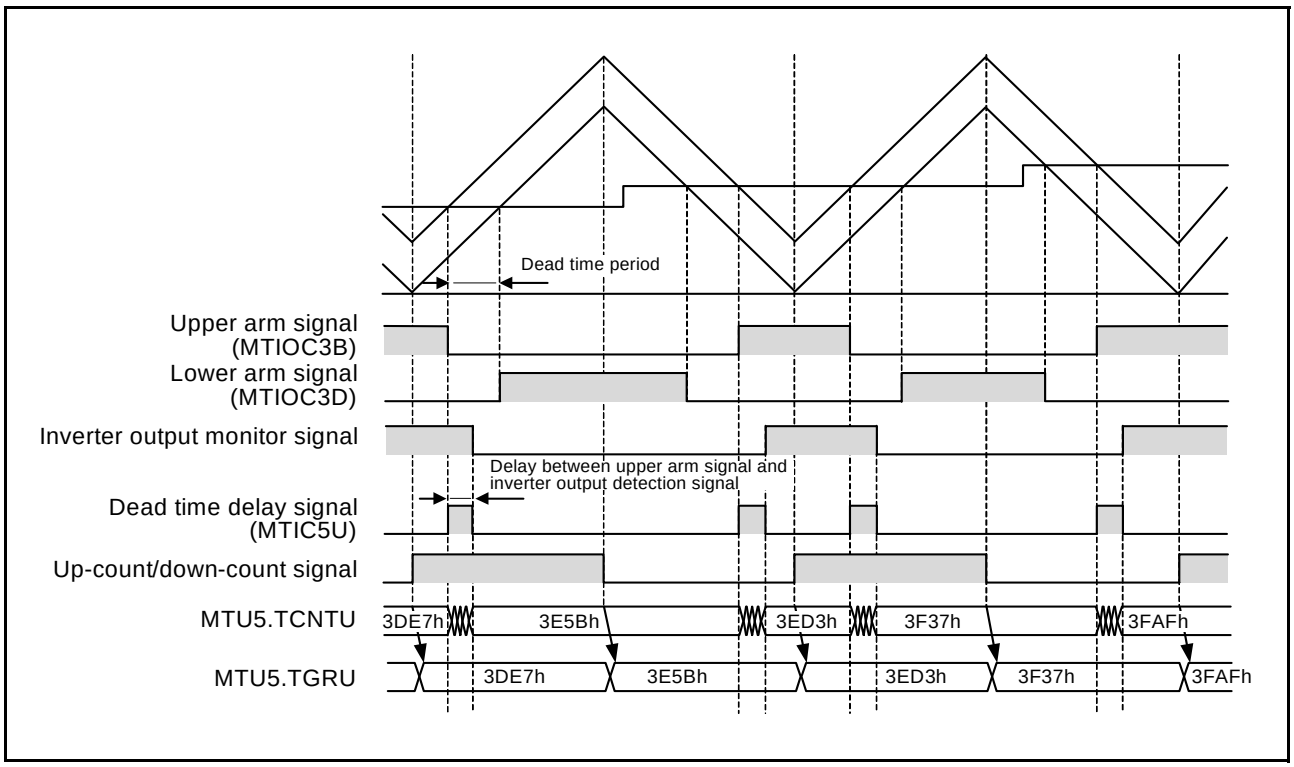


Figure 22.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

22.3.12 Noise Filter

Each pin for use in input capture and external pulse input to the MTU 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 cycles. 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 22.83 shows the timing of noise filtering.

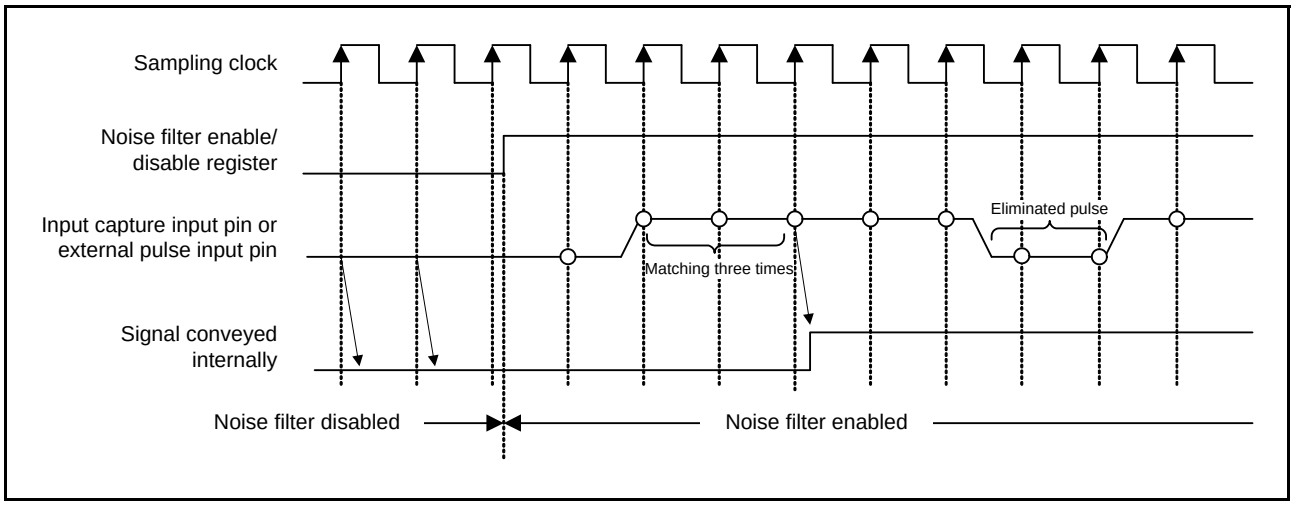


Figure 22.83 Timing of Noise Filtering

22.4 Interrupt Sources

22.4.1 Interrupt Sources and Priorities

There are three interrupt sources; 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 detected, an interrupt is requested if the corresponding enable/disable bit in TIER is set to 1. 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 (ICUb).

Table 22.57 lists the MTU interrupt sources.

Table 22.57 MTU Interrupt Sources (1)

Channel	Name	Interrupt Source	DMAC Activation	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible	High
	TGIB0	MTU0.TGRB input capture/compare match	Not possible	Possible	
	TGIC0	MTU0.TGRC input capture/compare match	Not possible	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Not possible	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Not possible	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Not possible	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Not possible	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Not possible	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Not possible	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Not possible	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Not possible	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Not possible	Possible	
	TCIV4	MTU4.TCNT overflow/underflow	Not possible	Possible	
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Not possible	Possible	
	TGIV5	MTU5.TGRV input capture/compare match	Not possible	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	Not possible	Possible	Low

Note: This table lists the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel. The MTU has 21 input capture/compare match interrupts (six for MTU0, four each for MTU3 and MTU4, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TCNT overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel).

(3) Underflow Interrupt

An interrupt is requested if the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

22.4.2 DTC/DMAC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, refer to section 18, Data Transfer Controller (DTCa).
The MTU provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, five for MTU4, and three for MTU5.

(2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, refer to section 17, DMA Controller (DMACA).
The MTU provides a total of five input capture/compare match interrupts that can be used as DMAC activation sources: one each for MTU0 to MTU4.
When the DMAC is activated by the MTU, the activation source is cleared when the DMAC requests the internal bus mastership. Therefore, there may be a wait period before DMAC transfer starts even when the activation source is cleared, depending on the internal bus state.

22.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 22.58 lists the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at MTU4.TCNT Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM mode operation is performed while the MTU4.TIER.TTGE2 bit is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT count (MTU4.TCNT = 0000h).
A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the MTU4.TIER.TTGE2 bit is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE

A compare match between MTU0.TCNT and MTU0.TGRE activates the A/D converter.

A/D converter start request signal TRG0EN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRE. If A/D converter start signal TRG0EN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRF

An input capture or compare match between MTU0.TCNT, MTU0.TGRA, and MTU0.TGRB activates the A/D converter. A compare match between MTU0.TCNT and MTU0.TGRF activates the A/D converter.
A/D converter start request signal TRG0FN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRF. If A/D converter start signal TRG0FN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or TGRB

The A/D converter can be activated when an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB.
When an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB. A/D converter start request signal TRG0AN or TRG0BN is issued. If A/D converter start signal TRG0AN or TRG0BN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(5) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the TADCORA or TADCORB value if the TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, or TADCR.DT4BE bit is set to 1. For details, refer to section 22.3.9, A/D Converter Start Request Delaying Function.
A/D conversion will start if A/D converter start signal TRG4ABN from the MTU is selected as the trigger in the A/D converter when TRG4AN or TRG4BN is generated.

Table 22.58 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGAN
MTU1.TGRA and MTU1.TCNT		
MTU2.TGRA and MTU2.TCNT		
MTU3.TGRA and MTU3.TCNT		
MTU4.TGRA and MTU4.TCNT		
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRG0AN
MTU0.TGRB and MTU0.TCNT		TRG0BN
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0EN
MTU0.TGRF and MTU0.TCNT		TRG0FN
TADCORA and MTU4.TCNT		TRG4AN
TADCORB and MTU4.TCNT		TRG4BN
TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT		TRG4ABN

22.5 Operation Timing

22.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 22.84 and Figure 22.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 22.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 22.87 shows the TCNT count timing in external clock operation (phase counting mode).

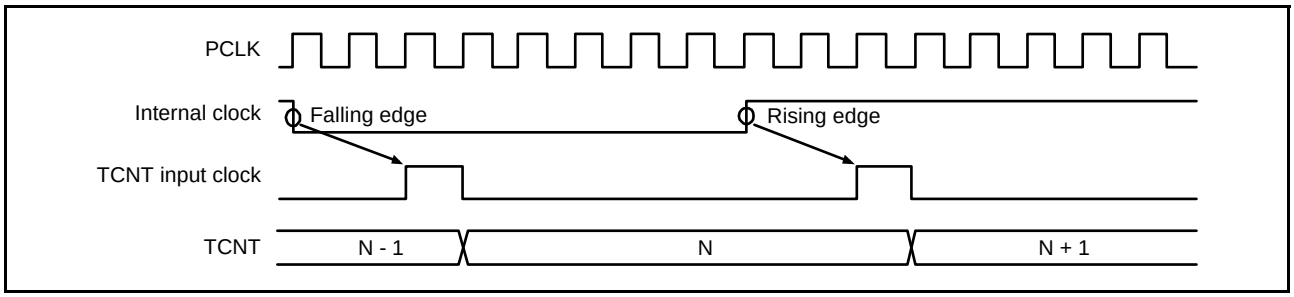


Figure 22.84 Count Timing in Internal Clock Operation (MTU0 to MTU4)

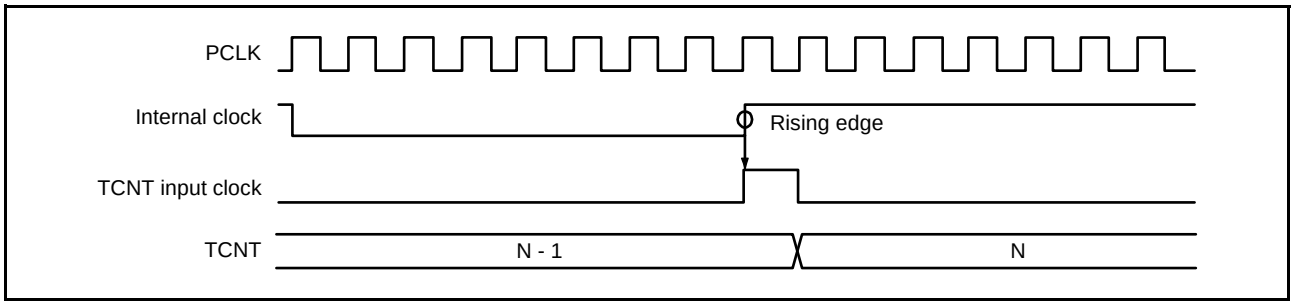


Figure 22.85 Count Timing in Internal Clock Operation (MTU5)

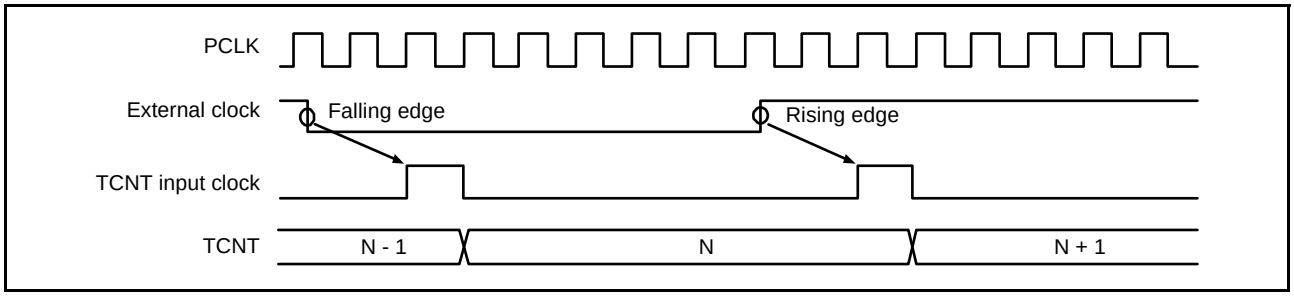


Figure 22.86 Count Timing in External Clock Operation (MTU0 to MTU4)

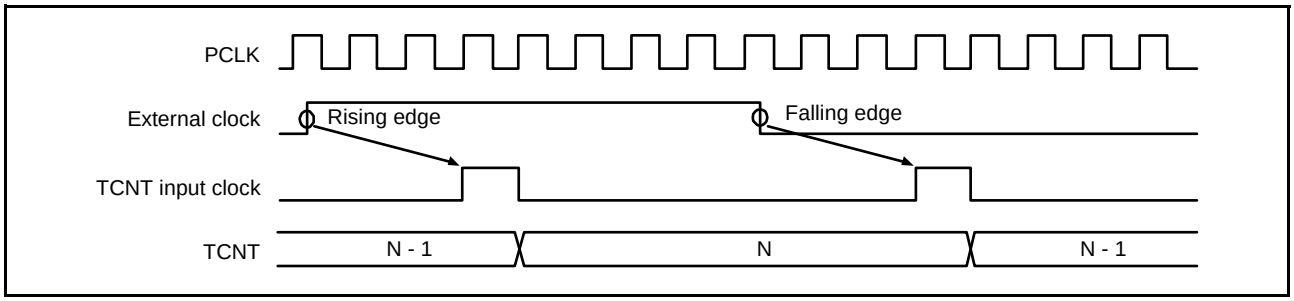


Figure 22.87 Count Timing in External Clock Operation (Phase Counting Mode)

(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 is updated by TCNT). When a compare match signal is generated, the value set in TIOR is output to the output compare output pin (MTIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 22.88 shows the output compare output timing (normal mode or PWM mode) and Figure 22.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

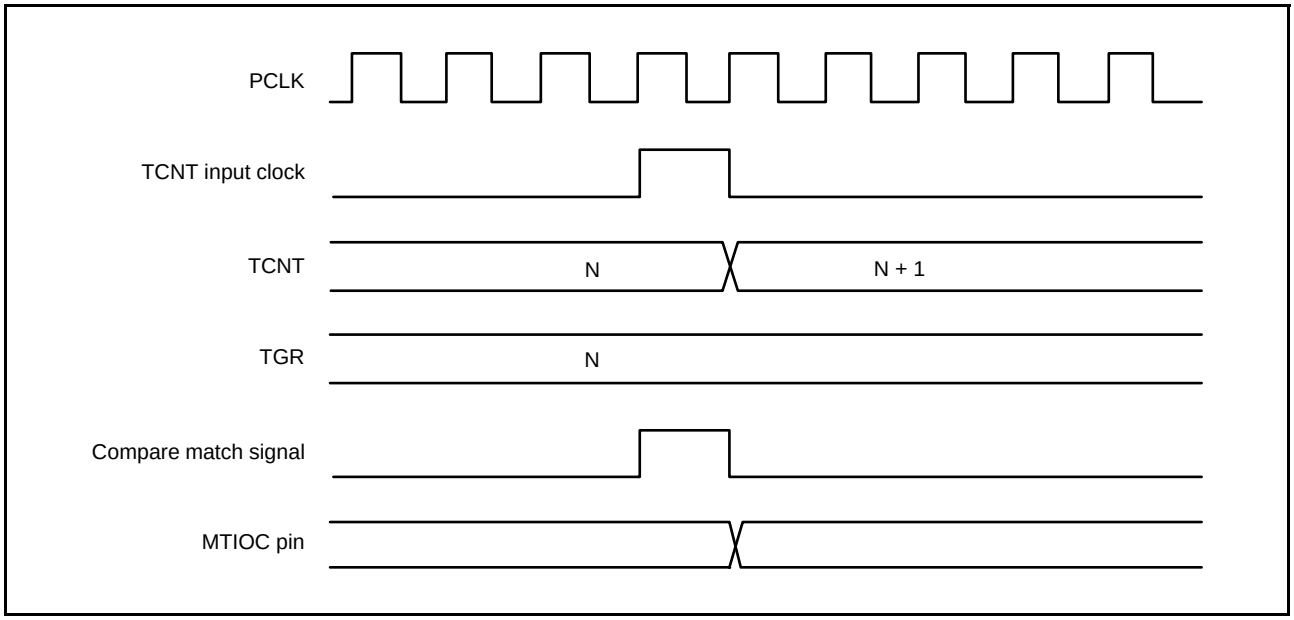


Figure 22.88 Output Compare Output Timing (Normal Mode or PWM Mode)

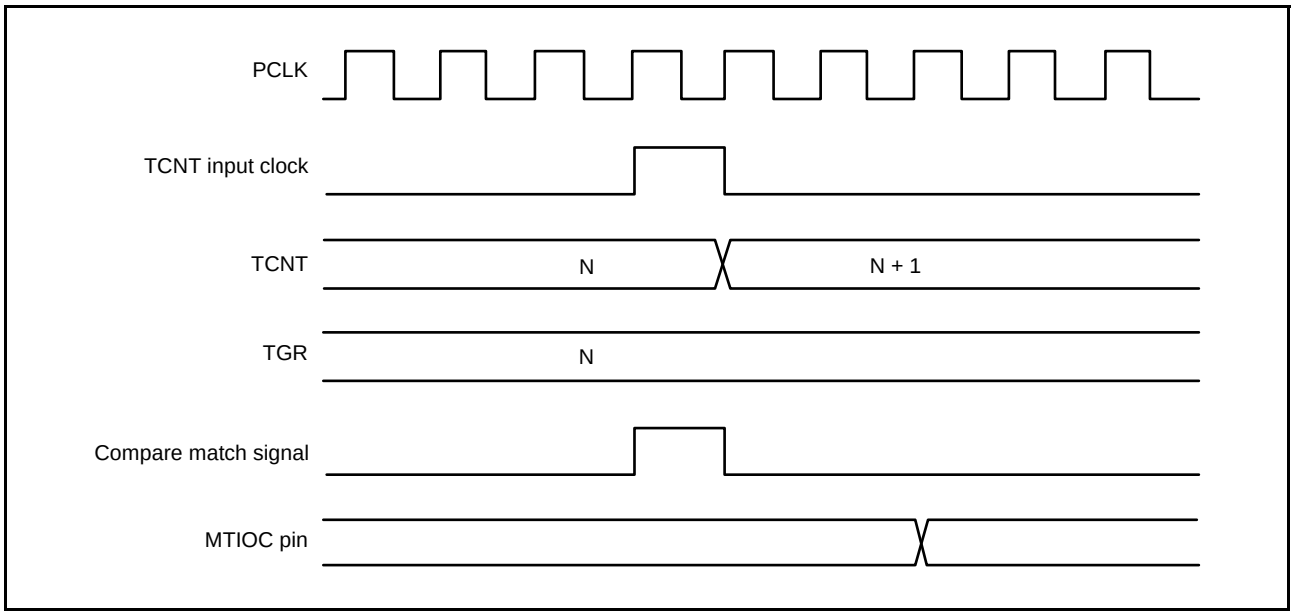


Figure 22.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 22.90 shows the input capture signal timing.

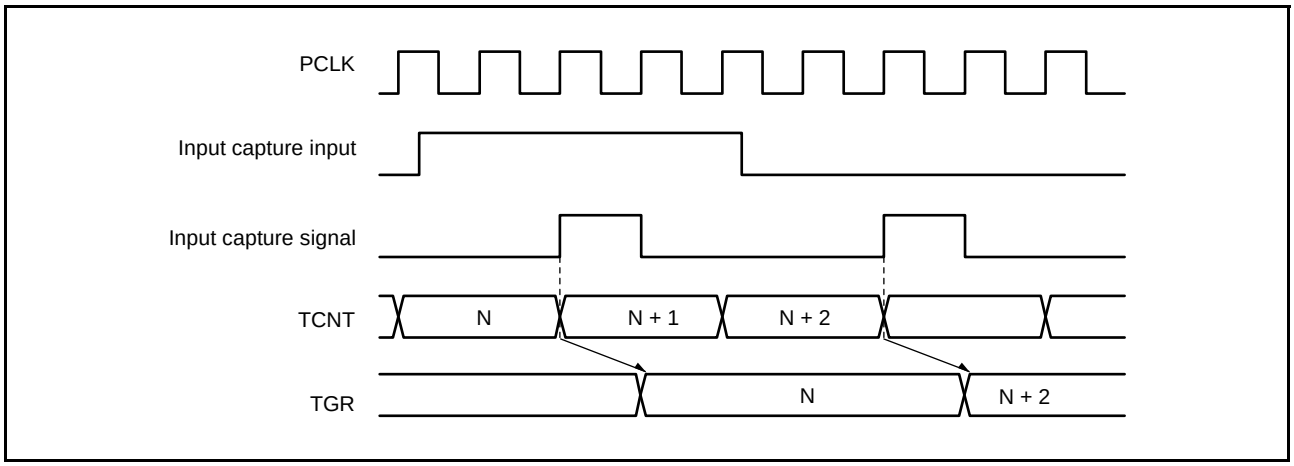


Figure 22.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 22.91 and Figure 22.92 show the timing when counter clearing on compare match is specified, and Figure 22.93 shows the timing when counter clearing on input capture is specified.

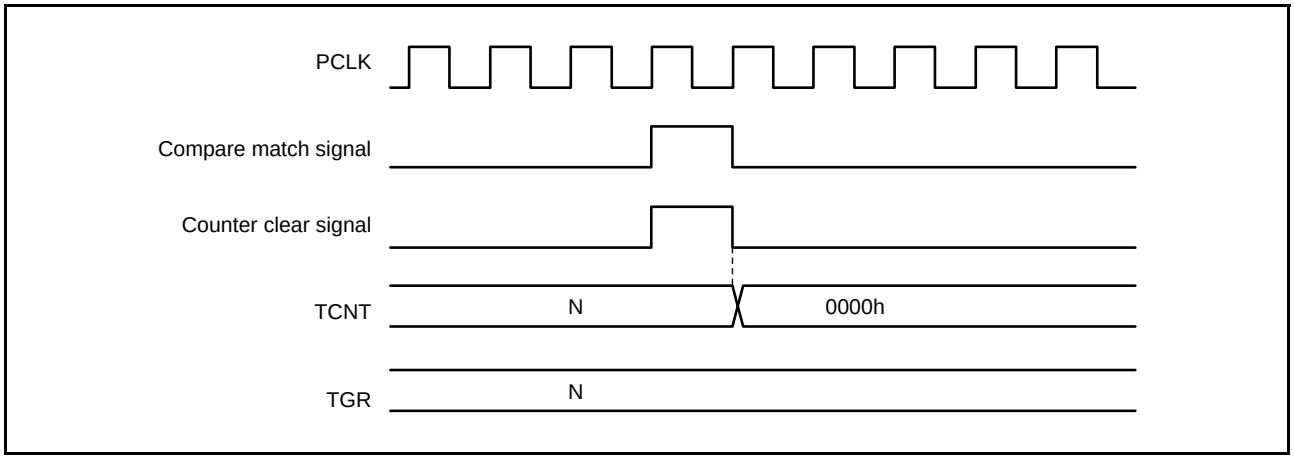


Figure 22.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

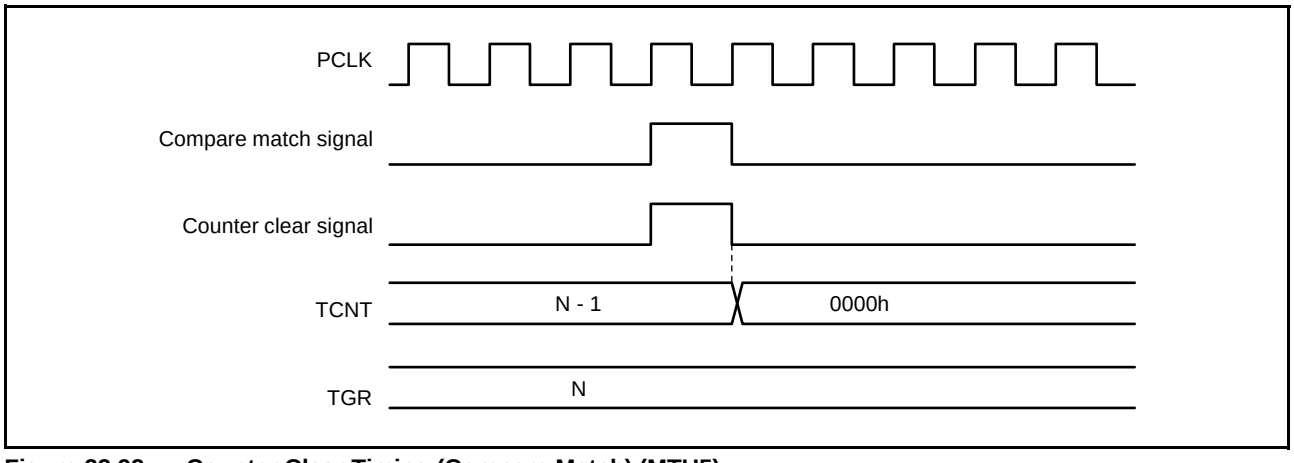


Figure 22.92 Counter Clear Timing (Compare Match) (MTU5)

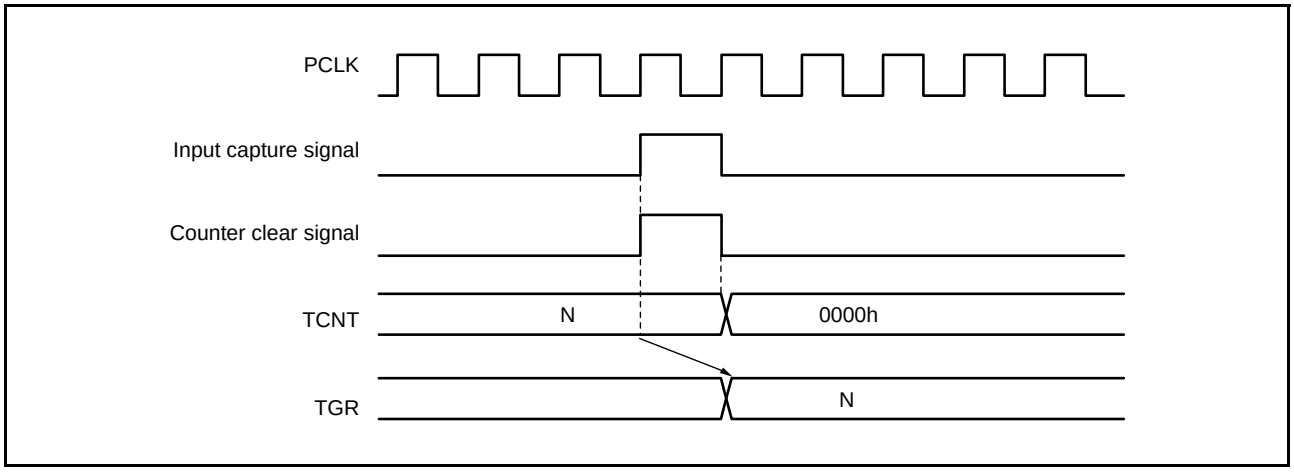


Figure 22.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 22.94 to Figure 22.96 show the timing in buffer operation.

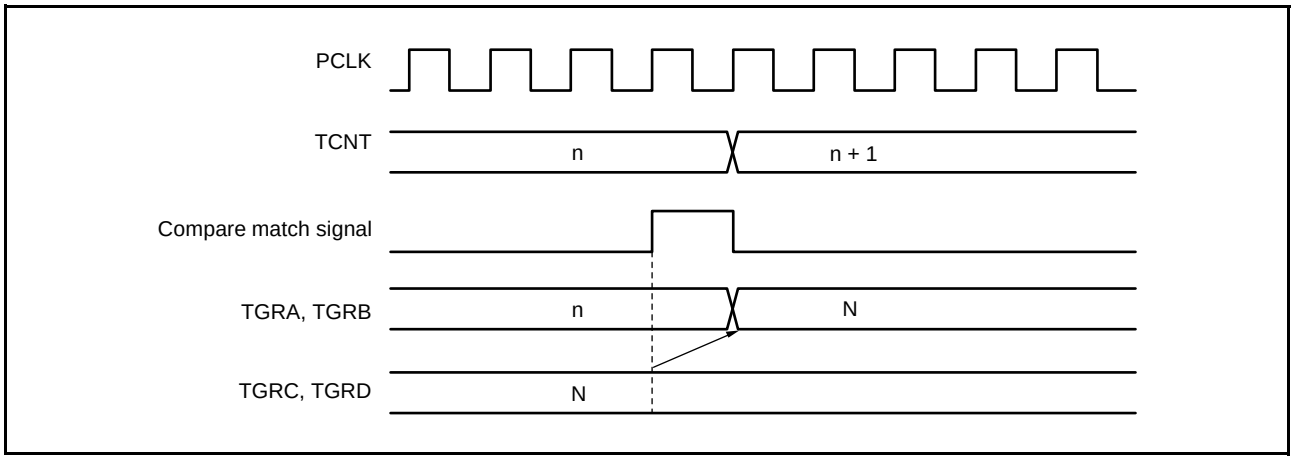


Figure 22.94 Buffer Operation Timing (Compare Match)

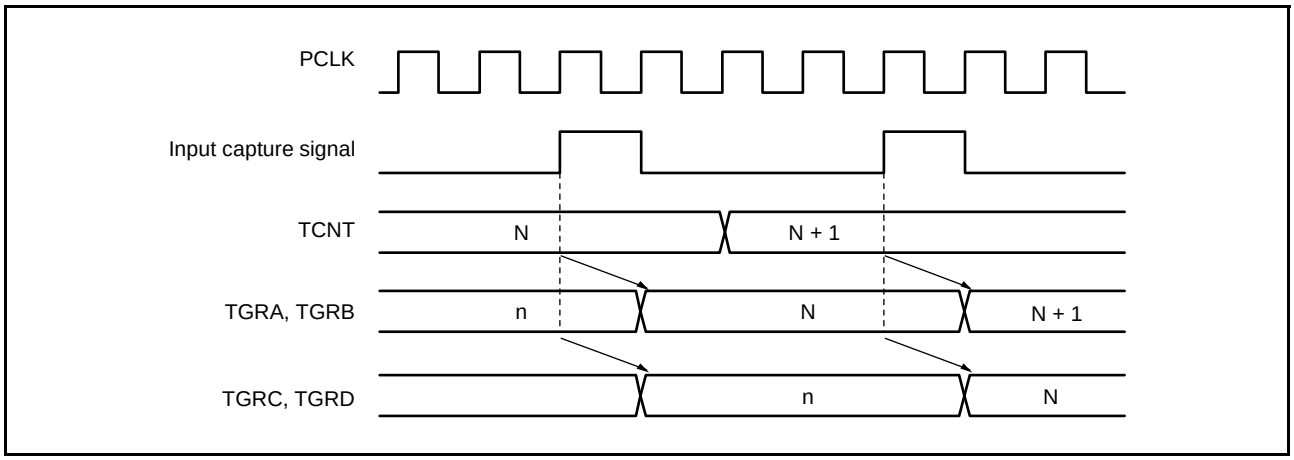


Figure 22.95 Buffer Operation Timing (Input Capture)

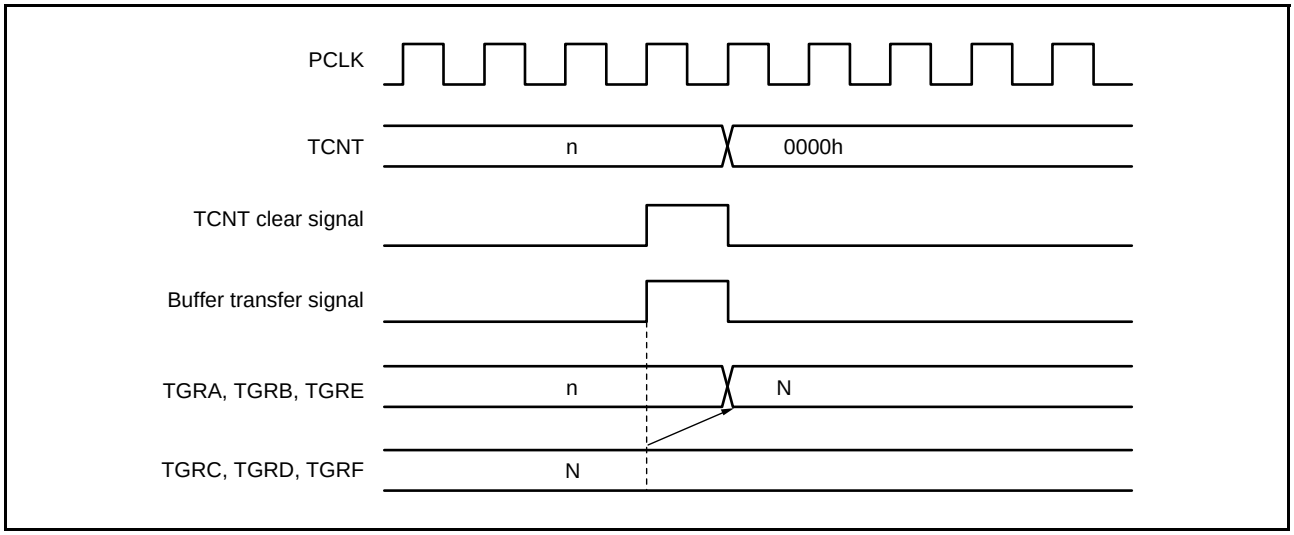


Figure 22.96 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 22.97 to Figure 22.99 show the buffer transfer timing in complementary PWM mode.

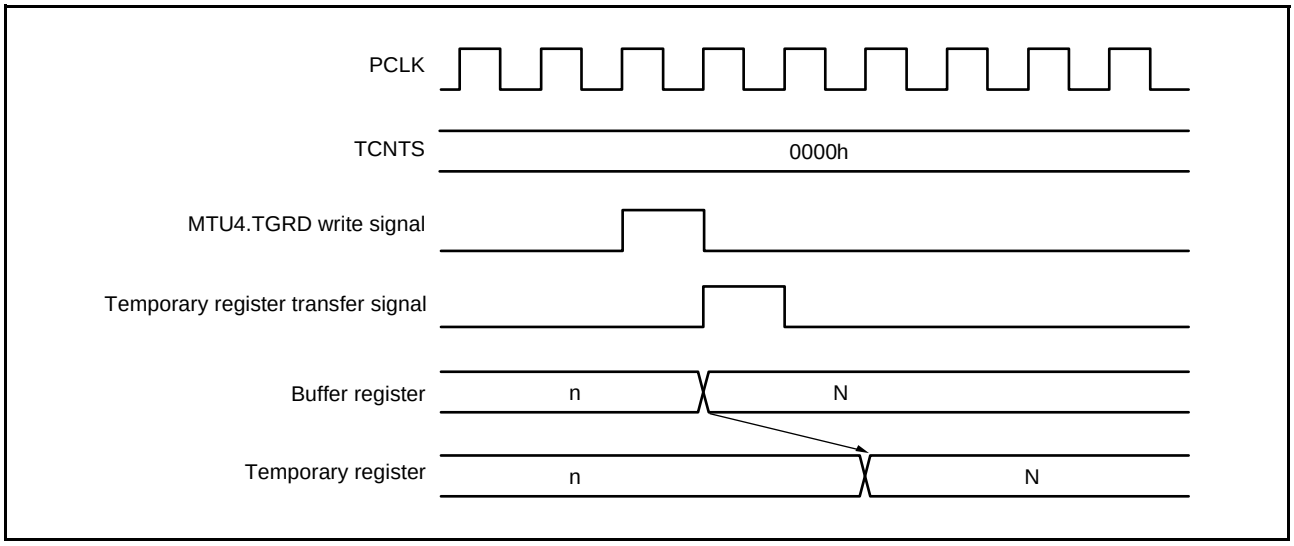


Figure 22.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

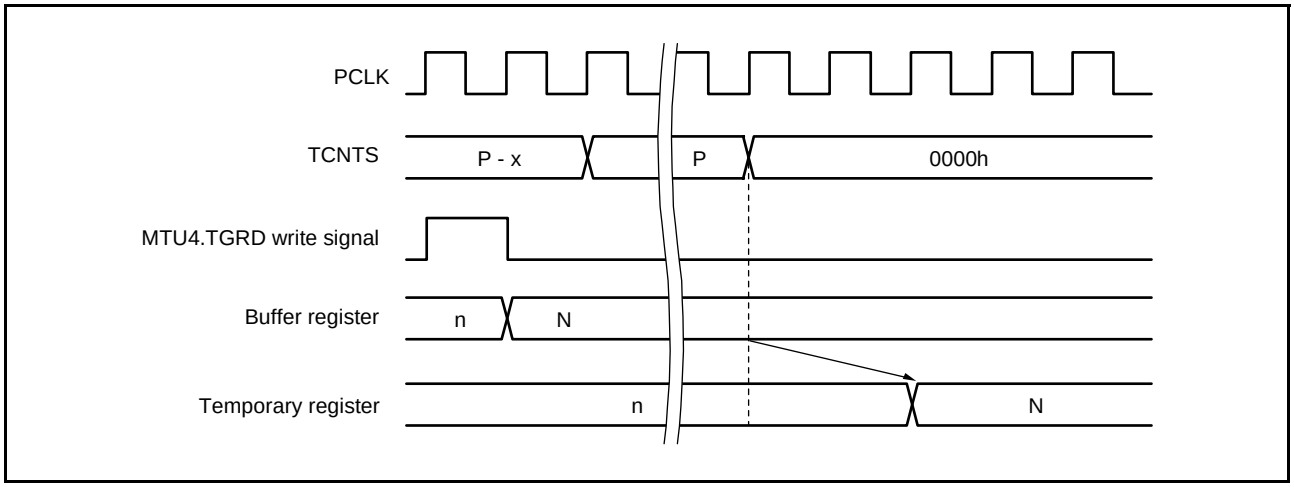


Figure 22.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

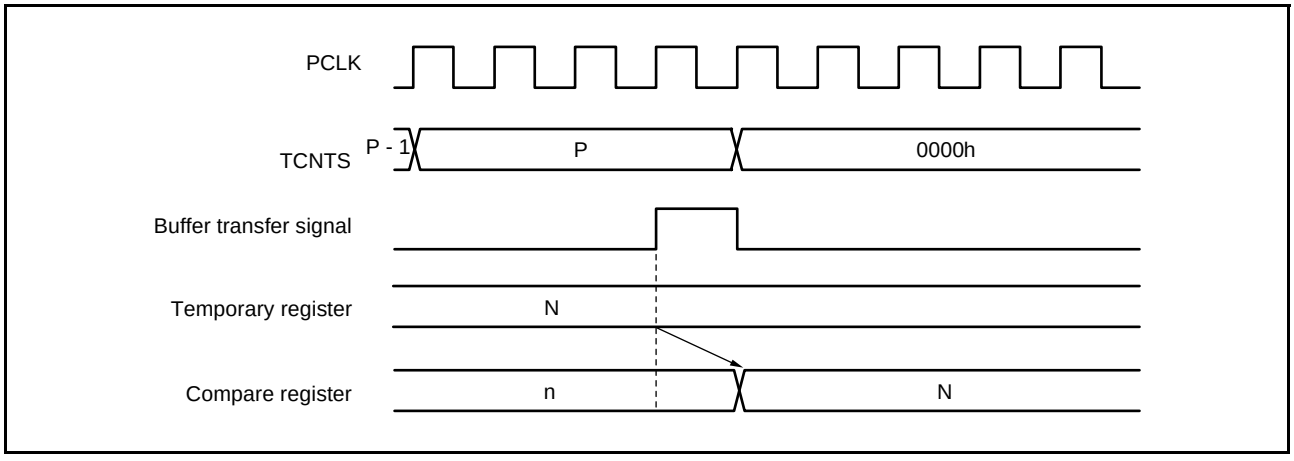


Figure 22.99 Transfer Timing from Temporary Register to Compare Register

22.5.2 Interrupt Signal Timing

(1) Timing for TGI Interrupt by Compare Match

Figure 22.100 and Figure 22.101 show the TGI interrupt request signal timing on compare match.

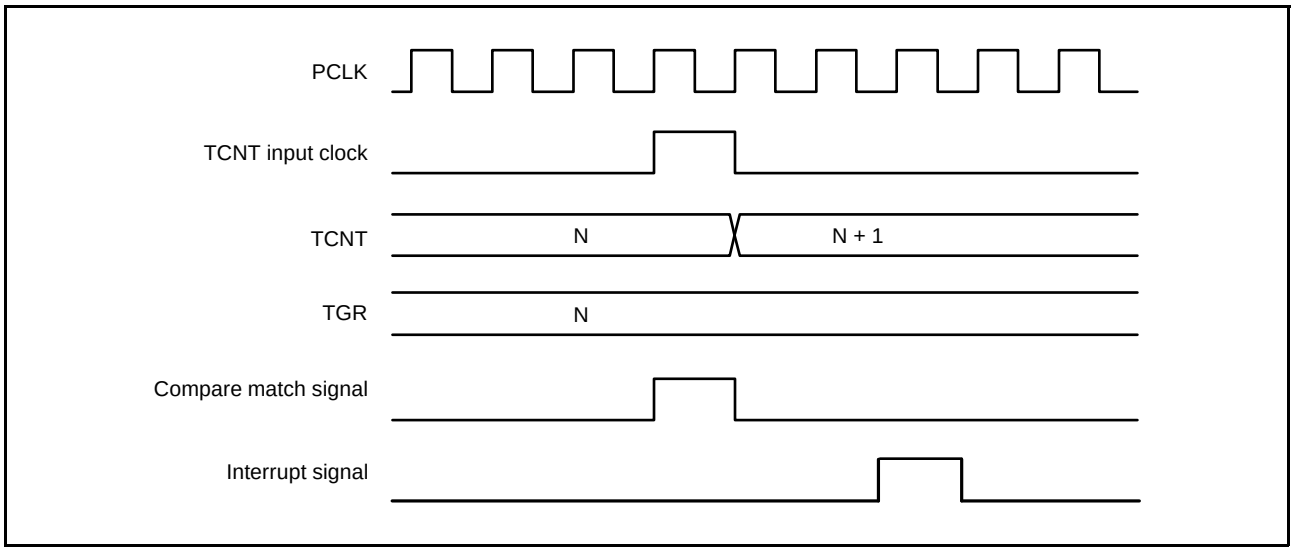


Figure 22.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

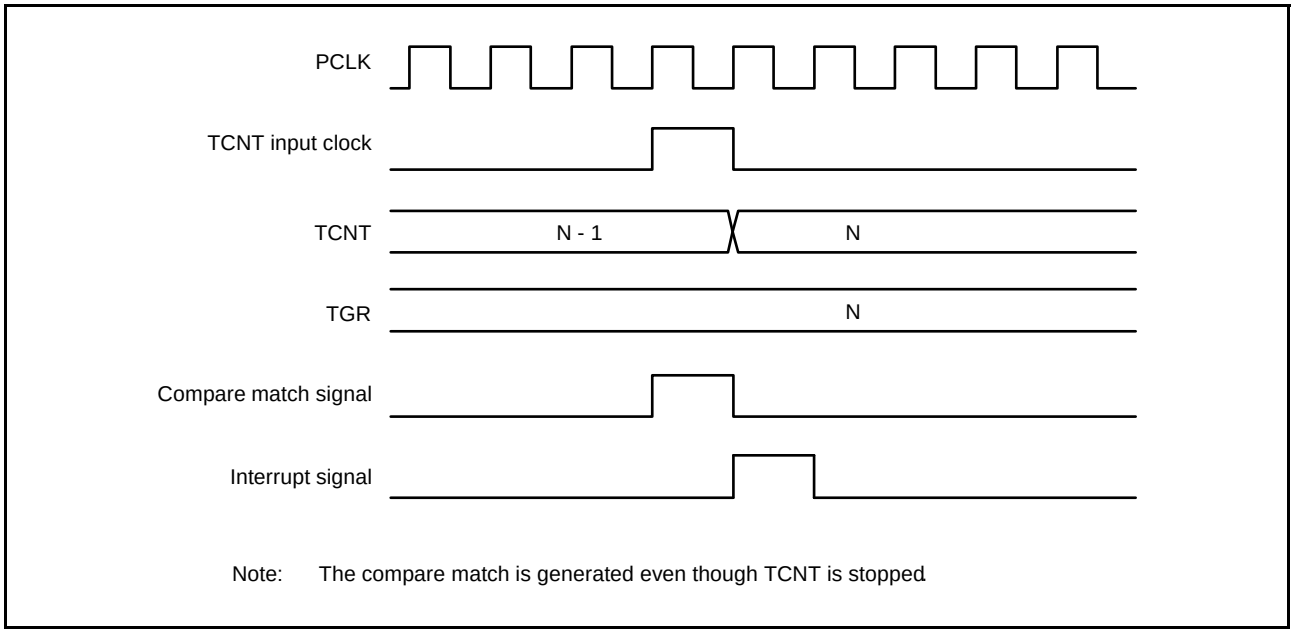


Figure 22.101 TGI Interrupt Timing (Compare Match) (MTU5)

(2) Timing for TGI Interrupt by Input Capture

Figure 22.102 and Figure 22.103 show TGI interrupt request signal timing on input capture.

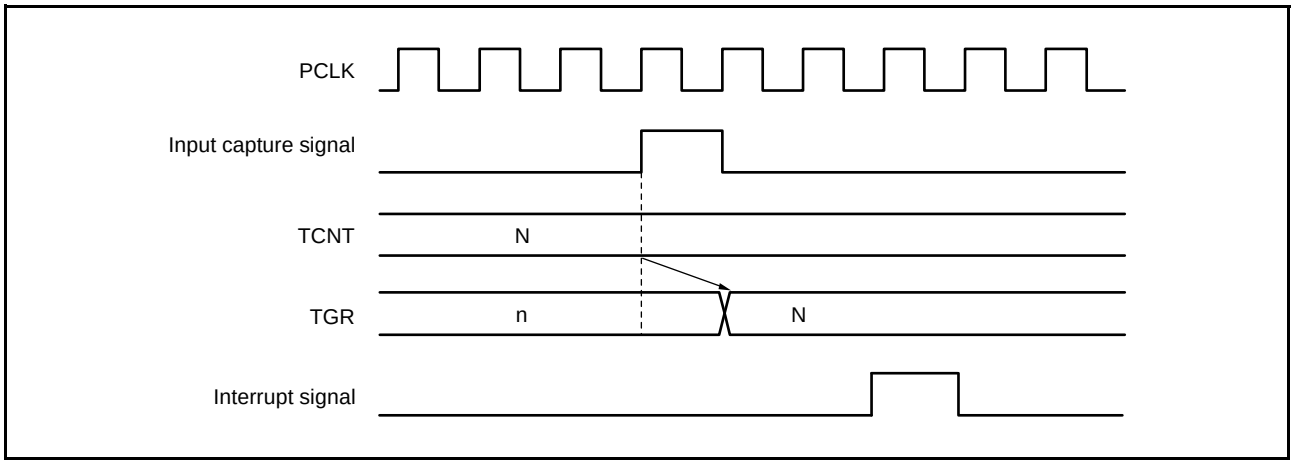


Figure 22.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

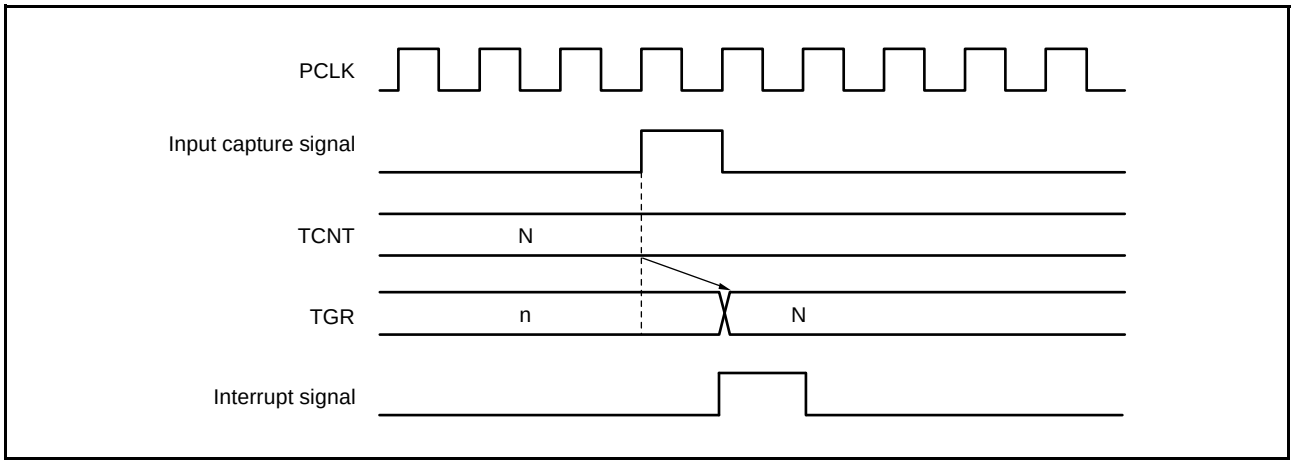


Figure 22.103 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 22.104 shows the TCIV interrupt request signal timing on overflow.
Figure 22.105 shows the TCIU interrupt request signal timing on underflow.

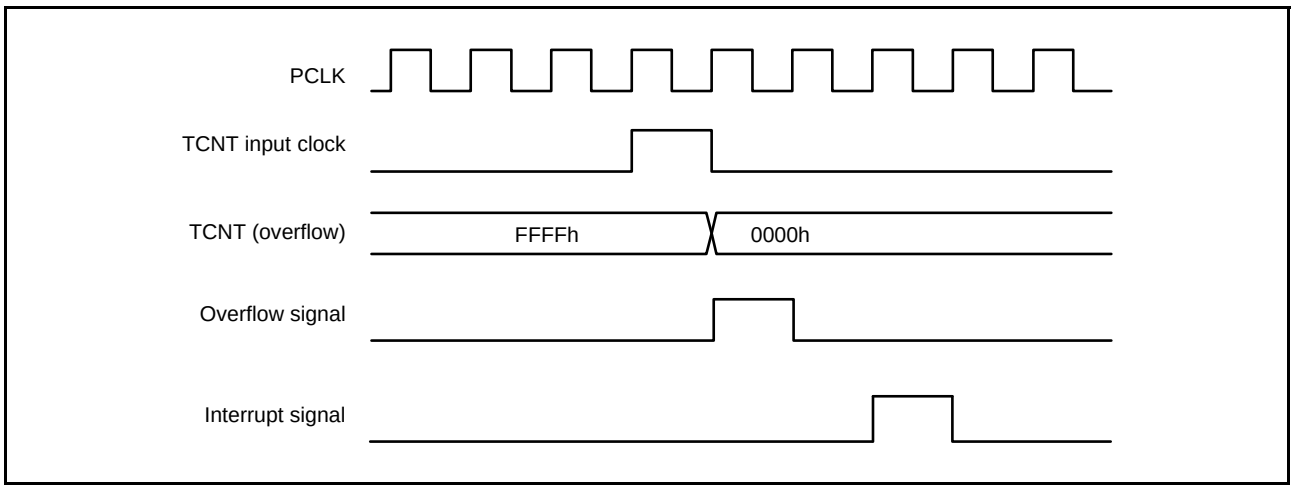


Figure 22.104 TCIV Interrupt Timing

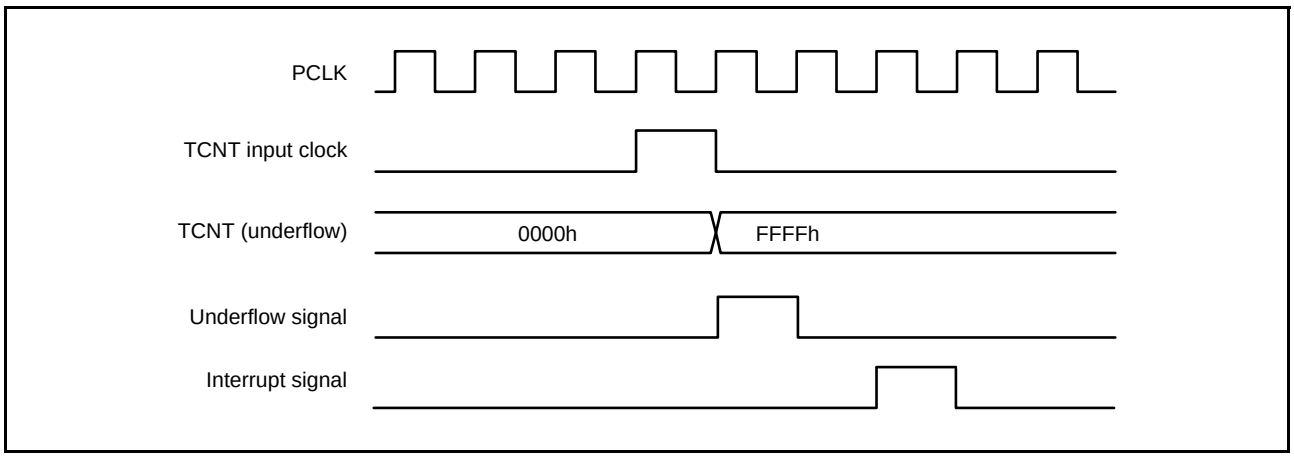


Figure 22.105 TCIU Interrupt Timing

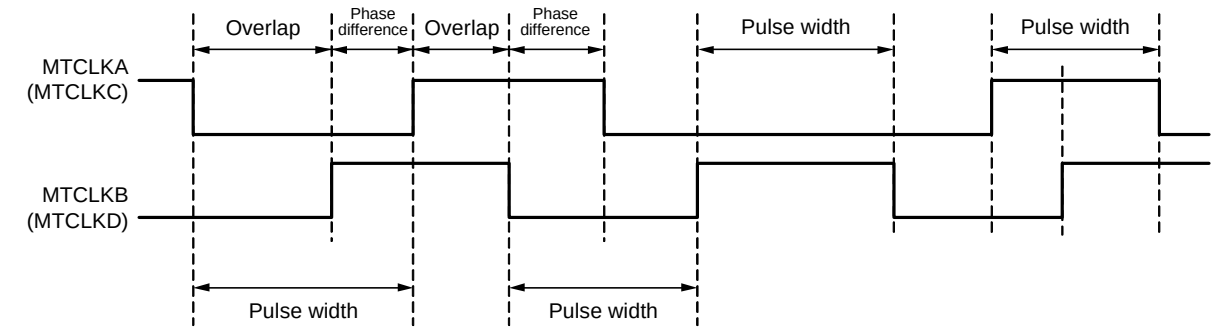
22.6 Usage Notes

22.6.1 Module Clock Stop Mode 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 mode. For details, refer to section 11, Low Power Consumption.

22.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles for single-edge detection, and at least 2.5 PCLK 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 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 22.106 shows the input clock conditions in phase counting mode.



Note: Phase difference and overlap: At least 1.5 PCLK cycles
Pulse width: At least 2.5 PCLK cycles

Figure 22.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

22.6.3 Notes on Cycle 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

$$f = \frac{CNTCLK}{(N + 1)}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] bits
N: TGR setting

22.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in a TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 22.107 shows the timing in this case.

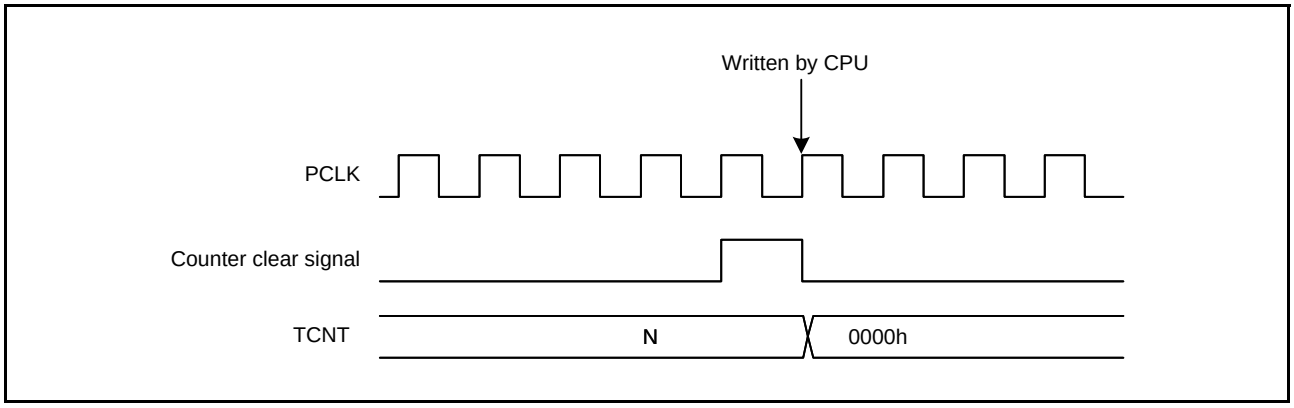


Figure 22.107 Contention between TCNT Write and Counter Clear Operations

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 22.108 shows the timing in this case.

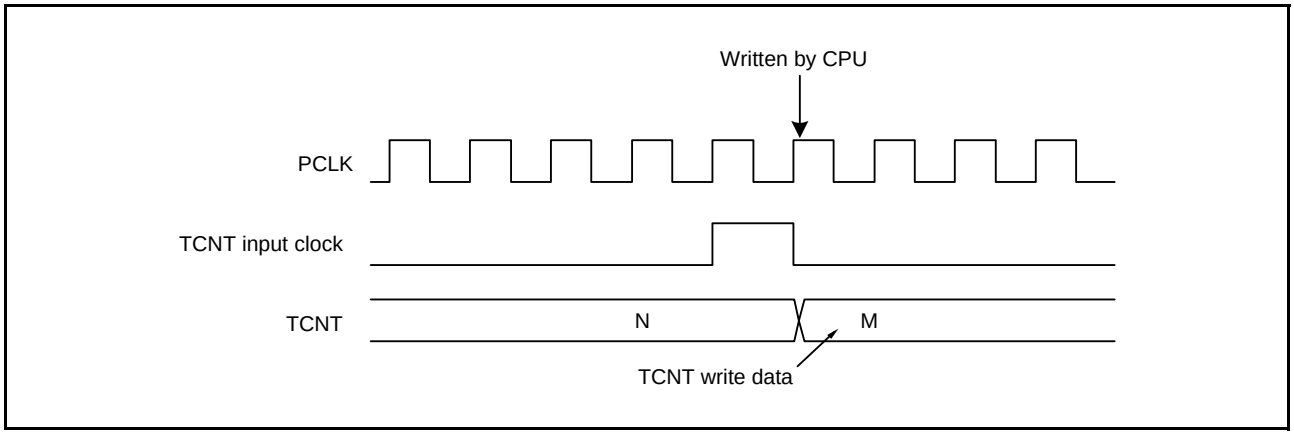


Figure 22.108 Contention between TCNT Write and Increment Operations

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 22.109 shows the timing in this case.

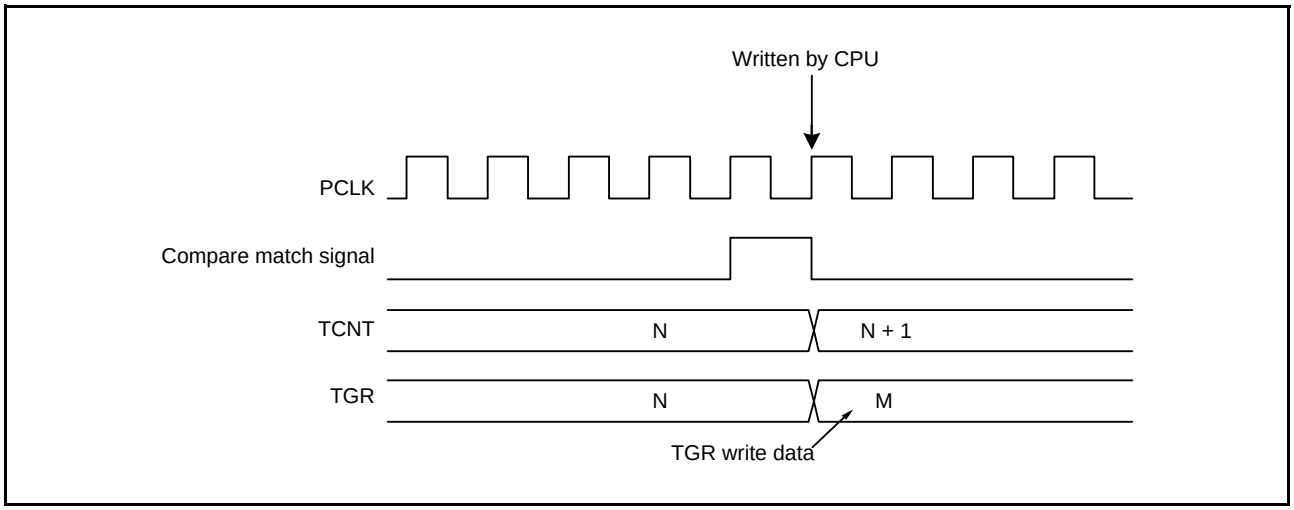


Figure 22.109 Contention between TGR Write Operation and Compare Match

22.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 22.110 shows the timing in this case.

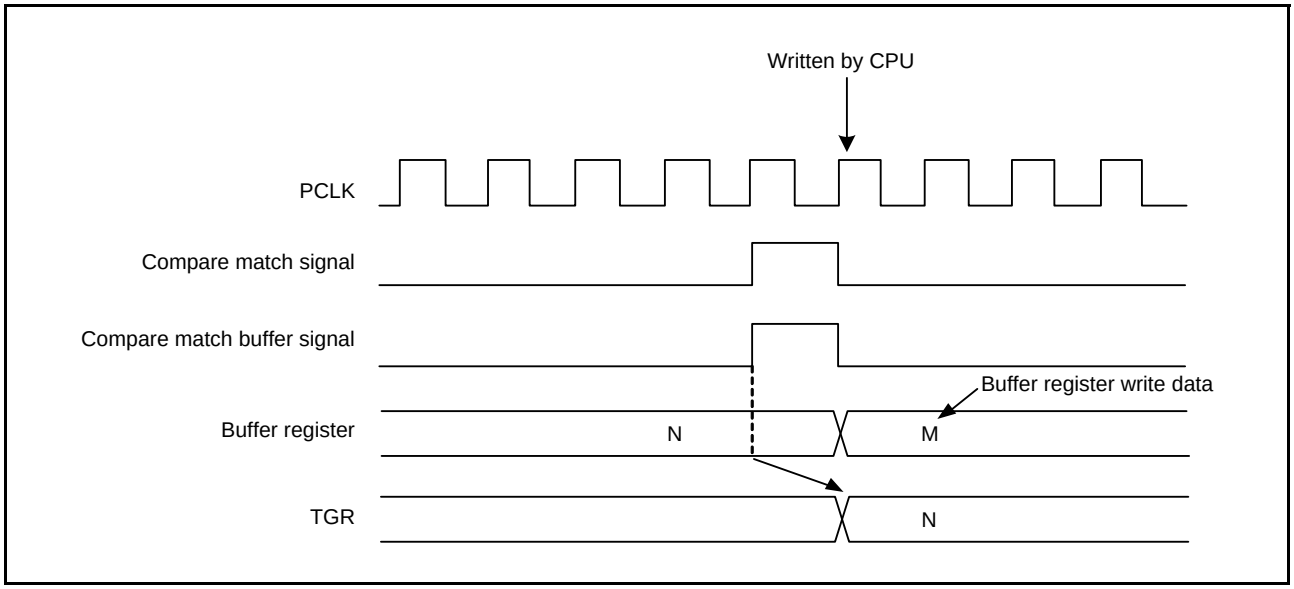


Figure 22.110 Contention between Buffer Register Write Operation and Compare Match

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 operation transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 22.111 shows the timing in this case.

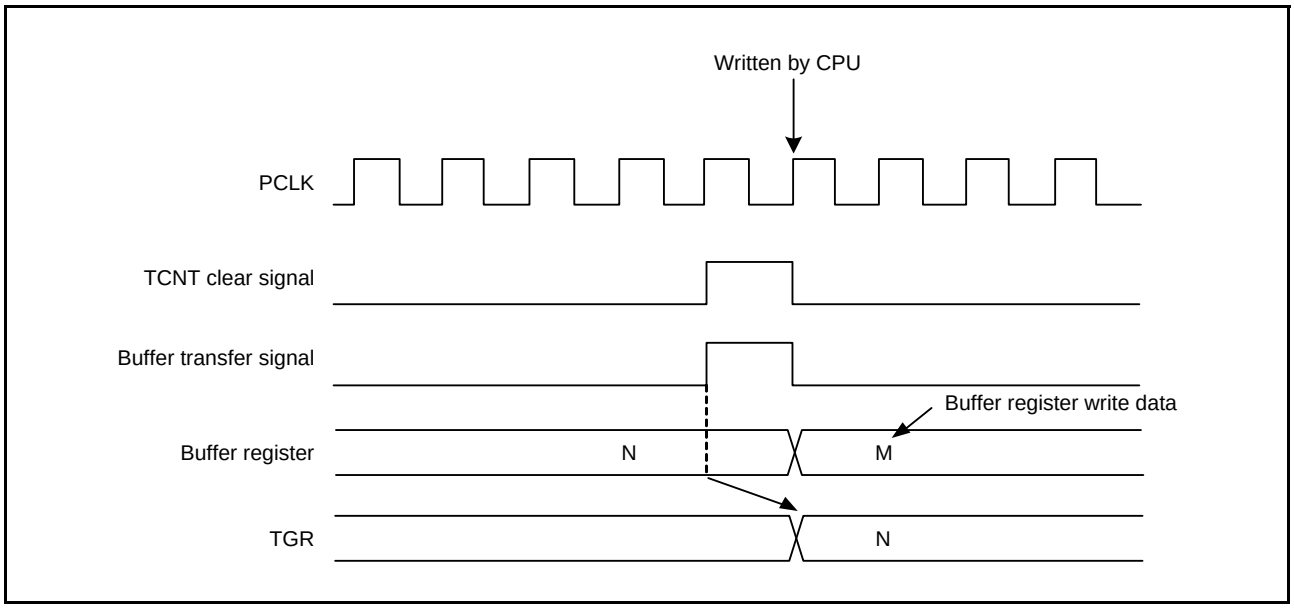


Figure 22.111 Contention between Buffer Register Write and TCNT Clear Operations

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 22.112 shows the timing in this case.

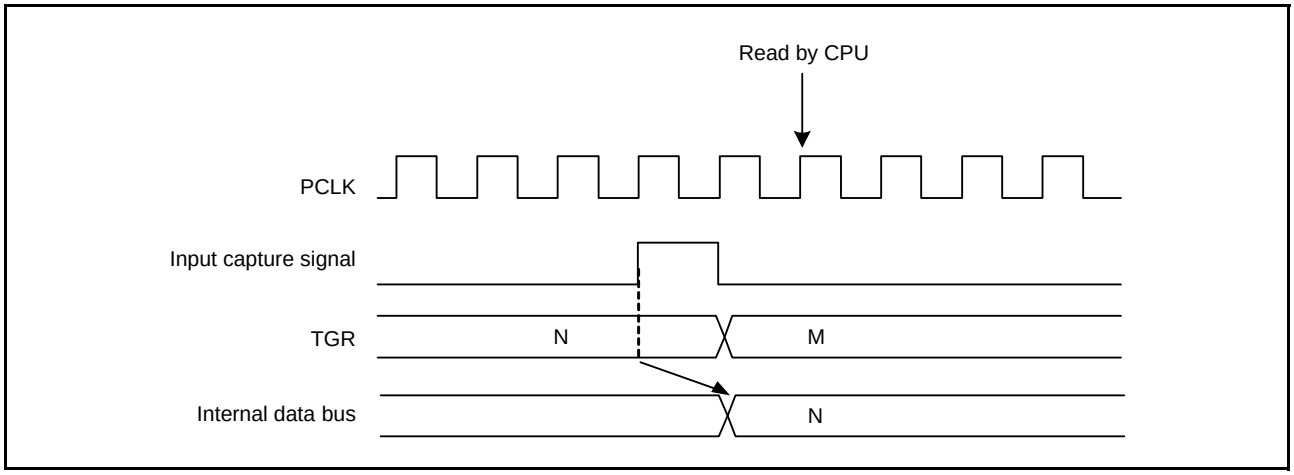


Figure 22.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

22.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 22.113 and Figure 22.114 show the timing in this case.

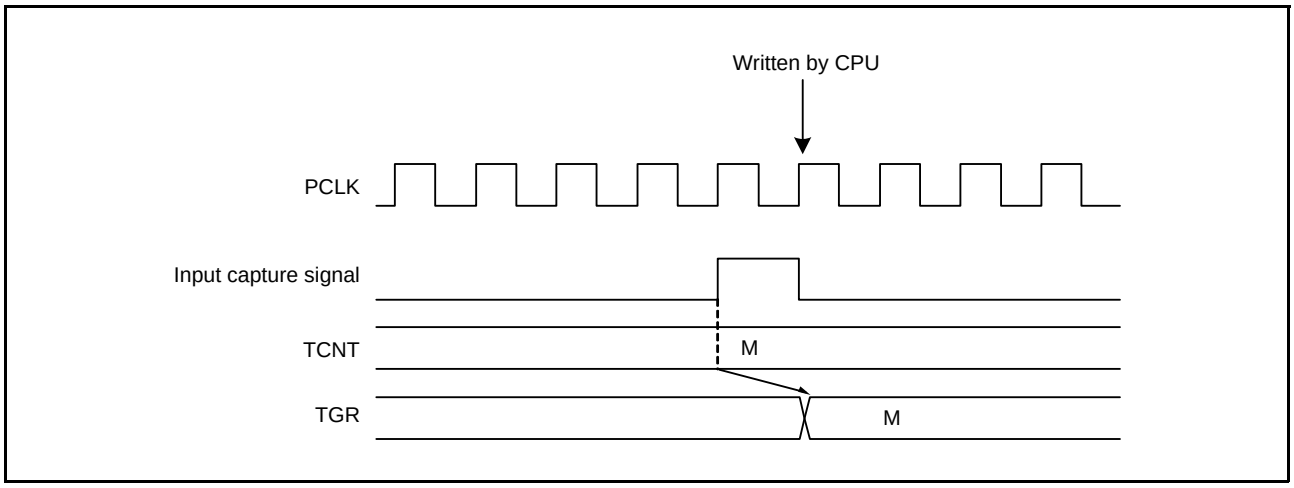


Figure 22.113 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

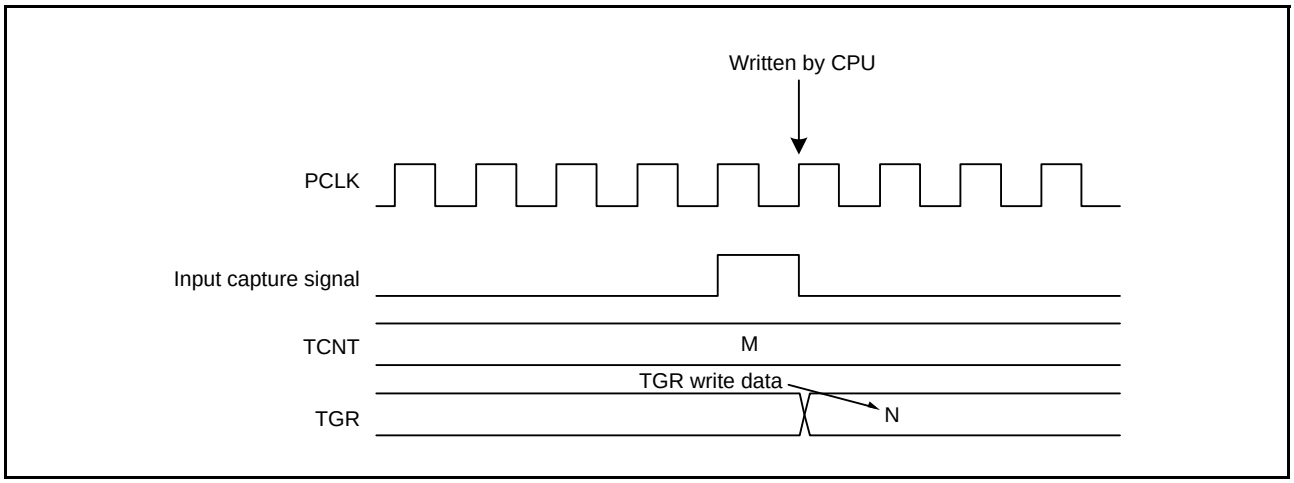


Figure 22.114 Contention between TGR Write Operation and Input Capture (MTU5)

22.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 22.115 shows the timing in this case.

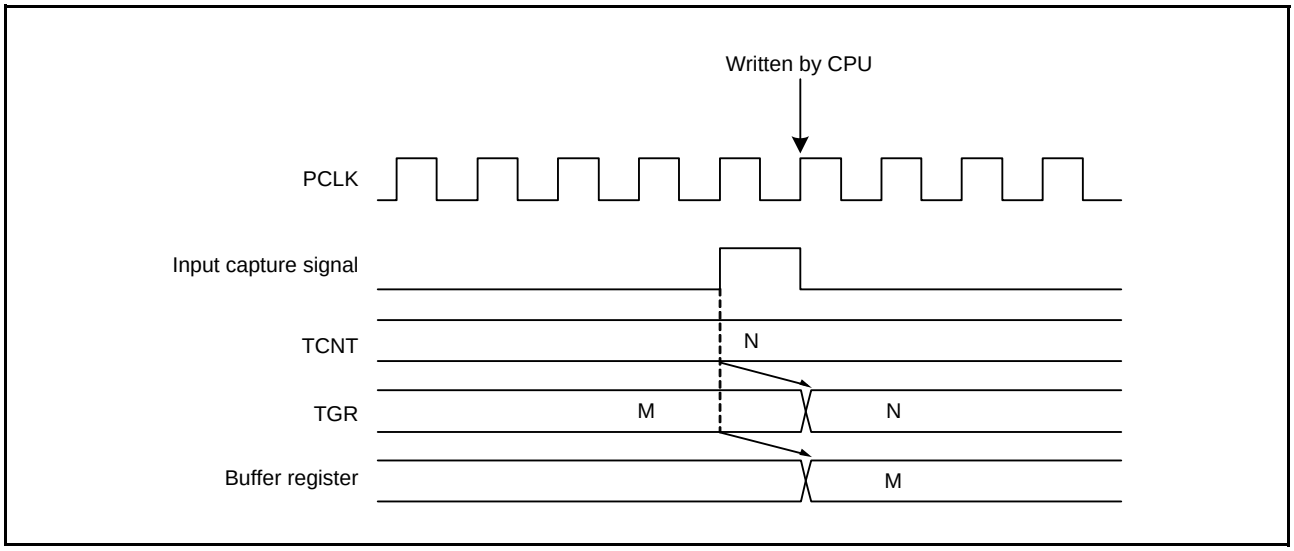


Figure 22.115 Contention between Buffer Register Write Operation and Input Capture

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 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 count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.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 22.116 shows the timing in this case.
When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

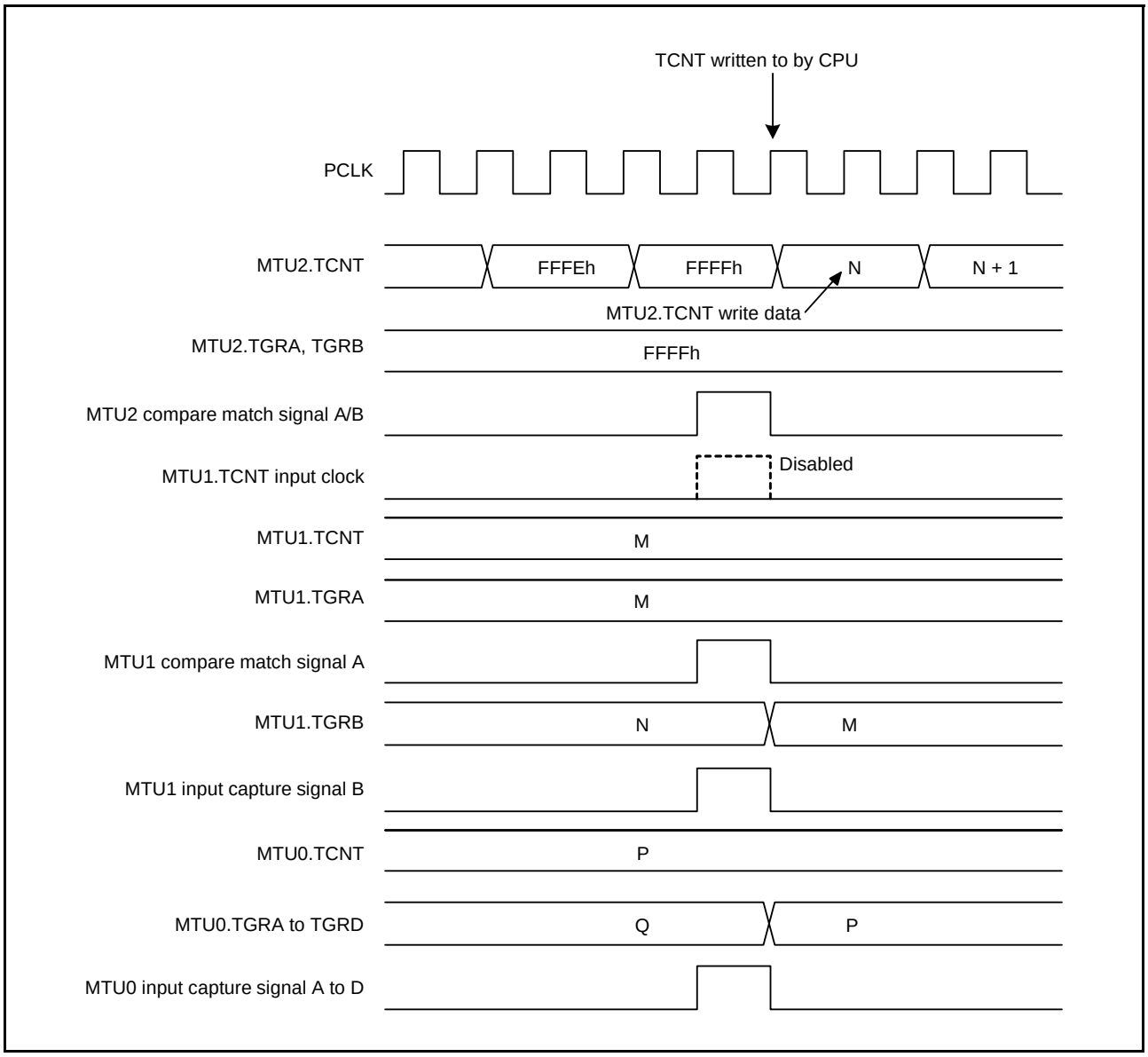


Figure 22.116 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

22.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, MTU3.TCNT is set to the timer dead time register (TDDR) value and MTU4.TCNT is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 22.117 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

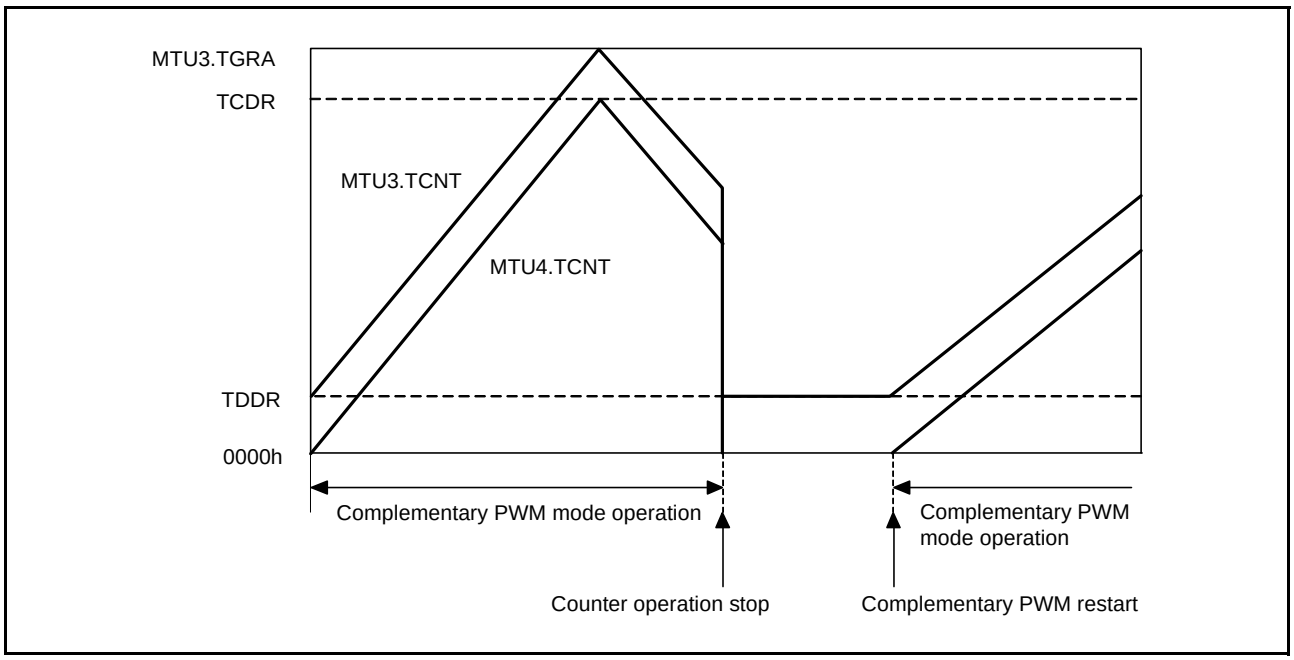


Figure 22.117 Counter Value When Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

22.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDR), and compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. Also, the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit should be set to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit to 1 also disables MTIOC4D pin waveform output.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR. When the MTU3.TMDR.BFA bit is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA, and TCBR functions as a buffer register for TCDR.

22.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit to 1 also disables MTIOC4D pin waveform output.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit. For example, if the MTU3.TMDR.BFA bit is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA.

While the MTU3.TGRC and MTU3.TGRD are operating as buffer registers, the corresponding TGIC and TGID interrupt requests are never generated.

Figure 22.118 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit set to 1 and the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit set to 0. (m = A to D)

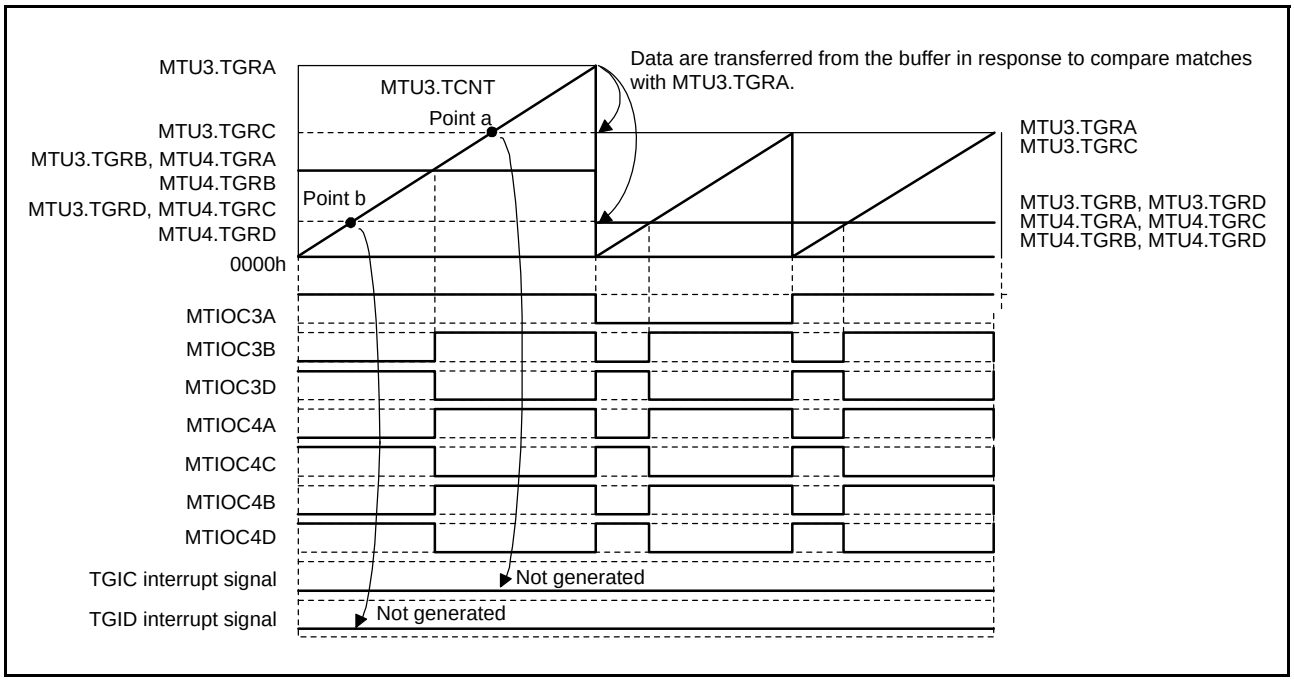


Figure 22.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

22.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the TSTR.CST3 bit is set to 1. In this state, the MTU4.TCNT count clock source and count edge are determined by the MTU3.TCR setting.

In reset-synchronized PWM mode, with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT count up to FFFFh, then a compare match occurs with MTU3.TGRA, and MTU3.TCNT and MTU4.TCNT are both cleared. In this case, the corresponding TCIV interrupt request is not generated.

Figure 22.119 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source.

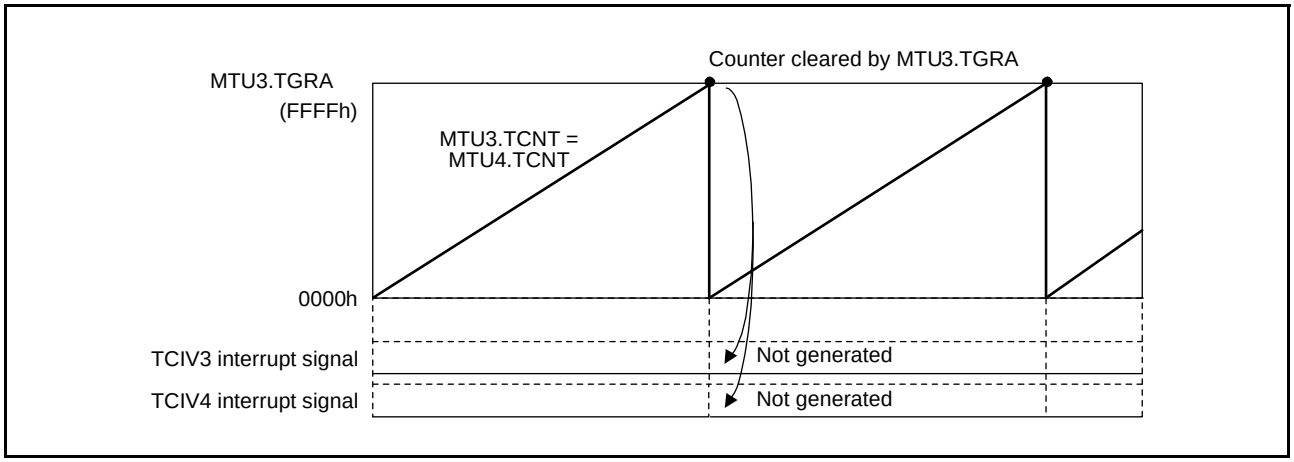


Figure 22.119 Overflow Flags in Reset-Synchronized PWM Mode

22.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the corresponding TCIV interrupt is not generated. If an overflow and counter clearing due to an input capture occur simultaneously, an input capture interrupt signal is output and an overflow interrupt signal is not output.
Figure 22.120 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

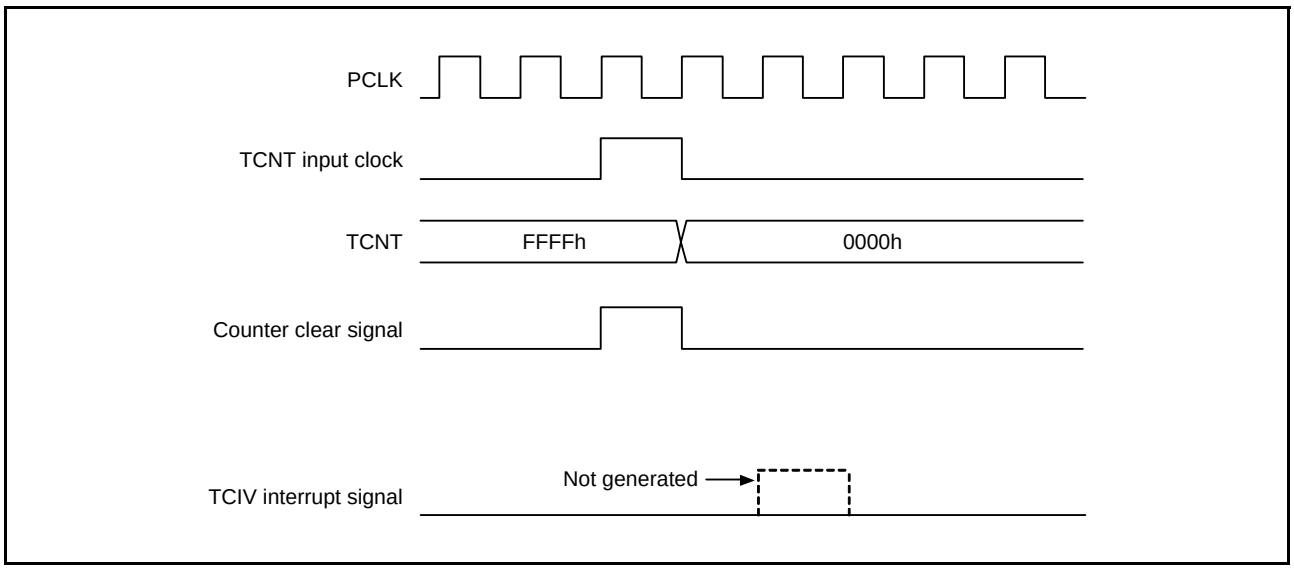


Figure 22.120 Contention between Overflow and Counter Clearing

22.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT up-count or down-count in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. The corresponding interrupt is not generated.
Figure 22.121 shows the operation timing when there is contention between TCNT write operation and overflow.

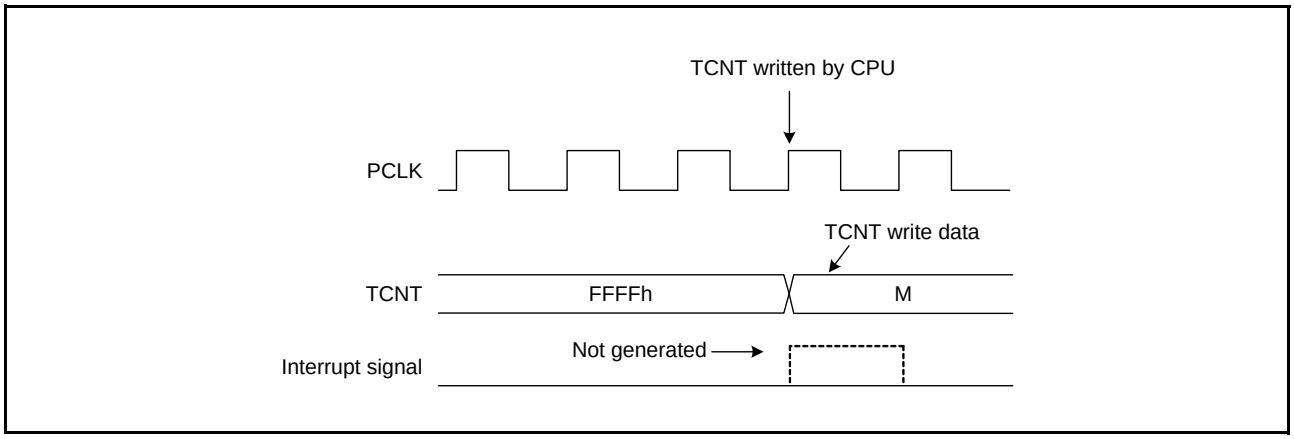


Figure 22.121 Contention between TCNT Write Operation and Overflow

22.6.19 Notes 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, if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D) 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 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 or Reset-Synchronized PWM Mode

When complementary PWM mode or reset-synchronized PWM mode is selected for MTU3 or MTU4, use the TOCR1.OLSP bit and TOCR1.OLSN bit to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h. The negative-phase output level when the TDER.TDER bit is set to 0 (no dead time is generated) in complementary PWM mode is the inverse of the positive-phase output level according to the TOCR1.OLSP bit setting, not the TOCR1.OLSN bit setting.

22.6.21 Interrupts during Periods in the Module Stop State

When an module that has issued an interrupt request enters the module stop state, clearing the source of the interrupt for the CPU or activation signal for the DTC/DMAC is not possible.
Accordingly, disable interrupts, etc. before making the settings for the module stop state.

22.6.22 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters (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 of 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 22.2.8, Timer Input Capture Control Register (TICCR).

22.6.23 Notes When Complementary PWM Mode Output Protection Functions are Not Used

The complementary PWM mode output protection functions are initially enabled. Refer to section 23, Port Output Enable 2 (POE2a), for details.

22.6.24 Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers

Do not set an MTU5.TGRm (m = U, V, W) bit to the value of the corresponding MTU5.TCNTm (m = U, V, W) register 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) register plus one while counting by the MTU5.TCNTm (m = U, V, W) register is stopped, a compare-match will be generated even though counting is stopped. In this case, if the corresponding MTU5.TIER.TGIE5m (m = U, V, W) bit is also set to 1 (interrupt requests enabled), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is also 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 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCR.WRE = 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 PWM inverse-phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 22.122, 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 22.123, synchronous clearing occurs when any condition from among MTU3.TGRB ≤ TDDR, MTU4.TGRA ≤ TDDR, or MTU4.TGRB ≤ TDDR is satisfied.

The following method avoids the above phenomena.

- Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) set to at least double the value of the dead time data register (TDDR).

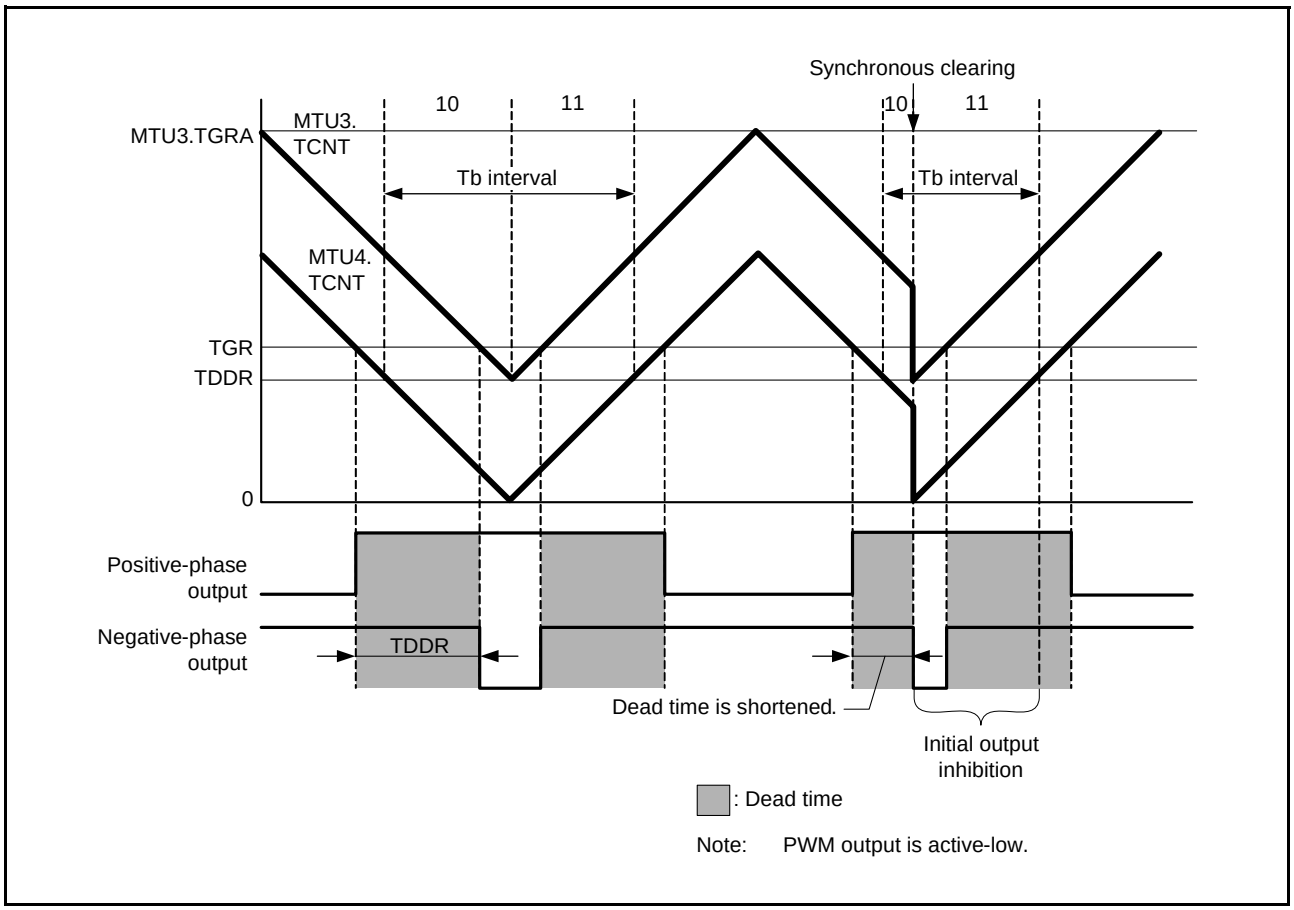


Figure 22.122 Example of Synchronous Clearing (When Condition 1 Applies)

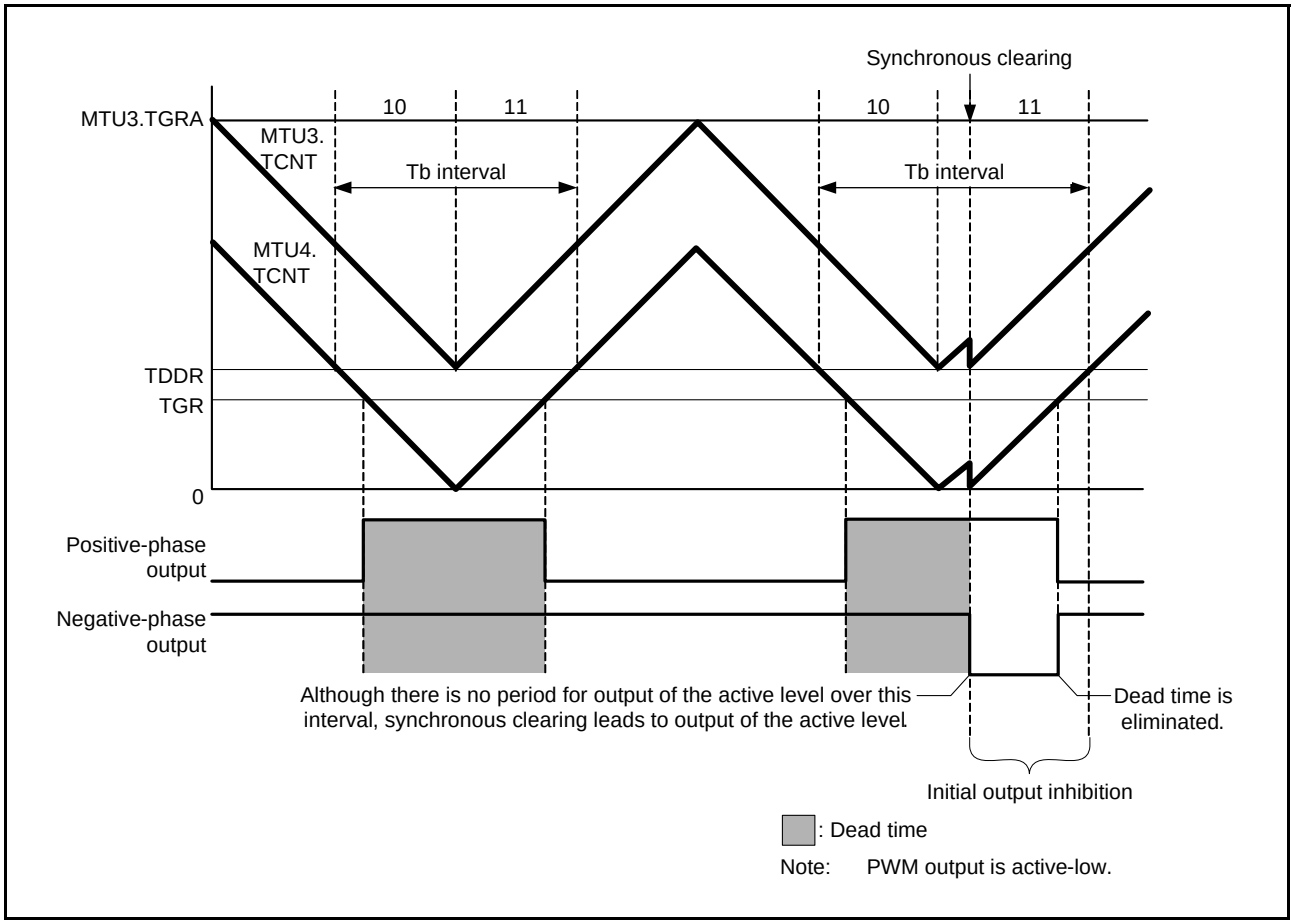


Figure 22.123 Example of Synchronous Clearing (When Condition 2 Applies)

22.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When TGR is set to 0000h, PCLK/1 clock is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the counter (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 22.124 shows the timing for continuous output of the interrupt signal in response to a compare match.

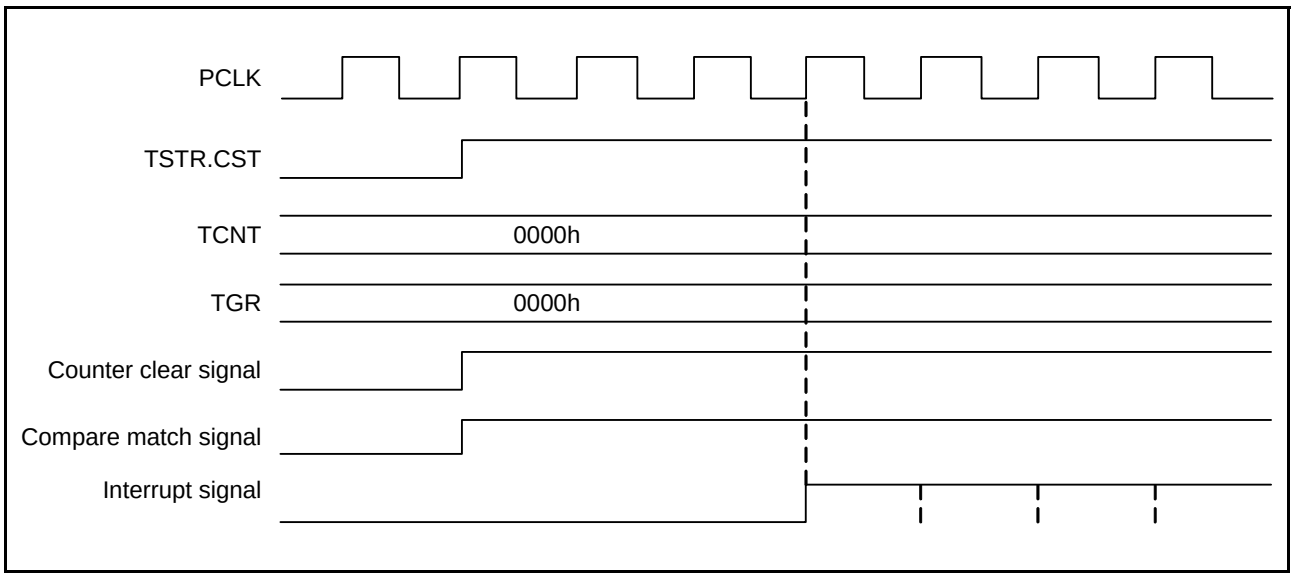


Figure 22.124 Continuous Output of Interrupt Signal in Response to a Compare Match

22.6.27 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to 0 and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during up-counting immediately after transfer. See Figure 22.125.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to the same value as the TRCR and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during down-counting immediately after transfer. See Figure 22.126.
- To issue an A/D converter start request linked with interrupt skipping, set the MTU4.TADCORA and MTU4.TADCORB registers so that $2 \leq$ MTU4.TADCORA/TADCORB $\leq$ TCDR - 2 is satisfied.

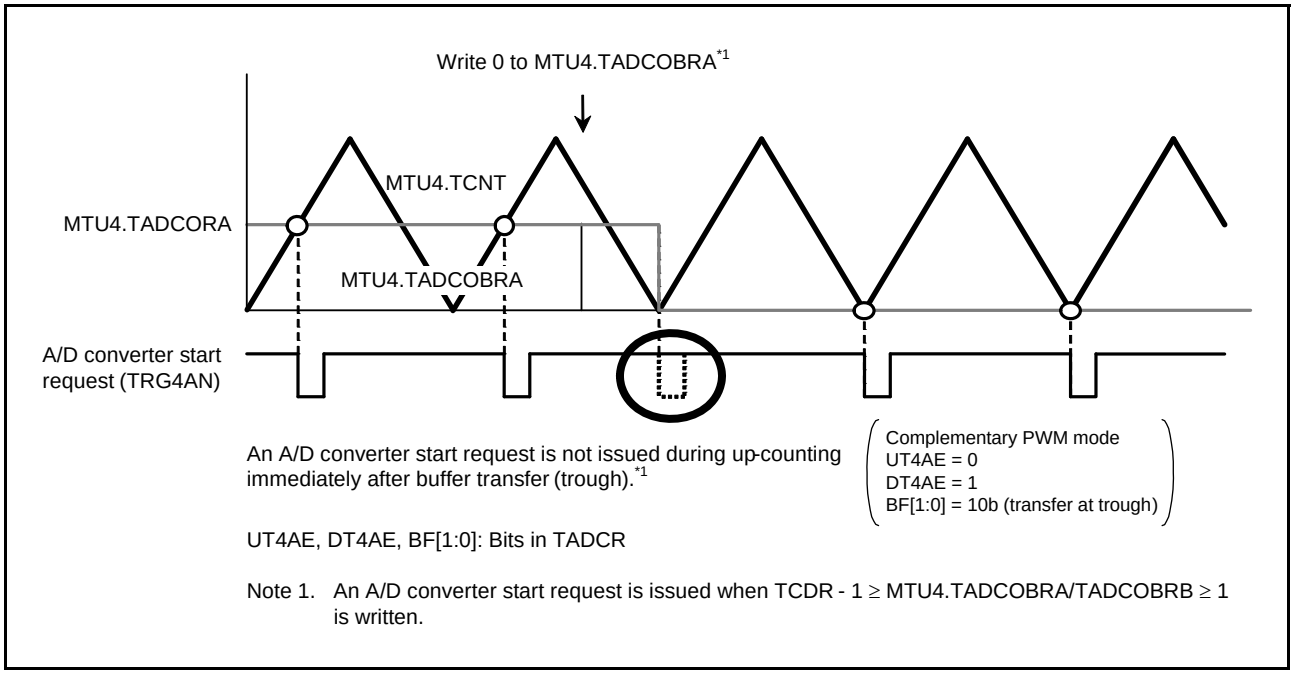


Figure 22.125 A/D Converter Start Request When 0 is Written to MTU4.TADCOBRA

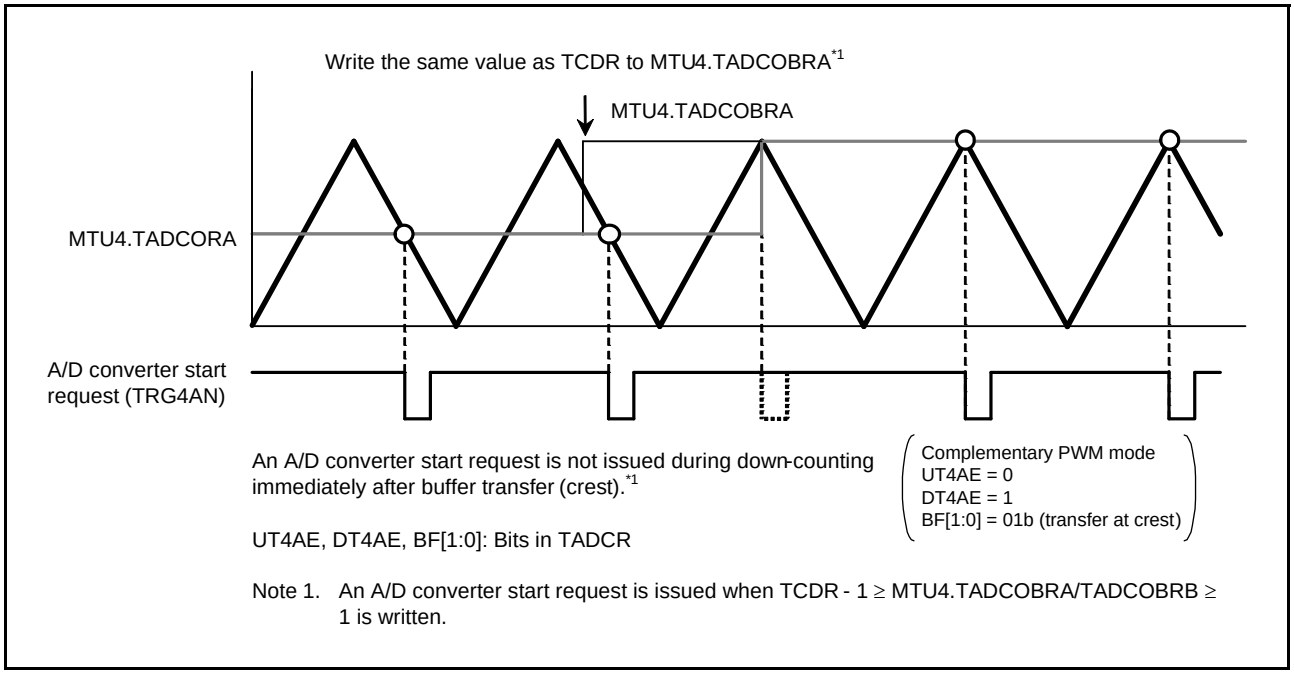


Figure 22.126 A/D Converter Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

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)
- PWM mode 1 (MTU0 to MTU4)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 4 (MTU1 and MTU2)
- Complementary PWM mode (MTU3 and MTU4)
- Reset-synchronized PWM mode (MTU3 and MTU4)

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. For an I/O port that is shut down, set the port direction registers (PDR), the port output data register (PODR), and the port mode register (PMR) to switch the port pins to be general output pins and for output of the non-active level. Set the TIOR for the MTU pins to disable output. Set the TOER for the complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D). For PWM output pins, output can also be cut by hardware, using port output enable 2(POE). 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 listed in Table 22.59.

Note that the following notations are used for operating modes.

Normal: Normal mode PWM1: PWM mode 1 PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4 CPWM: Complementary PWM mode RPWM: Reset-synchronized PWM mode

Table 22.59 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)

22.7.3 Overview of Pin 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, a waveform is not output to the MTIOCnB and MTIOCnD (n = 3, 4) pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 2, a waveform is not output to the cycle register pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In normal mode or PWM 2 mode, if the TGRC and TGRD operate as buffer registers, a waveform is not output to the pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register and there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports. Switch to normal mode, perform initialization with the TIOR register, and restore the TIOR register to its initial value. After that, operate the MTU in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: Channel number is substituted for “n” indicated in this section unless otherwise specified.

Pin initialization procedures are described below for the numbered combinations in Table 22.59. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 22.127 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

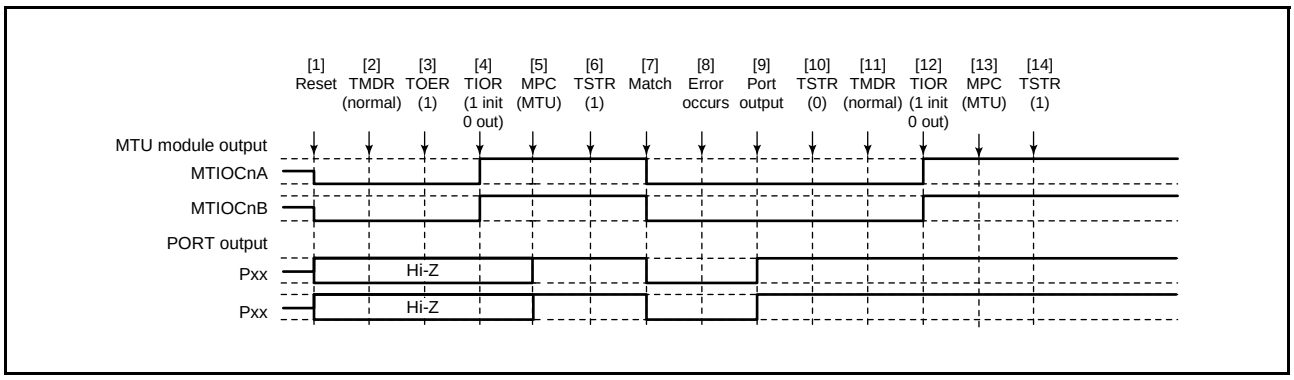


Figure 22.127 Error Occurrence in Normal Mode, Recovery in Normal Mode

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] After a reset, the TMDR setting is for normal mode.
[3] For MTU3 and MTU4, enable output with TOER before initializing the pins with TIOR.
[4] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] Output goes low on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR.
[11] This step is not necessary when restarting in normal mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 22.128 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

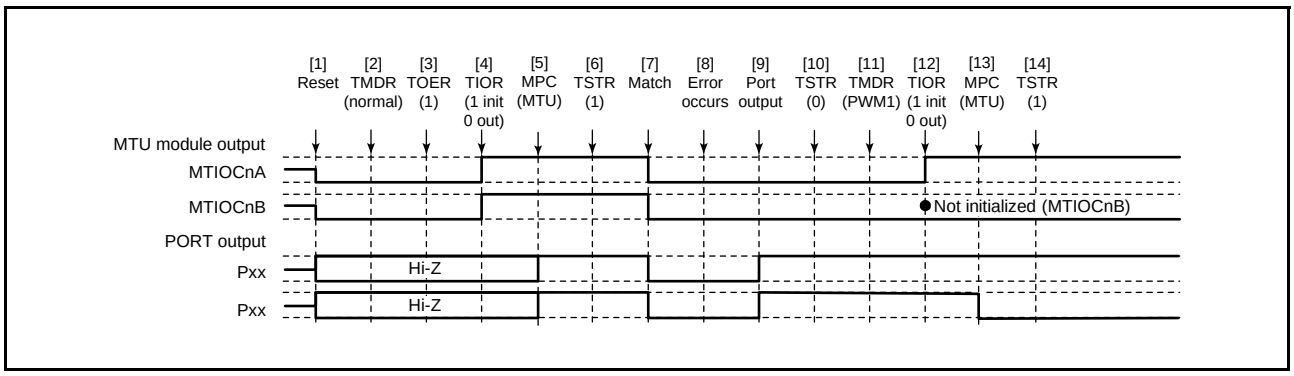


Figure 22.128 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.127.

[11] Set PWM mode 1.

[12] Set the TIOR register to initialize pins, i.e. so that the MTIOCnB (or MTIOCnD) does not produce a waveform in PWM mode 1. If a particular level should be output, set the port direction register (PDR) and the port output data register (PODR) so that the pins of the I/O port operate as general outputs.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 2

Figure 22.129 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

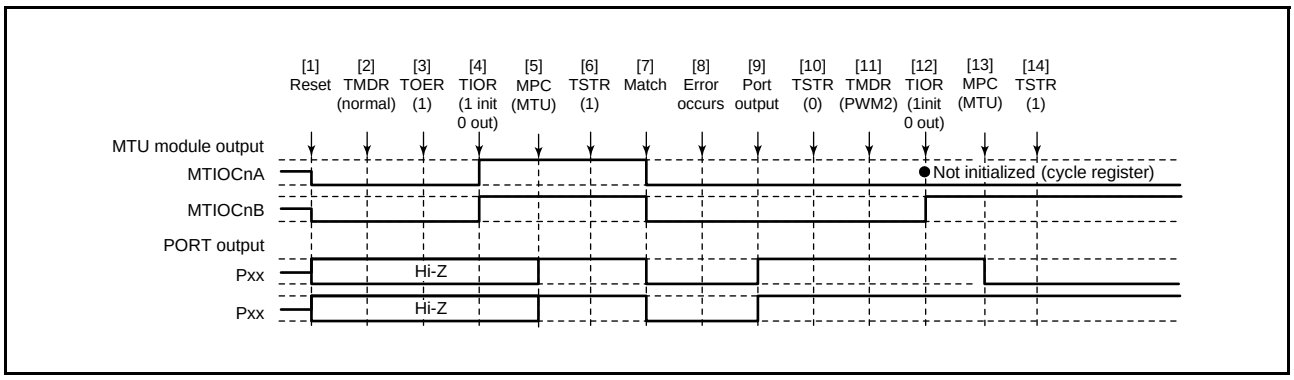


Figure 22.129 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 22.127.

[11] Set PWM mode 2.

[12] Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOER setting is not necessary.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 22.130 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

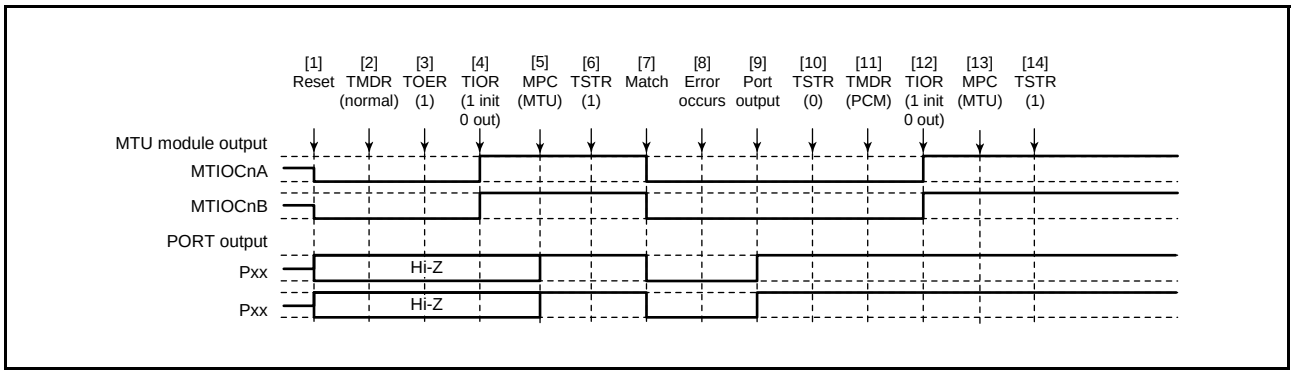


Figure 22.130 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 22.127.
[11] Set the phase counting mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.131 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

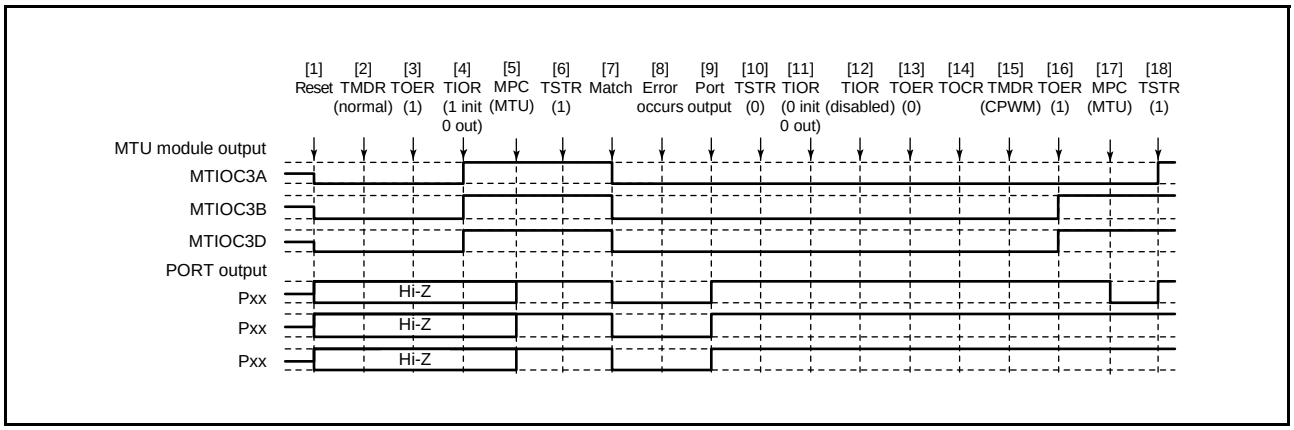


Figure 22.131 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.127.
[11] Initialize the normal mode waveform generation section with TIOR.
[12] Disable operation of the normal mode waveform generation section with TIOR.
[13] Disable output in MTU3 and MTU4 with TOER.
[14] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[15] Set complementary PWM mode.
[16] Enable output in MTU3 and MTU4 with TOER.
[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[18] Restart operation by setting TSTR.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.132 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

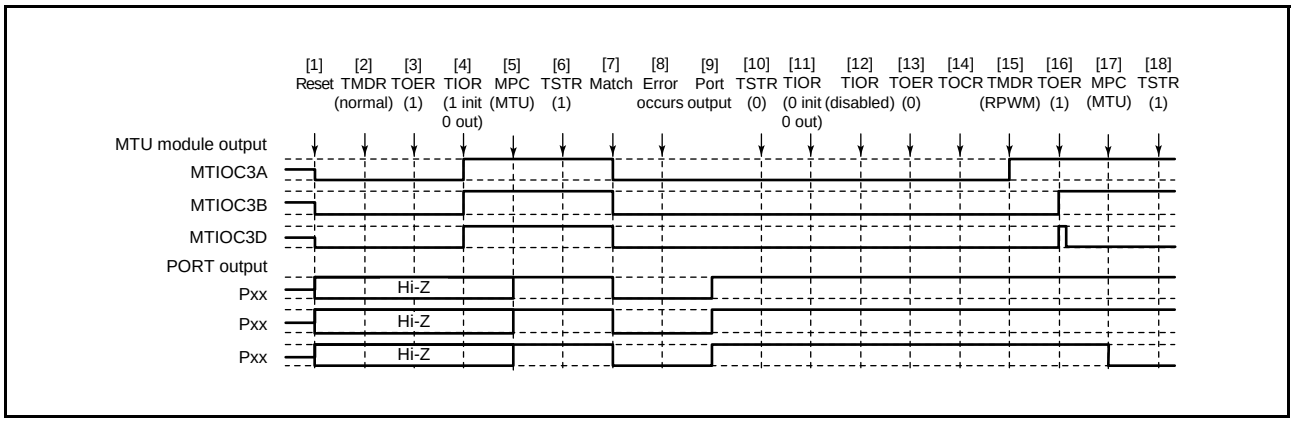


Figure 22.132 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [13] are the same as in Figure 22.127.

[14] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[15] Set reset-synchronized PWM mode.

[16] Enable output in MTU3 and MTU4 with TOER.

[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[18] Restart operation by setting TSTR.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 22.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

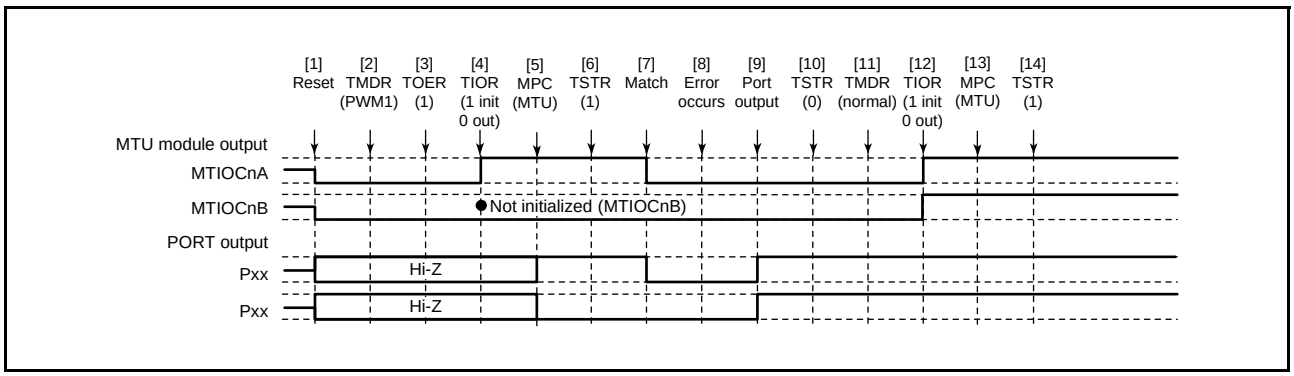


Figure 22.133 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

[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, enable output with TOER before initializing the pins with TIOR.
[4] Initialize the pins with TIOR. (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] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] Output goes low on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR.
[11] Set normal mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 22.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

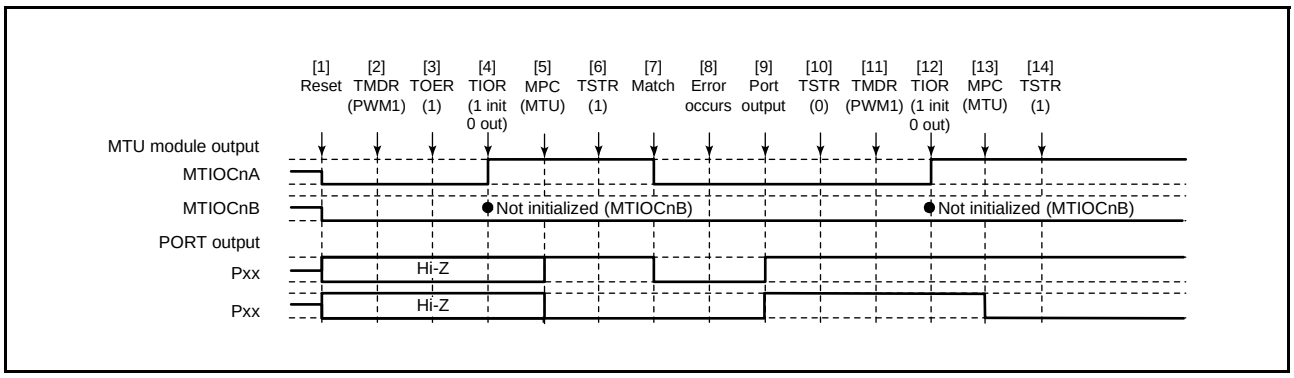


Figure 22.134 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.133.

[11] This step is not necessary when restarting in PWM mode 1.

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 22.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

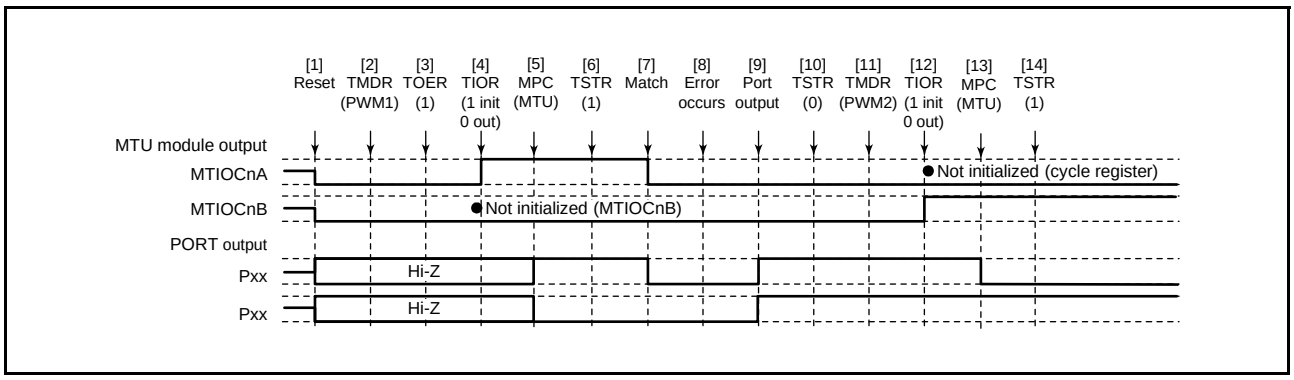


Figure 22.135 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 22.133.

[11] Set PWM mode 2.

[12] Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore, TOER setting is not necessary.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 22.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

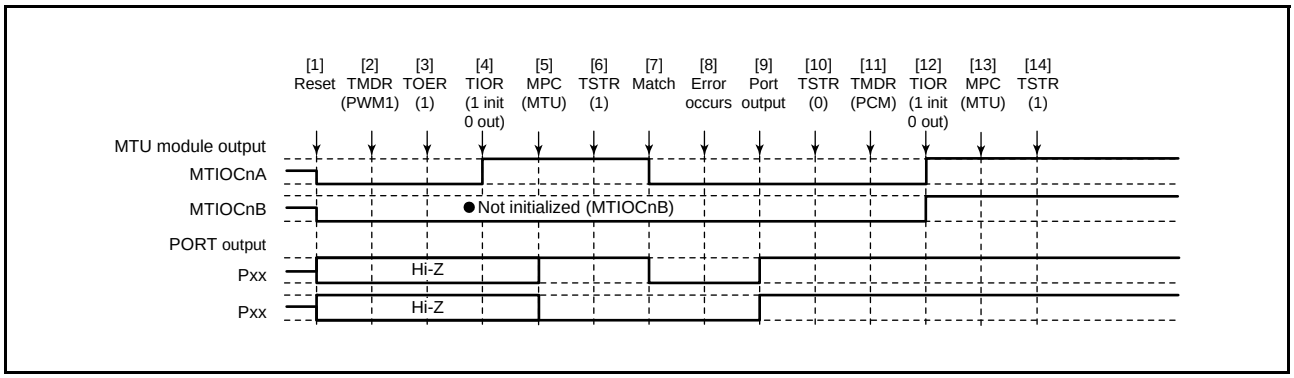


Figure 22.136 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 22.133.
[11] Set the phase counting mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 22.137 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

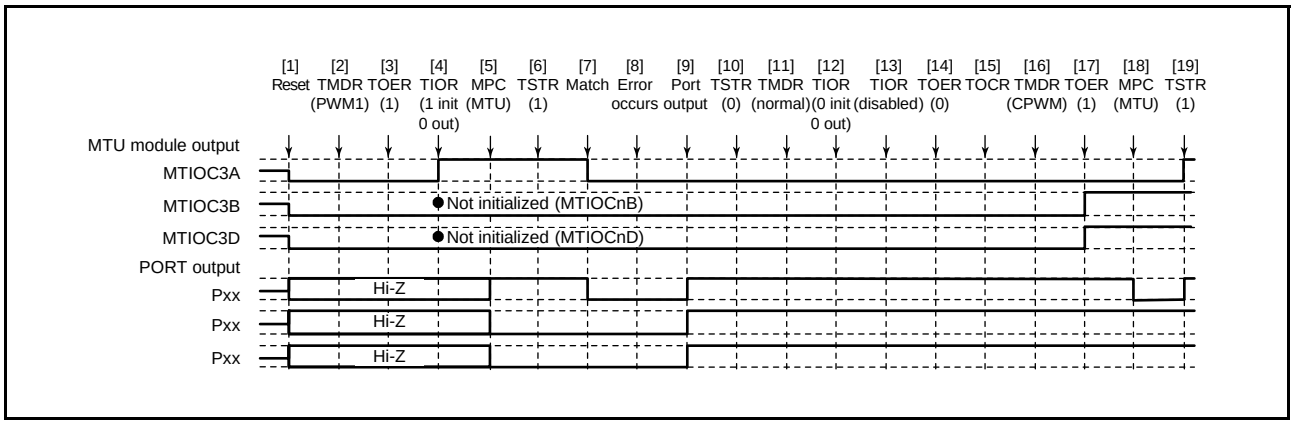


Figure 22.137 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.133.
[11] Set normal mode to initialize the normal mode waveform generation section.
[12] Initialize the PWM mode 1 waveform generation section with TIOR.
[13] Disable operation of the PWM mode 1 waveform generation section with TIOR
[14] Disable output in MTU3 and MTU4 with TOER.
[15] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[16] Set complementary PWM mode.
[17] Enable output in MTU3 and MTU4 with TOER.
[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[19] Restart operation by setting TSTR.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.138 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

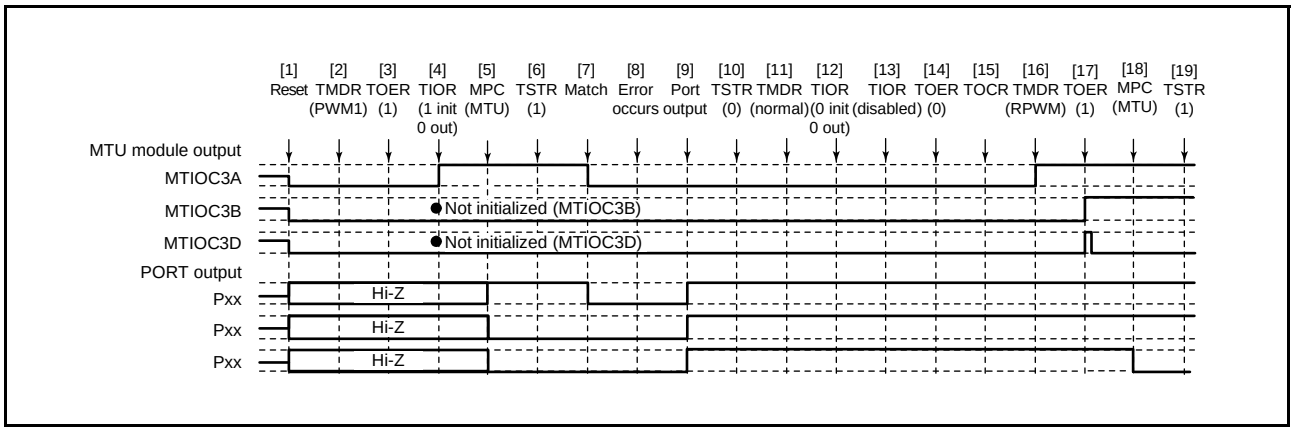


Figure 22.138 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

[1] to [14] are the same as in Figure 22.137.

[15] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[16] Set reset-synchronized PWM mode.

[17] Enable output in MTU3 and MTU4 with TOER.

[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[19] Restart operation by setting TSTR.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 22.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

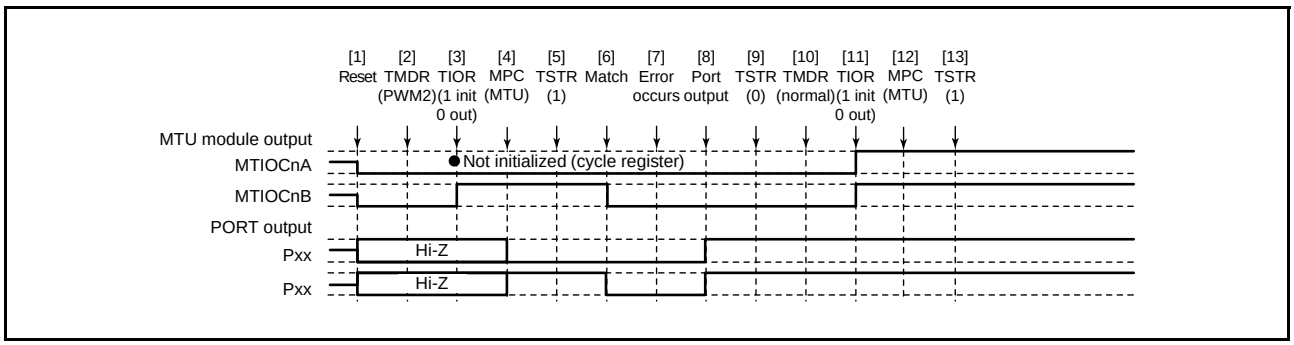


Figure 22.139 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

[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 TIOR. (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 cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)
[4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[5] Start count operation by setting TSTR.
[6] Output goes low on compare match occurrence.
[7] An error occurs.
[8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[9] Stop count operation by setting TSTR.
[10] Set normal mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 22.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

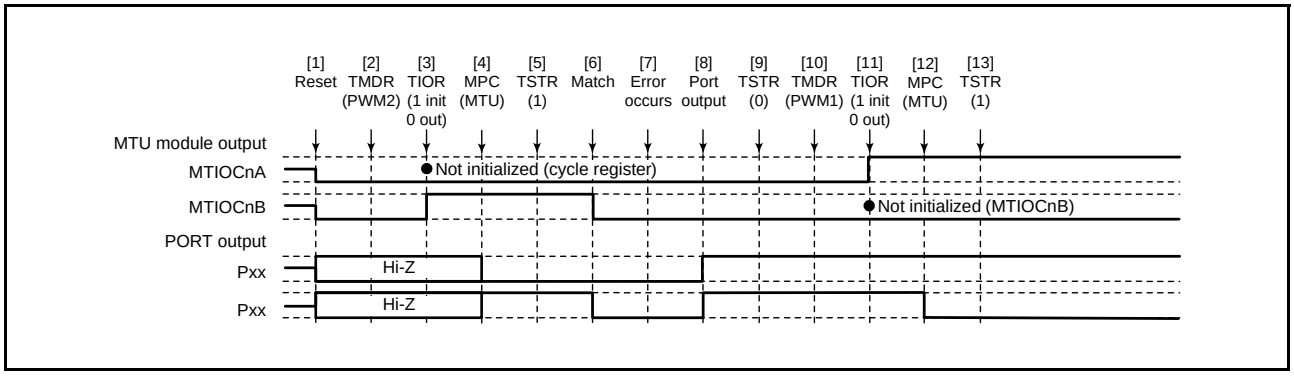


Figure 22.140 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 22.139.
[10] Set PWM mode 1.
[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 22.141 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

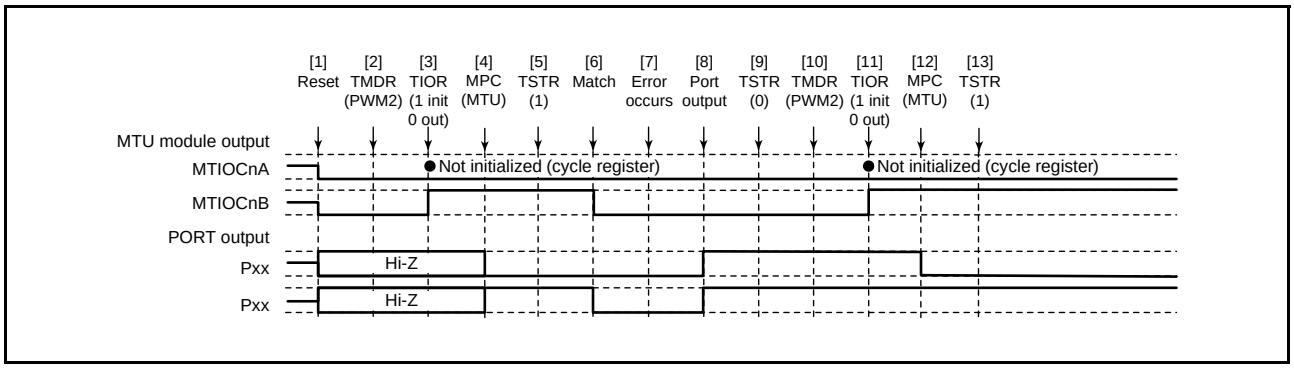


Figure 22.141 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 22.139.
[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 on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 22.142 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

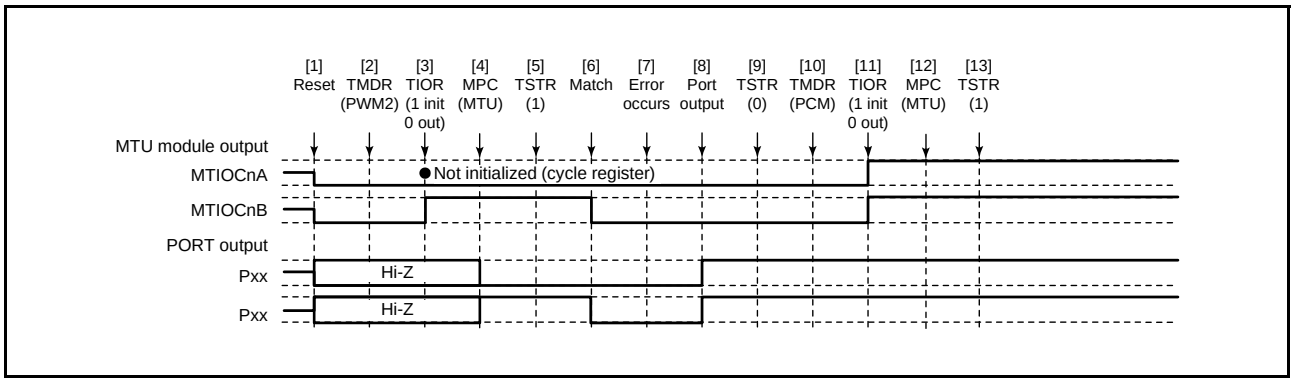


Figure 22.142 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 22.139.
[10] Set the phase counting mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 22.143 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

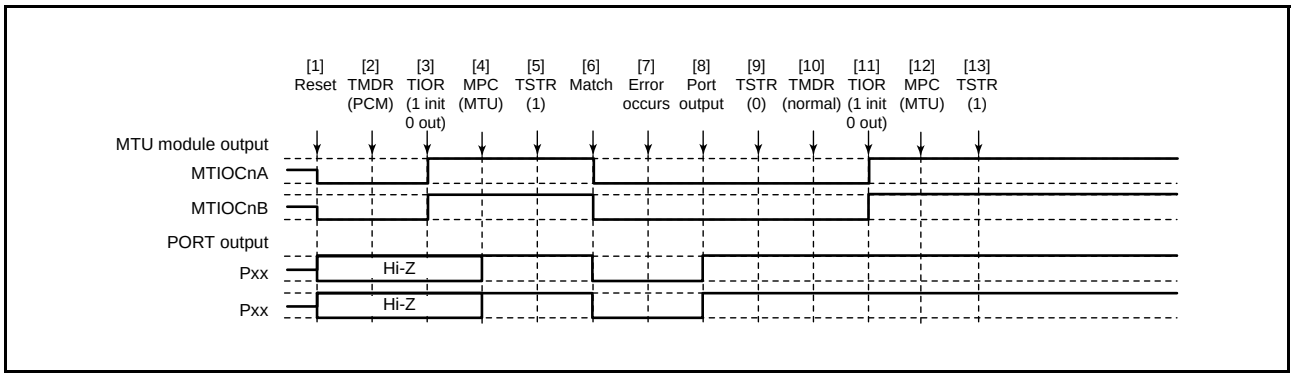


Figure 22.143 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

[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 TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
[4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[5] Start count operation by setting TSTR.
[6] Output goes low on compare match occurrence.
[7] An error occurs.
[8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[9] Stop count operation by setting TSTR.
[10] Set normal mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 22.144 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

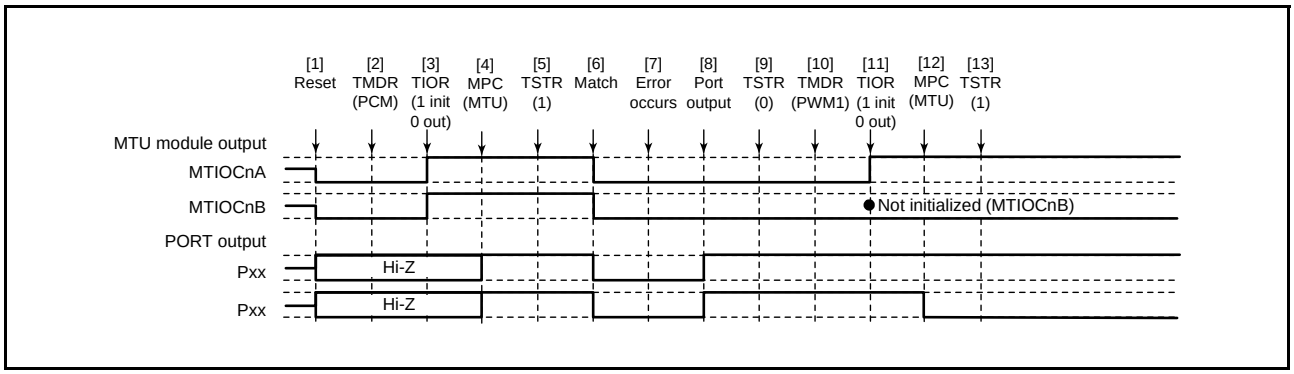


Figure 22.144 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 22.143.

[10] Set PWM mode 1.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 22.145 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

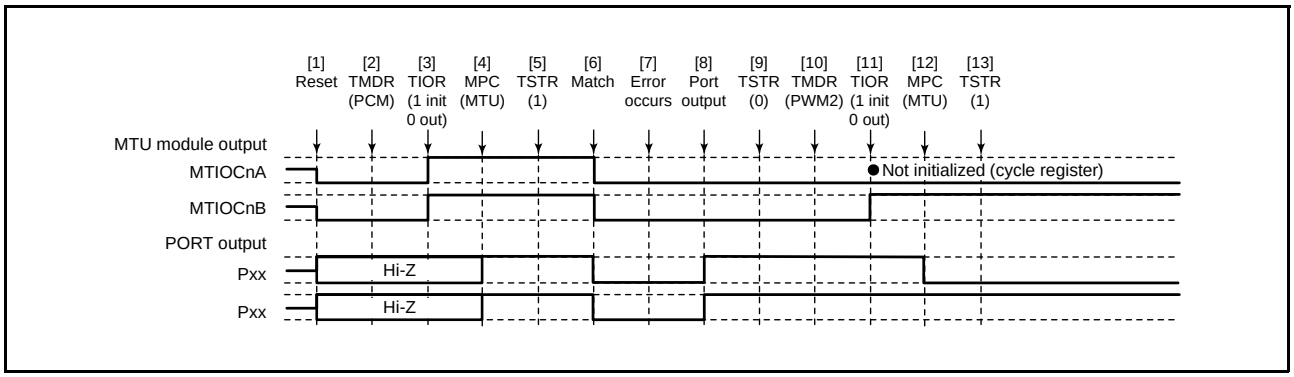


Figure 22.145 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 22.143.

[10] Set PWM mode 2.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 22.146 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

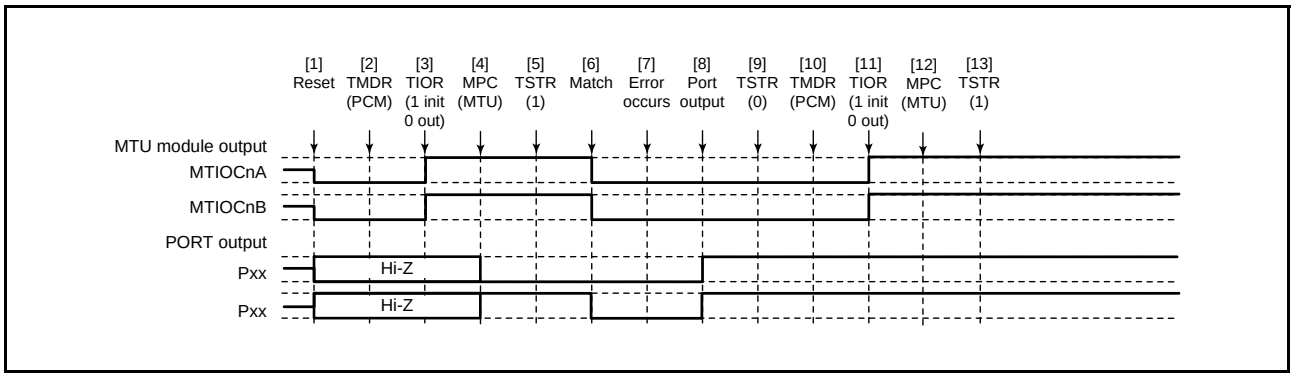


Figure 22.146 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 22.143.
[10] This step is not necessary when restarting in phase counting mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 22.147 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

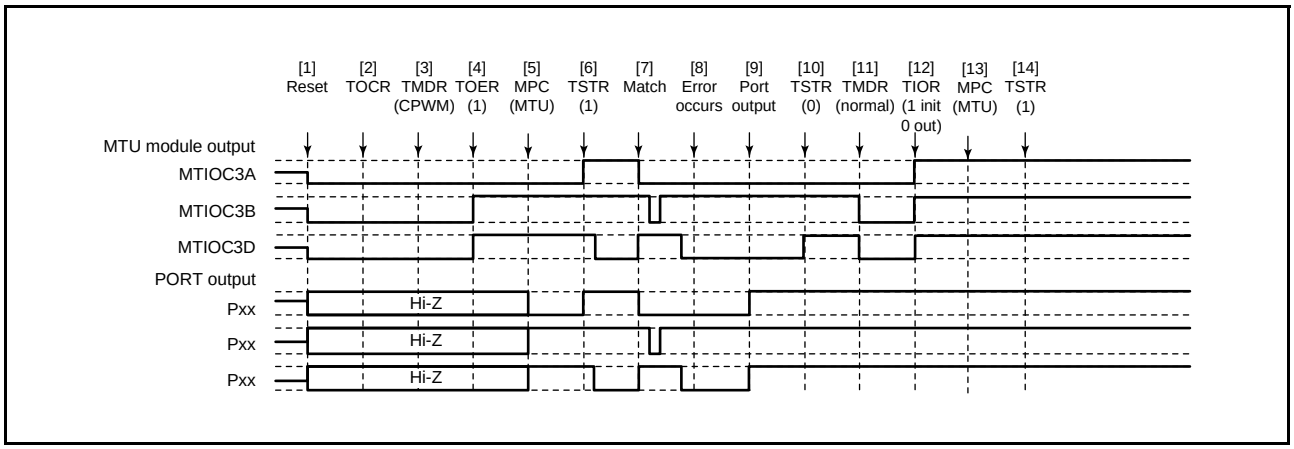


Figure 22.147 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[3] Set complementary PWM mode.
[4] Enable output in MTU3 and MTU4 with TOER.
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] The complementary PWM waveform is output on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR. (MTU output becomes the initial complementary PWM output value).
[11] Set normal mode (MTU output goes low).
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.148 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

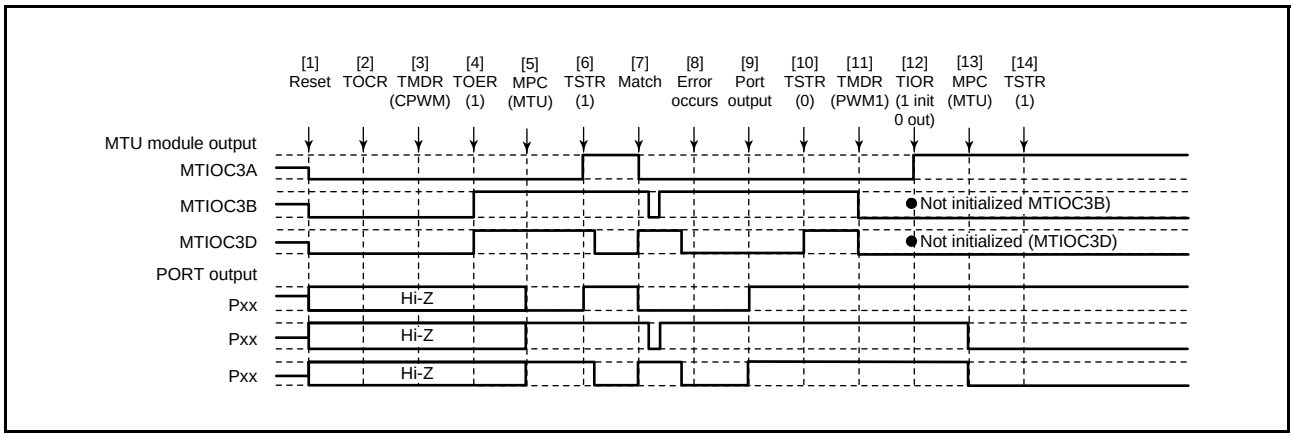


Figure 22.148 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.147.

[11] Set PWM mode 1 (MTU output goes low).

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.149 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 cycle and duty settings at the time of stopping the counter).

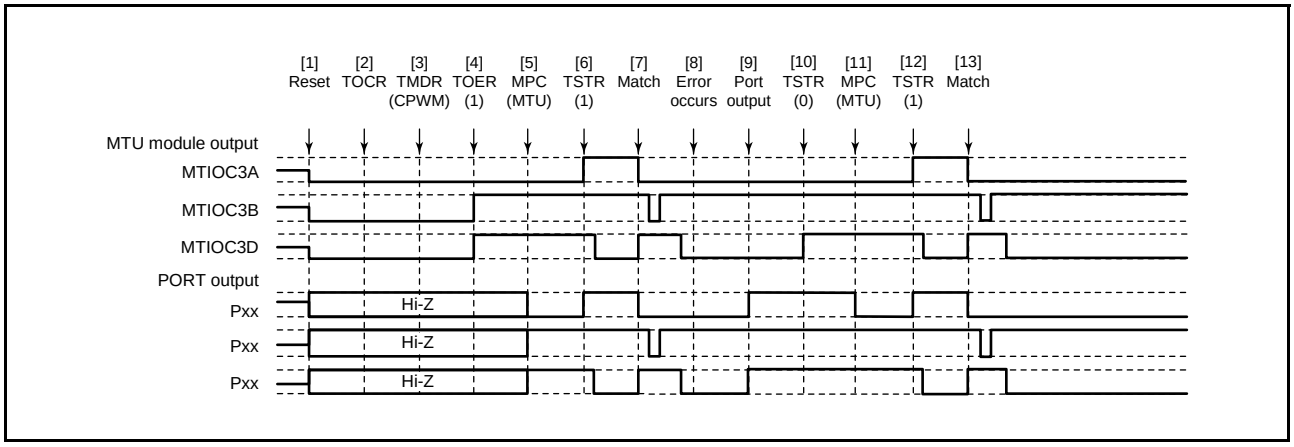


Figure 22.149 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.147.

[11] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[12] Restart operation by setting TSTR.

[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 22.150 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 cycle and duty settings).

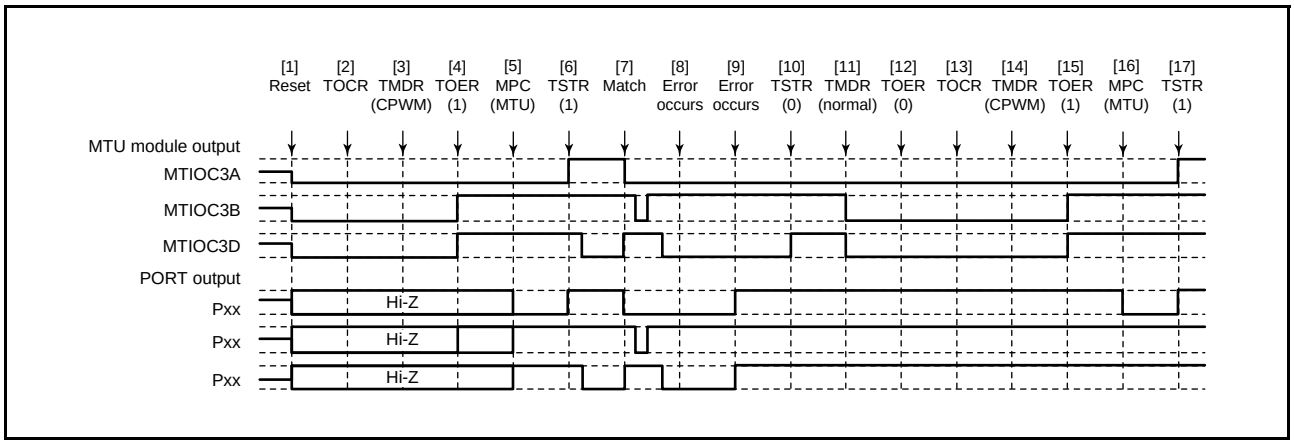


Figure 22.150 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.147.
[11] Set normal mode and make new settings (MTU output goes low).
[12] Disable output in MTU3 and MTU4 with TOER.
[13] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[14] Set complementary PWM mode.
[15] Enable output in MTU3 and MTU4 with TOER.
[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[17] Restart operation by setting TSTR.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.151 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

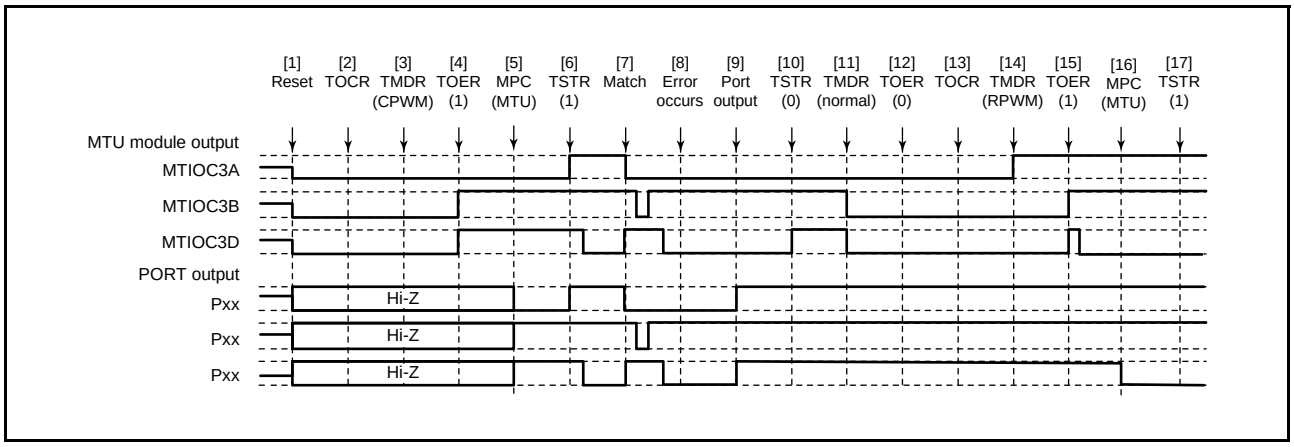


Figure 22.151 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [10] are the same as in Figure 22.147.
[11] Set normal mode (MTU output goes low).
[12] Disable output in MTU3 and MTU4 with TOER.
[13] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
[14] Set reset-synchronized PWM mode.
[15] Enable output in MTU3 and MTU4 with TOER.
[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[17] Restart operation by setting TSTR.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 22.152 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

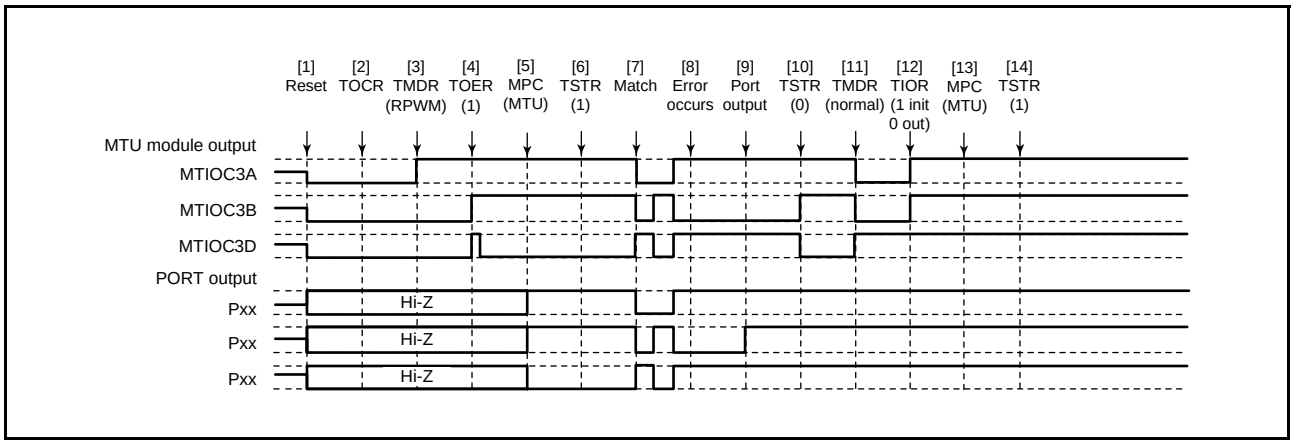


Figure 22.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

[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 TOCR.
[3] Set reset-synchronized PWM mode.
[4] Enable output in MTU3 and MTU4 with TOER.
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] The reset-synchronized PWM waveform is output on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR. (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 TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.153 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

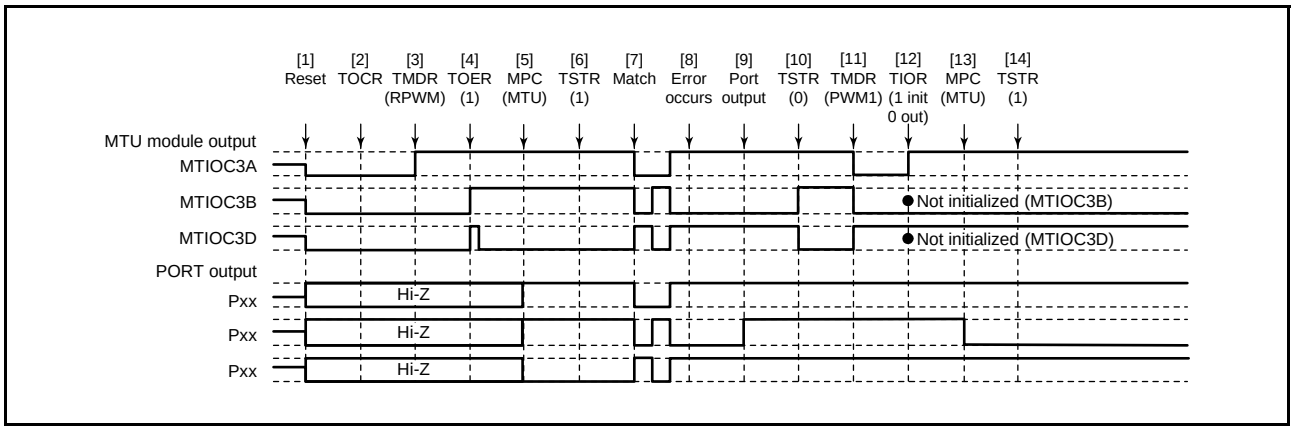


Figure 22.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.152.

[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, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.154 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

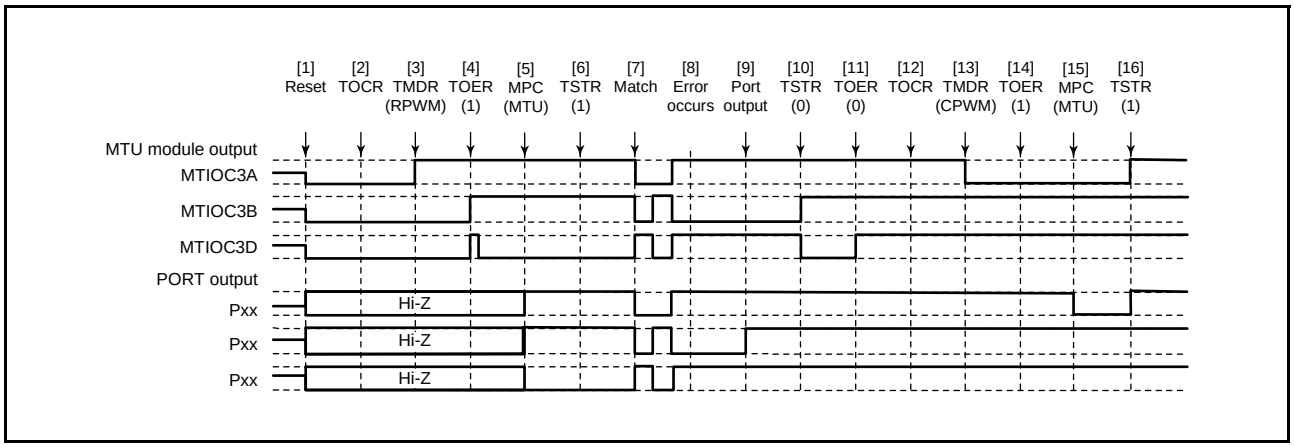


Figure 22.154 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.152.
[11] Disable output in MTU3 and MTU4 with TOER.
[12] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[13] Set complementary PWM mode (MTU cyclic output pin goes low).
[14] Enable output in MTU3 and MTU4 with TOER.
[15] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[16] Restart operation by setting TSTR.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.155 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.

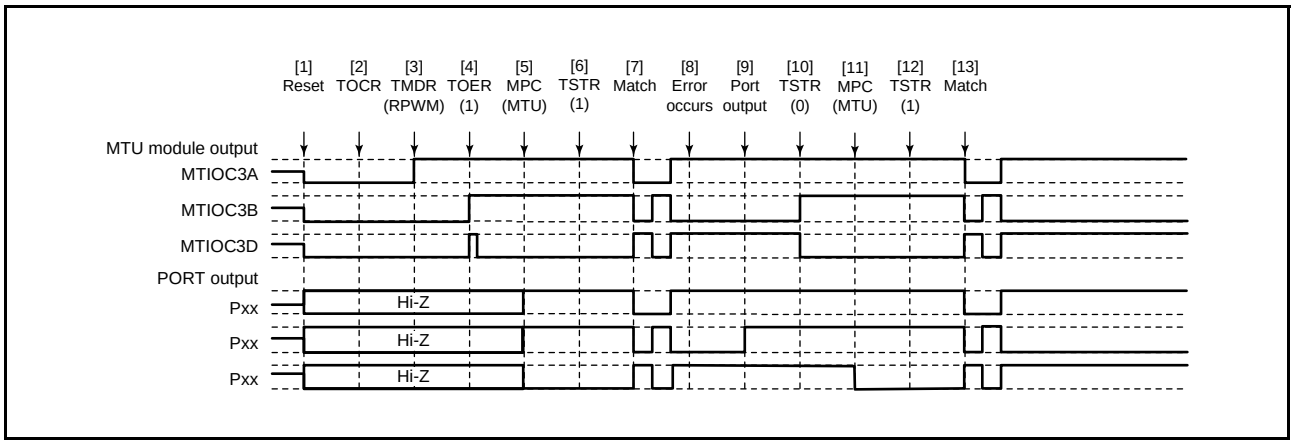


Figure 22.155 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [10] are the same as in Figure 22.152.

[11] Make MPC settings and port mode register (PMR) settings for the I/O port pins to operate as MTU outputs.

[12] Use the TSTR for a restart.

[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) Count Start Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions to channels 1 to 3, and ELOPB register functions to channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, the TSTR.CSTn bit shown in Table 22.60 is set to 1, then the MTU counter is started.
However, when the specified event is generated while TSTR.CSTn bit is set to 1, the event is disabled. Table 22.60 lists the TSTR register bits used for each channel.
For details on the count start operation setting, refer to section 22.3.1, (1) Counter Operation.

Table 22.60 Linkage Operating TSTR Register by the ELC

Channel No.	TSTR Register
MTU1	TSTR.CST1 bit
MTU2	TSTR.CST2 bit
MTU3	TSTR.CST3 bit
MTU4	TSTR.CST4 bit

(2) Input Capture Operation

The MTU is selected the input capture operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register handles channels 1 to 3, and ELOPB register handles channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT counter value capture to TGR register. When using the input capture operation, after setting the bit of MTU TIOR register to the input capture, TSTR.CSTn bit should be set to 1, and start the counter.
Then, the TIOCnA pin (input capture pin) input is disabled.
Table 22.61 lists the timer general register and timer I/O control register used in the input capture operation by the ELC.
For details on the input capture setting, refer to section 22.3.1, (3) Input Capture Function.

Table 22.61 Timer General Register and Timer I/O Control Register Used in the Input Capture Operation by the ELC

Channel No.	Register Name	Bit Name of TIOR Register
MTU1	TGRA register	TIOR.IOA[3:0] bits
MTU2	TGRA register	TIOR.IOA[3:0] bits
MTU3	TGRA register	TIORH.IOA[3:0] bits
MTU4	TGRA register	TIORH.IOA[3:0] bits

(3) Counter Restart Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions channels 1 to 3, and ELOPB register functions channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT (timer counter register) value is rewritten to initial value. When the CSTn bit in the TSTR register is 1, count operation can be continued. For details on the TSTR.CSTn bit, see Table 22.60.

22.8.3 Notes on MTU by Event Signal Reception from the ELC

The following describes usage notes when using MTU by the event link operation.

(1) Count Start Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TSTR.CSTn bit, the write cycle is not performed to the TSTR.CSTn bit, and the setting to 1 takes precedence by generated event.

(2) Count Restart Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TCNT counter, the write cycle is not performed to the TCNT counter, and count value initialization takes precedence by generated event.

23. Port Output Enable 2 (POE2a)

The port output enable 2 (POE) module can be used to place the states of the pins for complementary PWM output by the MTU (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D), and the states of pins for MTU0 (MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D) in the high-impedance in response to changes in the input levels on the POE0# to POE3# and POE8# pins, in the output levels on pins for complementary PWM output by the MTU, oscillation stop detection by the clock generation circuit, and changes to register settings (SPOER) or event signal input from the event link controller (ELC).
It can also generate simultaneous interrupt requests.
In this section, “PCLK” is used to refer to PCLKB.

23.1 Overview

Table 23.1 lists the specifications of the POE, and Figure 23.1 shows a block diagram of the POE.

Table 23.1 POE Specifications

Item	Description
High-impedance is controlled by the input level detection	• Falling-edge detection or sampling of the low level 16 times at PCLK/8, PCLK/16, or PCLK/128 clock cycles can be set for each of the POE0# to POE3# and POE8# input pins. • Pins for complementary PWM output from the MTU can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE0# to POE3# pins. • Pins for output from MTU0 can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE8# pin.
High-impedance is controlled by the output level comparison	• Levels output on pins for complementary PWM output from the MTU are compared, and when simultaneous output of the active level continues for one or more clock cycles, the pins can be placed in the high-impedance.
High-impedance is controlled by the oscillation stop detection	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance when oscillation by the clock generation circuit stops.
High-impedance is controlled by software (registers)	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance by modifying settings of POE registers.
High-impedance is controlled by the event signal	• Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance in response to an event signal from the event link controller (ELC).
Interrupts	• Interrupts can be generated in response to the results of POE0# to POE3# and POE8# input-level detection and MTU complementary PWM output-level comparison.

The POE has input-level detection circuits, output-level comparison circuits, an input for the oscillation stop detection signal from the clock generation circuit, and a high-impedance request/interrupt request generating circuit as shown in Figure 23.1.

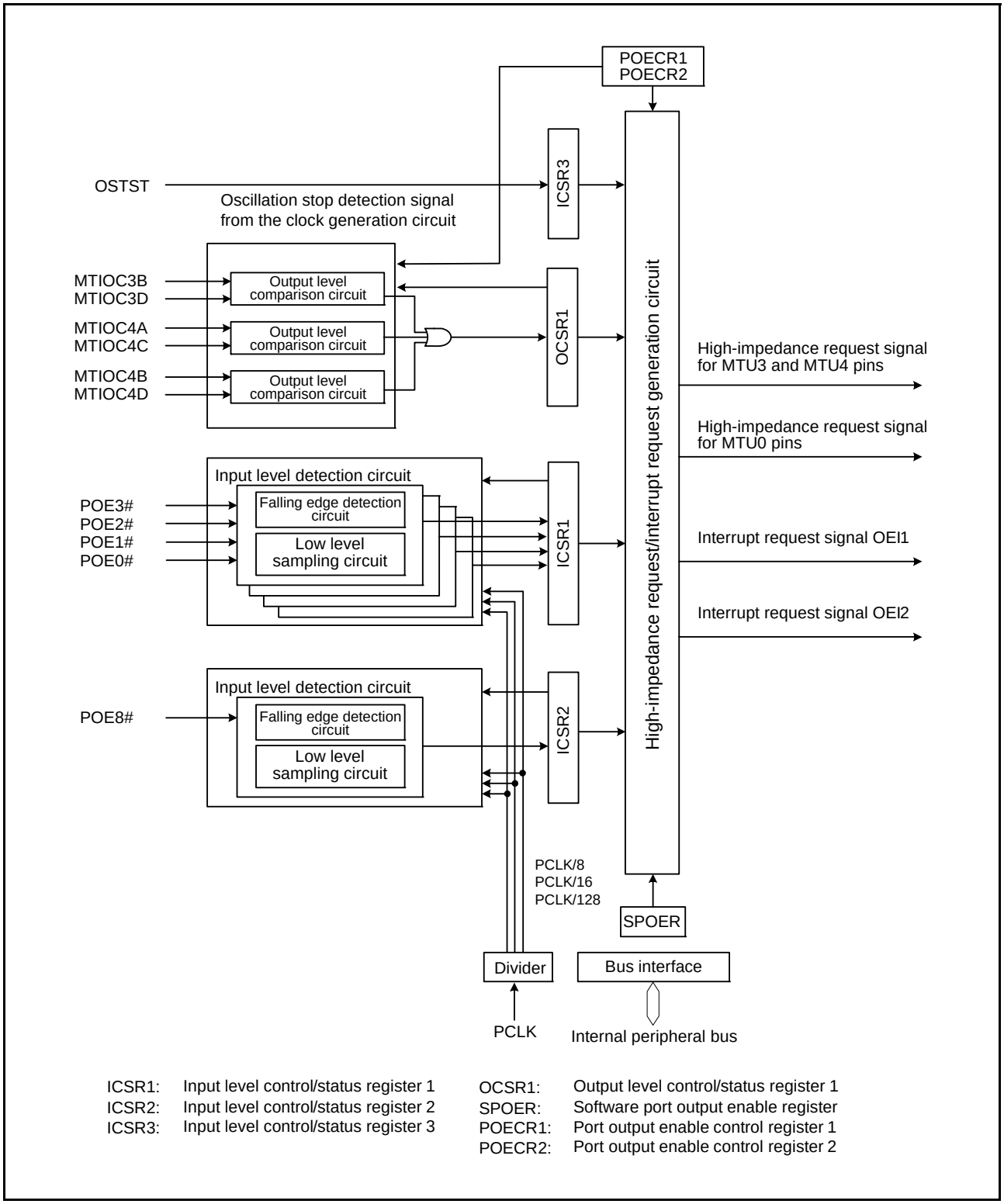


Figure 23.1 POE Block Diagram

Table 23.2 lists I/O pins to be used by the POE.

Table 23.2 POE I/O Pins

Pin Name	I/O	Description
POE0# to POE3#	Input	Request signals to place the pins for MTU complementary PWM output in high-impedance.
POE8#	Input	Request signal to place the MTU0 output pins in high-impedance.
MTIOC3B	Output	MTU3 complementary PWM output pin
MTIOC3D	Output	MTU3 complementary PWM output pin
MTIOC4A	Output	MTU4 complementary PWM output pin
MTIOC4B	Output	MTU4 complementary PWM output pin
MTIOC4C	Output	MTU4 complementary PWM output pin
MTIOC4D	Output	MTU4 complementary PWM output pin
MTIOC0A	Output	MTU0 output pin
MTIOC0B	Output	MTU0 output pin
MTIOC0C	Output	MTU0 output pin
MTIOC0D	Output	MTU0 output pin

Table 23.3 lists output-level comparisons with pin combinations.

Table 23.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	Pin combinations for output-level comparison and high-impedance control can be selected by POE registers. The pins for MTU complementary PWM output are placed in high-impedance when the pins simultaneously output an active level for one or more PCLK clock cycles. (When the MTU.TOCR1.TOCS bit = 0: The active level is low level if the MTU.TOCR1.OLSP and OLSN bits are 0, and the active level is high level if the MTU.TOCR1.OLSP and OLSN bits are 1. When the MTU.TOCR1.TOCS bit = 1: The active level is low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0, and the active level is high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 1.)
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

23.2 Register Descriptions

23.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 8900h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1	POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[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	POE0M[1:0]	POE0 Mode Select	b1 b0 0 0: Accepts a high-impedance request on the falling edge of the POE0# pin input. 0 1: Accepts a high-impedance request when the POE0# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b3, b2	POE1M[1:0]	POE1 Mode Select	b3 b2 0 0: Accepts a high-impedance request on the falling edge of the POE1# pin input. 0 1: Accepts a high-impedance request when the POE1# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b5, b4	POE2M[1:0]	POE2 Mode Select	b5 b4 0 0: Accepts a high-impedance request on the falling edge of the POE2# pin input. 0 1: Accepts a high-impedance request when the POE2# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE2# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7, b6	POE3M[1:0]	POE3 Mode Select	b7 b6 0 0: Accepts a high-impedance request on the falling edge of the POE3# pin input. 0 1: Accepts a high-impedance request when the POE3# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: OEI1 interrupt requests by the input level detection disabled 1: OEI1 interrupt requests by the input level detection 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 a high-impedance request has not been input to the POE0# pin. 1: Indicates that a high-impedance request has been input to the POE0# pin.	R/(W)*2
b13	POE1F	POE1 Flag	0: Indicates that a high-impedance request has not been input to the POE1# pin. 1: Indicates that a high-impedance request has been input to the POE1# pin.	R/(W)*2
b14	POE2F	POE2 Flag	0: Indicates that a high-impedance request has not been input to the POE2# pin. 1: Indicates that a high-impedance request has been input to the POE2# pin.	R/(W)*2
b15	POE3F	POE3 Flag	0: Indicates that a high-impedance request has not been input to the POE3# pin. 1: Indicates that a high-impedance request has been input to the POE3# pin.	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.

When low-level sampling has been set by the POE0M[1:0] to POE3M[1:0] bits, writing 0 to the POE0F to POE3F flags requires high-level input on the POE0# to POE3# pins.
For details, refer to section 23.3.6, Release from the High-Impedance.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when any one of the POE0F to POE3F flags is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.
[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

POE1F Flag (POE1 Flag)

This flag indicates that a high-impedance request has been input to the POE1# pin.
[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

POE2F Flag (POE2 Flag)

This flag indicates that a high-impedance request has been input to the POE2# pin.
[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

POE3F Flag (POE3 Flag)

This flag indicates that a high-impedance request has been input to the POE3# pin.
[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

23.2.2 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 8902h

	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	Output Short Interrupt Enable 1	0: OEI1 interrupt requests by the output level comparison disabled 1: OEI1 interrupt requests by the output level comparison enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Output Short 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.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the MTU complementary PWM output pins in high-impedance when the OSF1 flag is set to 1.

OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase outputs for MTU complementary PWM output to be compared in Table 23.3 has simultaneously become an active level. If the POECR2.PnCZEA (n = 1, 2, 3) bits are 0 or the output comparison function of the MTU is not enabled, the OSF1 flag will not be set to 1 even if both pins in the corresponding complementary output pair of the MTU are simultaneously active.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1
  The complementary output pins for the MTU must be at the inactive level when 0 is written to the flag.
  For details, refer to section 23.3.6, Release from the High-Impedance.

23.2.3 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 8908h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE2	—	—	—	—	—	—	POE8M[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	POE8M[1:0]	POE8 Mode Select	b1 b0 0 0: Accepts a high-impedance request on the falling edge of the POE8# pin input 0 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/16 clock cycles and all are low level. 1 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/128 clock cycles and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE2	Port Interrupt Enable 2	0: OEI2 interrupt requests disabled 1: OEI2 interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance.	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 a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	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.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables OEI2 interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in high-impedance when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.
[Setting condition]

- When the input set by ICSR2.POE8M[1:0] bits occurs at the POE8# pin

[Clearing conditions]

- Writing 0 to POE8F after reading POE8F = 1
  When writing 0 to the flag while low-level sampling is selected for the ICSR2.POE8M[1:0] bits, the POE8# pin input must be at the high level.
  For details, refer to section 23.3.6, Release from the High-Impedance.

23.2.4 Software Port Output Enable Register (SPOER)

Address(es): 0008 890Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CH0HIZ	CH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH34HIZ	MTU3 and MTU4 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b1	CH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit selects whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Setting conditions]

- By writing 1 to CH34HIZ
- An event signal from the event link controller (ELC) is received.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit selects whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Setting conditions]

- By writing 1 to CH0HIZ
- An event signal from the event link controller (ELC) is received.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

23.2.5 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 890Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PE3ZE	PE2ZE	PE1ZE	PE0ZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PE0ZE	MTIOC0A High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b1	PE1ZE	MTIOC0B High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b2	PE2ZE	MTIOC0C High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b3	PE3ZE	MTIOC0D High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	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.

23.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	P1CZEA	P2CZEA	P3CZEA	—	—	—	—
Value after reset:	0	1	1	1	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	P3CZEA	MTU Port 3 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b5	P2CZEA	MTU Port 2 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b6	P1CZEA	MTU Port 1 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

When this function is not used, write 00h to this register.

P3CZEA Bit (MTU Port 3 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4B and MTIOC4D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4B and MTIOC4D pins are compared.

P2CZEA Bit (MTU Port 2 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4A and MTIOC4C pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4A and MTIOC4C pins are compared.

P1CZEA Bit (MTU Port 1 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC3B and MTIOC3D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC3B and MTIOC3D pins are compared.

23.2.7 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 890Eh

	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	OSTST High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	OSTST High-Impedance Flag	0: Oscillation stop is not producing a request to place pins in the high-impedance. 1: Oscillation stop is producing a request to place pins in the high-impedance.	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.

OSTSTE Bit (OSTST High-Impedance Enable)

This bit permits or prohibits placement of pins for complementary PWM output from MTU and output pins for MTU0 in the high-impedance on detection that oscillation has stopped.

OSTSTF Flag (OSTST High-Impedance Flag)

The OSTSTF flag is a status flag that indicates the state of requests to place pins in the high-impedance due to oscillation having stopped. The value of the flag becomes 1 when oscillation stops. Ensure that the oscillation-stopped detection signal is negated when clearing the flag by writing 0 to it. Writing 0 to the OSTSTF flag will not clear the flag while the oscillation-stopped detection signal is being asserted; in other words, it will not clear the flag before 10 PCLK clock cycles have elapsed after stopped oscillation was detected.

[Setting condition]

- Detection of the oscillation-stopped state

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

23.3 Operation

The target pins for high-impedance control and conditions to place the pins in high-impedance are described below.

(1) MTU0 pin (MTIOC0A)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(2) MTU0 pin (MTIOC0B)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(3) MTU0 pin (MTIOC0C)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(4) MTU0 pin (MTIOC0D)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When the ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation
  When the OSTSTF flag is set to 1 with POECR1.PE3ZE and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(5) MTU3 pins (MTIOC3B and MTIOC3D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P1CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(6) MTU4 pins (MTIOC4A and MTIOC4C)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P2CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P2CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

(7) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

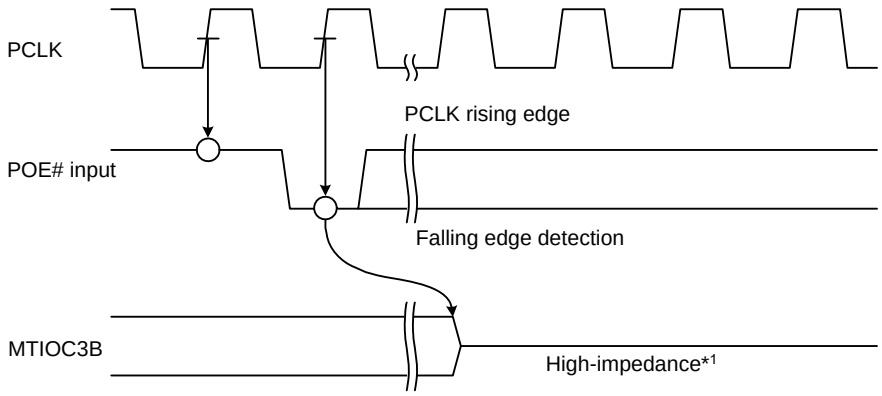
- POE0# to POE3# input level detection
  When the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison
  When the OCSR1.OSF1 flag is set to 1 with POECR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When the SPOER.CH34HIZ bit is set to 1 with POECR2.P3CZEA set to 1.
- Detection of stopped oscillation
  When the ICSR3.OSTSTF flag is set to 1 with POECR2.P3CZEA and ICSR3.OSTSTE set to 1.
- Event signal reception from the ELC

23.3.1 Input Level Detection Operation

If the input conditions set by the ICSR1 and ICSR2 registers occur on the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.
A falling edge is detected after PCLK causes sampling to proceed. If the low level is input to the POE0# to POE3# or POE8# pin over less than one PCLK cycle, whether the falling edge will or will not be detected cannot be guaranteed.
Figure 23.2 shows the timing of sampling after the level changes in input to the POE0# to POE3# and POE8# pins until the respective pins enter high-impedance.

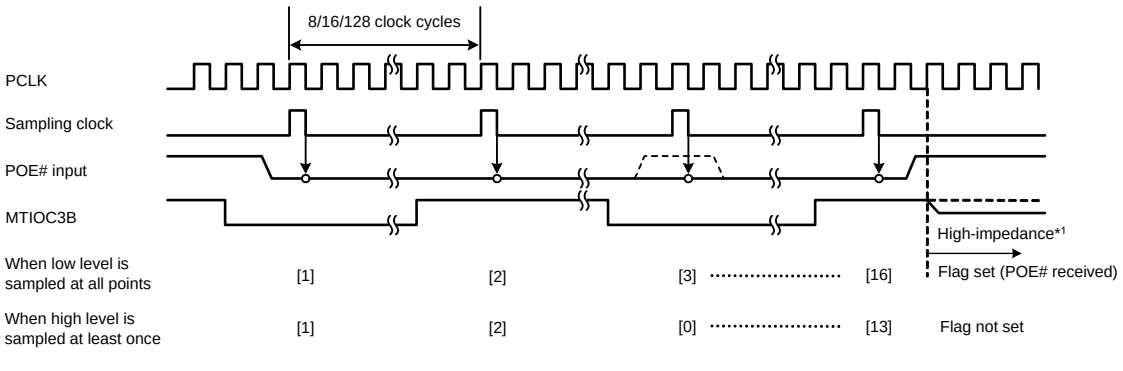


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 23.2 Falling Edge Detection

(2) Low-Level Detection

Figure 23.3 shows the low-level detection operation. When a low level is detected 16 times continuously with the sampling clock selected by the ICSR1 and ICSR2 registers, the detected level is recognized as low, and the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance. If even one high level is detected during this interval, the detected level is not recognized as low. Furthermore, in an interval over which the sampling clock is not being output, changes to the levels on the POE0# to POE3# and POE8# pins are ignored.

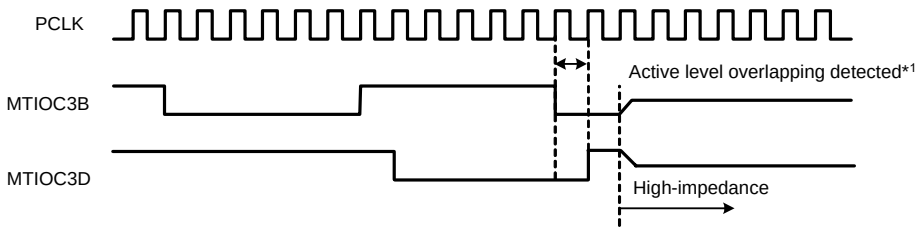


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 23.3 Low-Level Detection Operation

23.3.2 Output-Level Compare Operation

Figure 23.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D (MTU complementary PWM output pins). The operation is the same for the other pin combinations.



Note 1. In this case, the low level has been set as the active level for the MTIOC3B and MTIOC3D pins.

Figure 23.4 Output-Level Compare Operation

23.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled by writing to the software port output enable register (SPOER).
Setting the SPOER.CH34HIZ bit to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the POECR2 register in the high-impedance.
Setting the SPOER.CH0HIZ bit to 1 places the MTU0 output pins specified by port output enable control register 1 (POECR1) in the high-impedance.

23.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while the ICSR3.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register and the MTU0 output pins specified by the POECR1 register are placed in the high-impedance.

23.3.5 High-Impedance Control in Response to Receiving an Event Signal from the ELC

The MTU complementary PWM output and MTU0 pins can be placed in the high-impedance state in response to an event signal from the ELC.
To control the high-impedance state of the MTU complementary PWM output and MTU0 pins, preset the corresponding register (POECR1 or POECR2) to enable the high-impedance state. When an event signal is received from the ELC, the corresponding bit (SPOER.CH0HIZ or SPOER.CH34HIZ) is set to 1, and the MTU complementary PWM output pins or MTU0 pins are placed in the high-impedance state.

23.3.6 Release from the High-Impedance

Pins for complementary PWM output from MTU and pins for MTU0 which have been placed in the high-impedance due to input-level detection can be released from that state by either returning them to their initial state with a reset or clearing all of the ICSR1.POE3F to POE0F flags and the ICSR2.POE8F flag. Note, however, that when low-level sampling is selected by the ICSR1.POE3M[1:0], POE2M[1:0], POE1M[1:0], and POE0M[1:0] bits, and the ICSR2.POE8M[1:0] bits, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE#8 but has not yet been detected, writing 0 to the flag is ignored (the flag is not cleared).
MTU complementary PWM output pins which have been placed in the high-impedance due to output-level comparison can be released from that state by either returning them to their initial state with a reset or clearing the OCSR1.OSF1 flag. Note, however, that if the inactive level is not yet being output from the MTU complementary PWM output pins, writing 0 to the flag is ignored (the flag is not cleared). Inactive-level outputs can be obtained by setting the MTU registers.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance because oscillation by the clock generation circuit has stopped, clearing the ICSR3.OSTSTF or ICSR3.OSTSTE bit releases the pins from the high-impedance.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance by the SPOER.CH34HIZ or SPOER.CH0HIZ bit, clearing the corresponding bits (SPOER.CH34HIZ and SPOER.CH0HIZ) releases the pins from the high-impedance.

23.4 Interrupts

The POE issues a request to generate an interrupt when the corresponding condition below is matched during input-level detection, output-level comparison, or oscillation stop by the clock generation circuit. Table 23.4 lists the interrupt sources and their request conditions. On acceptance of an OEI1 or OEI2 interrupt, the first line of the exception handling routine for the given interrupt should confirm that the flag for the given flag has been set to 1.

Table 23.4 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F, POE1F, POE2F, POE3F, OSF1	When ICSR1.POE0F, POE1F, POE2F, or POE3F flag is set to 1 with ICSR1.PIE1 set to 1, or when OCSR1.OSF1 flag is set to 1 with OCSR1.OIE1 set to 1.
OEI2	Output enable interrupt 2	POE8F	When ICSR2.POE8F flag is set to 1 with ICSR2.PIE2 set to 1.

23.5 Usage Notes

23.5.1 Transitions to Software Standby Mode or Deep Software Standby Mode

When the POE is used, do not make a transition to software standby mode or deep software standby mode. In these modes, the POE stops and thus the high-impedance of pins cannot be controlled.

23.5.2 When the POE Is Not Used

When the POE is not used, write 00h to port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

23.5.3 Specifying Pins Corresponding to the MTU

The POE controls high-impedance outputs only when a pin has been specified so that the pin corresponds to the MTU by setting the PMR and PmnPFS registers. When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

23.5.4 Notes on High-Impedance Control by Event Signal Reception from the ELC

When writing 0 to the SPOER.CH34HIZ or SPOER.CH0HIZ bit and receiving an event signal conflict, the event signal takes priority and the corresponding bit is set to 1. If the MTU complementary PWM output and MTU0 pins are placed in the high-impedance state when an event signal is received from the ELC, no interrupt request is generated.

24. 16-Bit Timer Pulse Unit (TPUa)

This MCU has on-chip 16-bit timer pulse units (TPU) comprising six-channel 16-bit timers.
In this section, “PCLK” is used to refer to PCLKB.

24.1 Overview

Specifications of the TPU are listed in Table 24.1. Functions of TPU are listed in Table 24.2.
Figure 24.1 shows a block diagram of TPU.

Table 24.1 Specifications of TPU

Item	Description
Pulse input/output	Maximum 16
Count clocks	Seven or eight types are provided for each channel.
Settable operations	• Waveform output at compare match • Input capture function (noise filters can be set) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match and input capture • Synchronous input/output for registers by counter synchronous operation • Maximum of 15-phase PWM output by combination with synchronous operation • Cascaded operation
TPU0 and TPU3	Buffer operation can be set.
TPU1, TPU2, TPU4, and TPU5	Phase counting mode can be set.
Interrupt sources	26 sources
Buffer operation	Automatic transfer of register data
Generation of trigger	Conversion start trigger for the A/D converter can be generated.
Low power consumption function	Module stop state can be set for each unit.

Table 24.2 TPU Functions

Item		TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD
External clocks for phase counting mode		Not possible	TCLKA TCLKB	TCLKC TCLKD	Not possible	TCLKC TCLKD	TCLKA TCLKB
Timer general registers		TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible	Possible
Input capture function		Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation		Possible	Possible	Possible	Possible	Possible	Possible
PWM mode		Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode		Not possible	Possible	Possible	Not possible	Possible	Possible
Buffer operation		Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D conversion start trigger		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	Not possible
Interrupt sources		5 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 • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 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	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow
Module stop setting*2		MSTPCRA.MSTPA13 bit					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

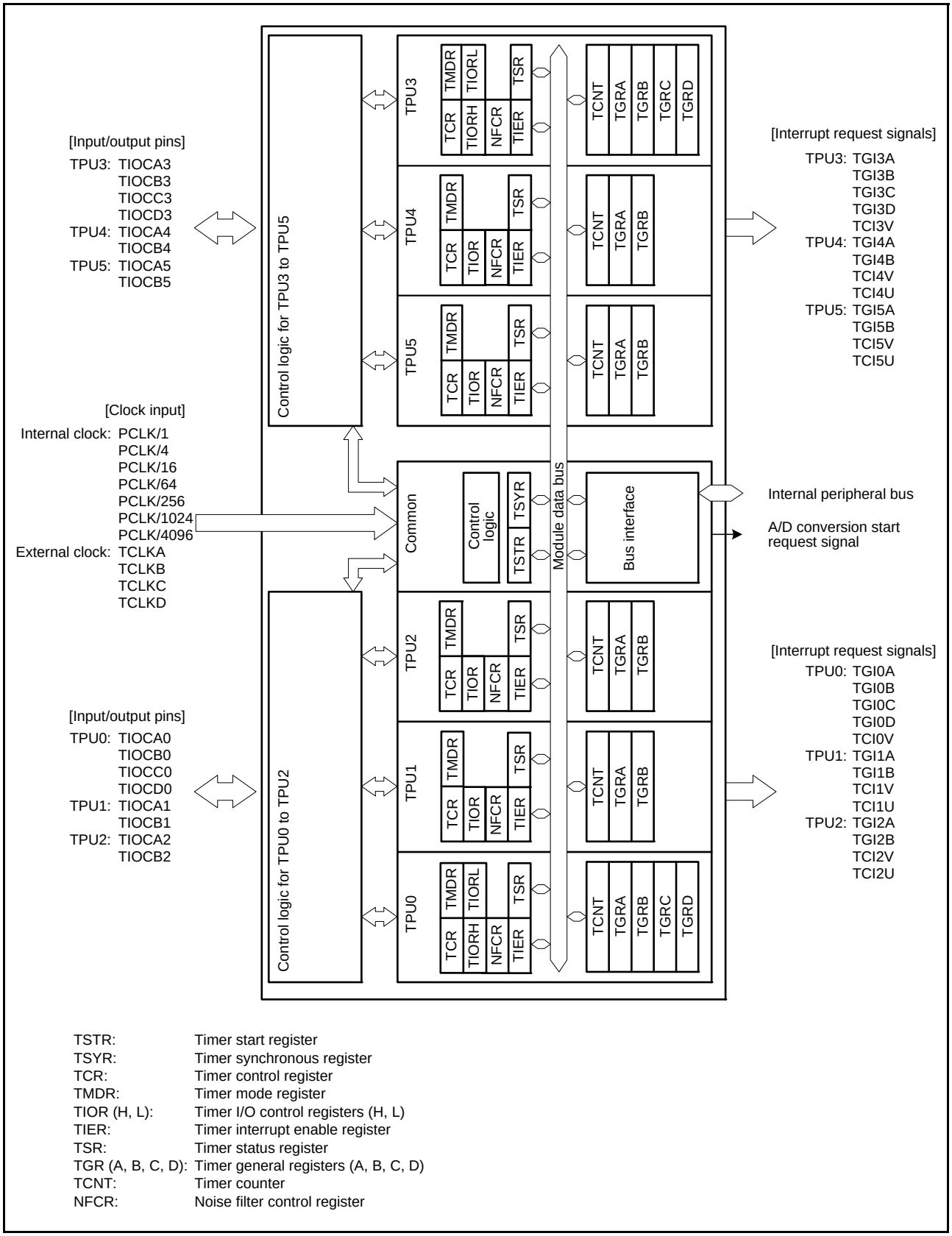


Figure 24.1 Block Diagram of TPU

Table 24.3 lists the input/output pins of the TPU.

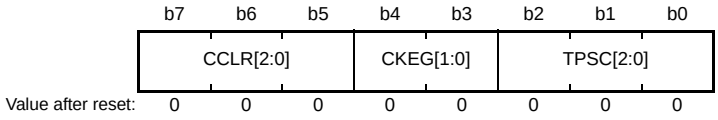
Table 24.3 Pin Configuration of TPU

Channel	Pin Name	I/O	Description
Common	TCLKA	Input	External clock A input pin (TPU1 and TPU5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (TPU1 and TPU5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (TPU2 and TPU4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (TPU2 and TPU4 phase counting mode B phase input)
TPU0	TIOCA0	I/O	TPU0.TGRA input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TPU0.TGRB input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TPU0.TGRC input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TPU0.TGRD input capture input/output compare output/PWM output pin
TPU1	TIOCA1	I/O	TPU1.TGRA input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TPU1.TGRB input capture input/output compare output/PWM output pin
TPU2	TIOCA2	I/O	TPU2.TGRA input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TPU2.TGRB input capture input/output compare output/PWM output pin
TPU3	TIOCA3	I/O	TPU3.TGRA input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TPU3.TGRB input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TPU3.TGRC input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TPU3.TGRD input capture input/output compare output/PWM output pin
TPU4	TIOCA4	I/O	TPU4.TGRA input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TPU4.TGRB input capture input/output compare output/PWM output pin
TPU5	TIOCA5	I/O	TPU5.TGRA input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TPU5.TGRB input capture input/output compare output/PWM output pin

24.2 Register Descriptions

24.2.1 Timer Control Register (TCR)

Address(es): TPU0.TCR 0008 8110h, TPU1.TCR 0008 8120h, TPU2.TCR 0008 8130h, TPU3.TCR 0008 8140h, TPU4.TCR 0008 8150h, TPU5.TCR 0008 8160h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 24.4 to Table 24.9.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 24.10.	R/W
b7 to b5	CCLR[2:0]*1	Counter Clear Source Select	See Table 24.11 and Table 24.12.	R/W

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT operation is stopped.

TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT clock. The clock source can be selected independently for each channel.
To select the external clock as the clock source, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 20, I/O Ports.

CKEG[1:0] Bits (Input Clock Edge Select)

These bits select the input clock edge.
When the internal clock is counted using both edges, the input clock period is halved (e.g. Both edges of PCLK/4 = PCLK/2 rising edge).
Internal clock edge selection is valid when the input clock is PCLK/4 or slower. This setting is ignored if the input clock is PCLK/1, or when overflow/underflow of another channel is selected.

Table 24.4 Bits TPSC[2:0] (TPU0)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU0: counts on TCLKA pin input
	1	0	1	External clock • TPU0: counts on TCLKB pin input
	1	1	0	External clock • TPU0: counts on TCLKC pin input
	1	1	1	External clock • TPU0: counts on TCLKD pin input

Table 24.5 Bits TPSC[2:0] (TPU1)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU1: counts on TCLKA pin input
	1	0	1	External clock • TPU1: counts on TCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	• TPU1 Counts on TPU2.TCNT overflow/underflow

Note: This setting is invalid when TPU1 is in phase counting mode.

Table 24.6 Bits TPSC[2:0] (TPU2)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU2: counts on TCLKA pin input
	1	0	1	External clock • TPU2: counts on TCLKB pin input
	1	1	0	External clock • TPU2: counts on TCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is invalid when TPU2 is in phase counting mode.

Table 24.7 Bits TPSC[2:0] (TPU3)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU3: counts on TCLKA pin input
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Internal clock: counts on PCLK/4096

Table 24.8 Bits TPSC[2:0] (TPU4)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU4: counts on TCLKA pin input
	1	0	1	External clock • TPU4: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	• TPU4 Counts on TPU5.TCNT overflow/underflow

Note: This setting is invalid when TPU4 is in phase counting mode.

Table 24.9 Bits TPSC[2:0] (TPU5)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock • TPU5: counts on TCLKA pin input
	1	0	1	External clock • TPU5: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock • TPU5: counts on TCLKD pin input

Note: This setting is invalid when TPU5 is in phase counting mode.

Table 24.10 Bits CKEG[1:0]

Bits CKEG[1:0]		Input Clock	
b4	b3	Internal Clock	External clock
0	0	Counted at falling edge	Counted at rising edge
0	1	Counted at rising edge	Counted at falling edge
1	0	Counted at both edges	Counted at both edges
1	1	Counted at both edges	Counted at both edges

Table 24.11 Bits CCLR[2:0] (TPU0, TPU3)

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
TPU0, TPU3	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 for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*1
	1	1	0	TCNT cleared by TGRD compare match/input capture*1
	1	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2

Note 1. 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.
Note 2. Synchronous operation is selected by setting the TPU.TSYR.SYNCj bit (j = 0, 3) to 1.

Table 24.12 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5)

Channel	Bits CCLR[2:0]*1			Description
	b7	b6	b5	
TPU1, TPU2, TPU4, TPU5	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 for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	Setting prohibited
	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.
Note 2. Synchronous operation is selected by setting the TPU.TSYR.SYNCj bit (j = 1, 2, 4, 5) to 1.

24.2.2 Timer Mode Register (TMDR)

Address(es): TPU0.TMDR 0008 8111h, TPU1.TMDR 0008 8121h, TPU2.TMDR 0008 8131h, TPU3.TMDR 0008 8141h, TPU4.TMDR 0008 8151h, TPU5.TMDR 0008 8161h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICSELD	ICSELB	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	b3 b0 0 0 0 0: Normal operation 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*1 0 1 0 1: Phase counting mode 2*1 0 1 1 0: Phase counting mode 3*1 0 1 1 1: Phase counting mode 4*1 Settings other than above are prohibited.	R/W
b4	BFA*2	Buffer Operation A	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3)	R/W
b5	BFB*3	Buffer Operation B	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3)	R/W
b6	ICSELB	TGRB Input Capture Input Select	0: Input capture input source is TIOCBn pin 1: Input capture input source is TIOCAn pin (n = 0 to 5)	R/W
b7	ICSELD*3	TGRD Input Capture Input Select	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCCn pin (n = 0, 3)	R/W

Note 1. Phase counting mode cannot be set for TPU0 and TPU3. A 0 should be written to bit 2 for them.
Note 2. Bit 4 of TPU1, TPU2, TPU4, and TPU5 that do not have TGRC is reserved. This bit is read as 0. The write value should be 0.
Note 3. Bits 5 and 7 of TPU1, TPU2, TPU4, and TPU5 that do not have TGRD are reserved. These bits are read as 0. The write value should be 0.

TPUm.TMDR settings should be made while TPUm.TCNT operation is stopped.

BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3) are to be used together for buffer operation.

When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

BFB Bit (Buffer Operation B)

Specifies whether TPUm.TGRB (m = 0, 3) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3) are to be used together for buffer operation.

When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

ICSELB Bit (TGRB Input Capture Input Select)

Selects the input capture input for TPUm.TGRB (m = 0 to 5).

This function allows measurement of high-level width and period of the input pulse on a TIOCAn input pin.

ICSELD Bit (TGRD Input Capture Input Select)

Selects the input capture input for TPUm.TGRD (m = 0, 3).

This function allows measurement of high-level width and period of the input pulse on a TIOCCn input pin.

24.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)

- TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR

Address(es): TPU0.TIORH 0008 8112h, TPU1.TIOR 0008 8122h, TPU2.TIOR 0008 8132h, TPU3.TIORH 0008 8142h, TPU4.TIOR 0008 8152h, TPU5.TIOR 0008 8162h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control	See Table 24.13 to Table 24.18.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 24.13 to Table 24.18.*1	R/W

Note 1. If the IOn[3:0] bit (n = A, B) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCAn/TIOCBn pin (n = 0 to 5) is placed in high impedance state.

- TPU0.TIORL, TPU3.TIORL

Address(es): TPU0.TIORL 0008 8113h, TPU3.TIORL 0008 8143h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 24.19 and Table 24.20.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 24.19 and Table 24.20.*1	R/W

Note 1. If the IOn[3:0] bit (n = C, D) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCCn/TIOCDn pin (n = 0, 3) is placed in high impedance state.

TPU has two TIORH registers, one for TPU0 and TPU3, and two TIORL registers, one for TPU0 and TPU3, and also has four TIOR registers, one for TPU1, TPU2, TPU4, and TPU5. Thus the TPU has eight timer I/O control registers in total.

TIORH, TIORL, and TIOR control registers TGRA, TGRB, TGRC, and TGRD.

Note that TIORH, TIORL, and TIOR are affected by the TMDR setting. For details, see Table 24.13 to Table 24.20.

The initial output specified by TIORH, TIORL, and TIOR is valid when the counter is stopped (the TPU.TSTR.CSTj bit (j = 0 to 5) is cleared to 0). In PWM mode 2, the output at the time when the TCNT is cleared to 0 is specified as the initial output.

When buffer operation has been selected for register TGRC or TGRD, the settings of the IOC[3:0] or IOD[3:0] bits become ineffective, and the TGRC or TGRD register simply operates as a buffer.

To specify the input capture pin in TIORH, TIORL, or TIOR, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 20, I/O Ports.

IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 0 to 5).

IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 5).

IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 0, 3).

IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 0, 3).

Table 24.13 TPU0.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPU0.TGRA Function	TIOCA0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA0 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRB Function	TIOCB0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at both edges
1	1	x	x		• Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1

x: Don't care

Note 1. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU1.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU0.TMDR.

Table 24.14 TPU1.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU1.TGRA Function**	**TIOCA1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA1 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA1 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU0.TGRA compare match/input capture Input capture at generation of TPU0.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU1.TGRB Function**	**TIOCB1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at both edges
1	1	x	x		• Capture input source is TPU0.TGRC compare match/input capture Input capture at generation of TPU0.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU1.TMDR.

Table 24.15 TPU2.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU2.TGRA Function**	**TIOCA2 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA2 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA2 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA2 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU2.TGRB Function**	**TIOCB2 Pin (Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at both edges

x: Don't care

Note 1. Selected by the ICSELB bit in TPU2.TMDR.

Table 24.16 TPU3.TIORH

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU3.TGRA Function**	**TIOCA3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA3 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU3.TGRB Function**	**TIOCB3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at both edges
1	1	x	x		• Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1

x: Don’t care

Note 1. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU4.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU3.TMDR.

Table 24.17 TPU4.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU4.TGRA Function**	**TIOCA4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA4 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU3.TGRA compare match/input capture Input capture at generation of TPU3.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU4.TGRB Function**	**TIOCB4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at both edges
1	1	x	x		• Capture input source is TPU3.TGRC compare match/input capture Input capture at generation of TPU3.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU4.TMDR.

Table 24.18 TPU5.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU5.TGRA Function**	**TIOCA5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA5 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA5 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA5 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU5.TGRB Function**	**TIOCB5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at both edges

x: Don't care

Note 1. Selected by the ICSELB bit in TPU5.TMDR.

Table 24.19 TPU0.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU6.TGRC Function	TIOCC6 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC0 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU6.TGRD Function	TIOCD6 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at both edges
1	1	x	x		• Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3

x: Don't care

Note 1. When the BFA bit in TPU0.TMDR is set to 1 (TPU0.TGRA and TPU0.TGRC are used for buffer operation) and TPU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU0.TMDR is set to 1 (TPU0.TGRB and TPU0.TGRD are used for buffer operation) and TPU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU0.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU0.TMDR.

Table 24.20 TPU3.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU3.TGRC Function	TIOCC3 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC3 pin; input capture at both edges
1	1	x	x		• Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU3.TGRD Function	TIOCD3 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at both edges
1	1	X	x		• Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3

x: Don’t care

Note 1. When the BFA bit in TPU3.TMDR is set to 1 (TPU3.TGRA and TPU3.TGRC are used for buffer operation) and TPU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU3.TMDR is set to 1 (TPU3.TGRB and TPU3.TGRD are used for buffer operation) and TPU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU3.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU3.TMDR.

24.2.4 Timer Interrupt Enable Register (TIER)

Address(es): TPU0.TIER 0008 8114h, TPU1.TIER 0008 8124h, TPU2.TIER 0008 8134h, TPU3.TIER 0008 8144h, TPU4.TIER 0008 8154h, TPU5.TIER 0008 8164h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGRA Interrupt Enable	0: Interrupt requests (TGImA) disabled 1: Interrupt requests (TGImA) enabled (m = 0 to 5)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 5)	R/W
b2	TGIEC*1	TGRC Interrupt Enable	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3)	R/W
b3	TGIED*1	TGRD Interrupt Enable	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 5)	R/W
b5	TCIEU*2	Underflow Interrupt Enable	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE*3	A/D Conversion Start Request Enable	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

Note 1. Bits 3 and 2 in TIER of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.
Note 2. Bit 5 in TIER of TPU0 and TPU3 is reserved. This bit is read as 0. The write value should be 0.
Note 3. Bit 7 in TIER of TPU5 is reserved. This bit is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

Enables/disables generation of A/D conversion start requests by TPUm.TGRA (m = 0 to 4) input capture/compare match.

24.2.5 Timer Status Register (TSR)

Address(es): TPU0.TSR 0008 8115h, TPU1.TSR 0008 8125h, TPU2.TSR 0008 8135h, TPU3.TSR 0008 8145h, TPU4.TSR 0008 8155h, TPU5.TSR 0008 8165h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGFA	Input Capture/Output Compare Flag A	0: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has not occurred. 1: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has occurred. (m = 0 to 5)	R/W*2
b1	TGFB	Input Capture/Output Compare Flag B	0: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has not occurred. 1: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has occurred. (m = 0 to 5)	R/W*2
b2	TGFC*4	Input Capture/Output Compare Flag C	0: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has not occurred. 1: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has occurred. (m = 0, 3)	R/W*2
b3	TGFD*4	Input Capture/Output Compare Flag D	0: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has not occurred. 1: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has occurred. (m = 0, 3)	R/W*2
b4	TCFV	Overflow Flag	0: TPUm.TCNT has not overflowed. 1: TPUm.TCNT has overflowed. (m = 0 to 5)	R/W*2
b5	TCFU*3	Underflow Flag	0: TPUm.TCNT has not underflowed. 1: TPUm.TCNT has underflowed. (m = 1, 2, 4, 5)	R/W*2
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R
b7	TCFD*1	Counting Direction Flag	0: TPUm.TCNT counts down. 1: TPUm.TCNT counts up. (m = 1, 2, 4, 5)	R

Note 1. Bit 7 of registers TPU0.TSR and TPU3.TSR is reserved. The bit is read as 1. The write value should be 1.
Note 2. Only writing 0 to this bit is possible; this clears the flag.
Note 3. Bit 5 of registers TPU0.TSR and TPU3.TSR is reserved. The bit is read as 0. The write value should be 0.
Note 4. Bits 2 and 3 of registers TPU1.TSR, TPU2.TSR, TPU4.TSR and TPU5.TSR are reserved. The bits are read as 0. The write value should be 0.

TGFA Flag (Input Capture/Output Compare Flag A)

This status flag indicates that input capture to TPUm.TGRA or compare match with TPUm.TGRA (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRA holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRA.
- When TPUm.TGRA is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRA.

[Clearing conditions]

- Activation of the DTC by the TGImA interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFA after reading its value as 1.

TGFB Flag (Input Capture/Output Compare Flag B)

This status flag indicates that input capture to TPUm.TGRB or compare match with TPUm.TGRB (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRB holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRB.
- When TPUm.TGRB is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRB.

[Clearing conditions]

- Activation of the DTC by the TGImB interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFB after reading its value as 1.

TGFC Flag (Input Capture/Output Compare Flag C)

This status flag indicates that input capture to TPUm.TGRC or compare match with TPUm.TGRC (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRC holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRC.
- When TPUm.TGRC is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRC.

[Clearing conditions]

- Activation of the DTC by the TGImC interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFC after reading its value as 1.

TGFD Flag (Input Capture/Output Compare Flag D)

This status flag indicates that input capture to TPUm.TGRD or compare match with TPUm.TGRD (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRD holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRD.
- When TPUm.TGRD is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRD.

[Clearing conditions]

- Activation of the DTC by the TGImD interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFD after reading its value as 1.

TCFV Flag (Overflow Flag)

This status flag indicates an overflow of TPUm.TCNT (m = 0 to 5).

[Setting condition]

- Overflow of the value in TPUm.TCNT (TCNT counted from FFFFh to 0000h).

[Clearing condition]

- Writing 0 to TCFV after reading its value as 1.

TCFU Flag (Underflow Flag)

This status flag indicates an underflow of TPUm.TCNT (m = 1, 2, 4, 5).

[Setting condition]

- Underflow of the value in TPUm.TCNT (TCNT counted from 0000h to FFFFh).

[Clearing condition]

- Writing 0 to TCFU after reading its value as 1.

24.2.6 Timer Counter (TCNT)

Address(es): TPU0.TCNT 0008 8116h, TPU1.TCNT 0008 8126h, TPU2.TCNT 0008 8136h, TPU3.TCNT 0008 8146h, TPU4.TCNT 0008 8156h, TPU5.TCNT 0008 8166h

	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

TPUm.TCNT is a readable/writable counter that counts the internal clock or external events.

24.2.7 Timer General Register A (TGRA)
Timer General Register B (TGRB)
Timer General Register C (TGRC)
Timer General Register D (TGRD)

Address(es): TPU0.TGRA 0008 8118h, TPU0.TGRB 0008 811Ah, TPU0.TGRC 0008 811Ch, TPU0.TGRD 0008 811Eh, TPU1.TGRA 0008 8128h, TPU1.TGRB 0008 812Ah, TPU2.TGRA 0008 8138h, TPU2.TGRB 0008 813Ah, TPU3.TGRA 0008 8148h, TPU3.TGRB 0008 814Ah, TPU3.TGRC 0008 814Ch, TPU3.TGRD 0008 814Eh, TPU4.TGRA 0008 8158h, TPU4.TGRB 0008 815Ah, TPU5.TGRA 0008 8168h, TPU5.TGRB 0008 816Ah

	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

TPU has 16 TGR registers in total, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5. TPUm.TGRA (m = 0 to 5), TPUm.TGRB (m = 0 to 5), TPUm.TGRC (m = 0, 3), and TPUm.TGRD (m = 0, 3) are readable/writable registers with a dual function as output compare and input capture registers.
TPUm.TGRC and TPUm.TGRD can also be specified for operation as buffer registers. Register combinations during buffer operations are TPUm.TGRA—TPUm.TGRC and TPUm.TGRB—TPUm.TGRD.

24.2.8 Timer Start Register (TSTR)

Address(es): TPU.TSTR 0008 8100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST5	CST4	CST3	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: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST1	Counter Start 1		R/W
b2	CST2	Counter Start 2		R/W
b3	CST3	Counter Start 3		R/W
b4	CST4	Counter Start 4		R/W
b5	CST5	Counter Start 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSTR starts or stops TCNT operation for TPU0 to TPU5.
Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT operation.

CSTn Bit (Counter Start n) (n = 0 to 5)

This bit starts or stop the TCNT.
When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D; n = 0 to 5) specified for output, count operation stops but the output compare output level of the corresponding TIOCyn pin is retained.
If TIORH, TIORL, or TIOR is written to when the CSTn bit is 0, the pin output level will be changed to the set initial output value.

24.2.9 Timer Synchronous Register (TSYR)

Address(es): TPU.TSYR 0008 8101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronization 0	0: TCNT operates independently (TCNT setting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous setting/synchronous clearing is possible)	R/W
b1	SYNC1	Timer Synchronization 1		R/W
b2	SYNC2	Timer Synchronization 2		R/W
b3	SYNC3	Timer Synchronization 3		R/W
b4	SYNC4	Timer Synchronization 4		R/W
b5	SYNC5	Timer Synchronization 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 0 to 5) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the TCR.CCLR[2:0] bits in addition to the SYNCn bit.

TPU.TSYR selects independent operation or synchronous operation for TCNT of TPU0 to TPU5.

SYNCn Bit (Timer Synchronization n) (n = 0 to 5)

This bit selects whether the TCNT operation is independent of or synchronized with TCNT of other channels. When synchronous operation is selected, synchronous setting of multiple TCNT and synchronous clearing through counter clearing on another channel are possible.

24.2.10 Noise Filter Control Register (NFCR)

Address(es): TPU0.NFCR 0008 8108h, TPU1.NFCR 0008 8109h, TPU2.NFCR 0008 810Ah, TPU3.NFCR 0008 810Bh, TPU4.NFCR 0008 810Ch, TPU5.NFCR 0008 810Dh

	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 Enable A	0: The noise filter for TIOCAm is disabled. 1: The noise filter for TIOCAm is enabled. (m = 0 to 5)	R/W
b1	NFBEN	Noise Filter Enable B	0: The noise filter for TIOCBm is disabled. 1: The noise filter for TIOCBm is enabled. (m = 0 to 5)	R/W
b2	NFCEN*1	Noise Filter Enable C	0: The noise filter for TIOCCm is disabled. 1: The noise filter for TIOCCm is enabled. (m = 0, 3)	R/W
b3	NFDEN*1	Noise Filter Enable D	0: The noise filter for TIOCDm is disabled. 1: The noise filter for TIOCDm is enabled. (m = 0, 3)	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: Clock source that drives counting	R/W
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits is not possible.	R

Note 1. Bits 2 and 3 of TPU1.NFCR, TPU2.NFCR, TPU4.NFCR, and TPU5.NFCR are reserved. The bits are read as 0. Writing to these bits is not possible.

Only set the TPUm.NFCR registers while the TPUm.TCNT is stopped.

NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register before changing the NFAEN value.

NFBEN Bit (Noise Filter Enable B)

This bit disables or enables the noise filter for the TIOCBm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register before changing the NFBEN value.

NFCEN Bit (Noise Filter Enable C)

This bit disables or enables the noise filter for the TIOCCm pin (m = 0, 3).

Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register before changing the NFCEN value.

NFDEN Bit (Noise Filter Enable D)

This bit disables or enables the noise filter for the TIOCDm pin (m = 0, 3).
Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register before changing the NFDEN value.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the noise filter.
When the count source is selected with NFCS[1:0] bits set to 11b, the clock that can be used as sampling clock are the internal clocks other than PCLK/1 specified with the TPSC[2:0] bits and the external clock. To select the PCLK/1 as both the count clock and the sampling clock, set the NFCS[1:0] bits to 00b.
The input-capture signal is sampled on rising edges of the selected clock signal. If the sampled levels match three times in a row, the given level is passed through as the input-capture signal. If the levels do not match, the existing value is retained.
After setting the NFCS[1:0] bits, wait for two selected sampling periods before setting the input capture function.

24.3 Operation

24.3.1 Basic Functions

Each channel has a TPUm.TCNT and a TPUm.TGRy register (y = A to D).
TCNT is a 16-bit up-counter, which can function as a free-running counter, periodic counter, or event counter.
TGRy can be used as an input capture register or output compare register.

(1) Counter Operation

When the CSTj bit (j = 0 to 5) in TPU.TSTR is set to 1, the TCNT for the corresponding channel starts counting.

(a) Example of count operation setting procedure

Figure 24.2 shows an example of the count operation setting procedure.

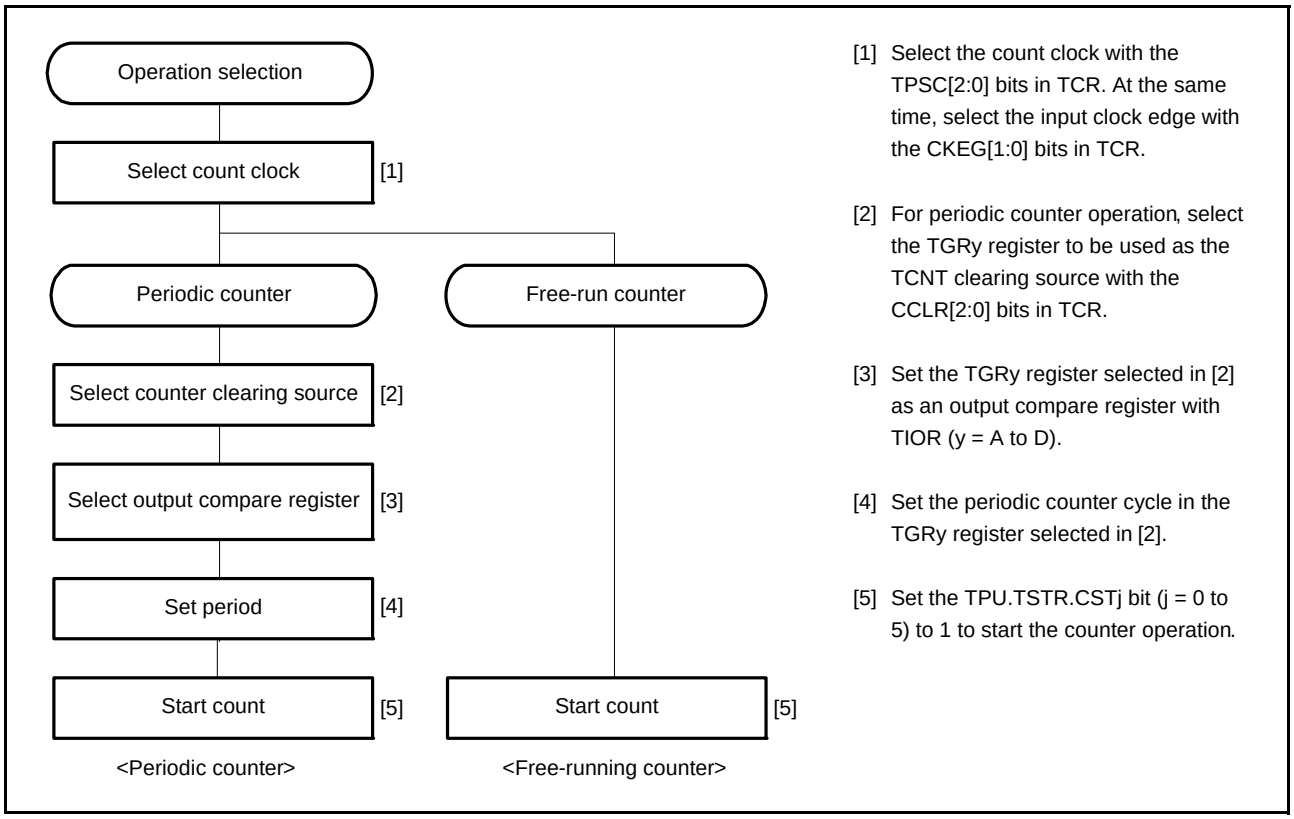


Figure 24.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, TPUm.TCNT are all set as free-running counters. When the relevant bit in TPU.TSTR is set to 1, the corresponding TCNT starts up-count operation as a free-running counter. When TCNT overflows (changes from FFFFh to 0000h), the TPU requests an interrupt. After an overflow, TCNT restarts counting up from 0000h.
Figure 24.3 shows free-running counter operation.

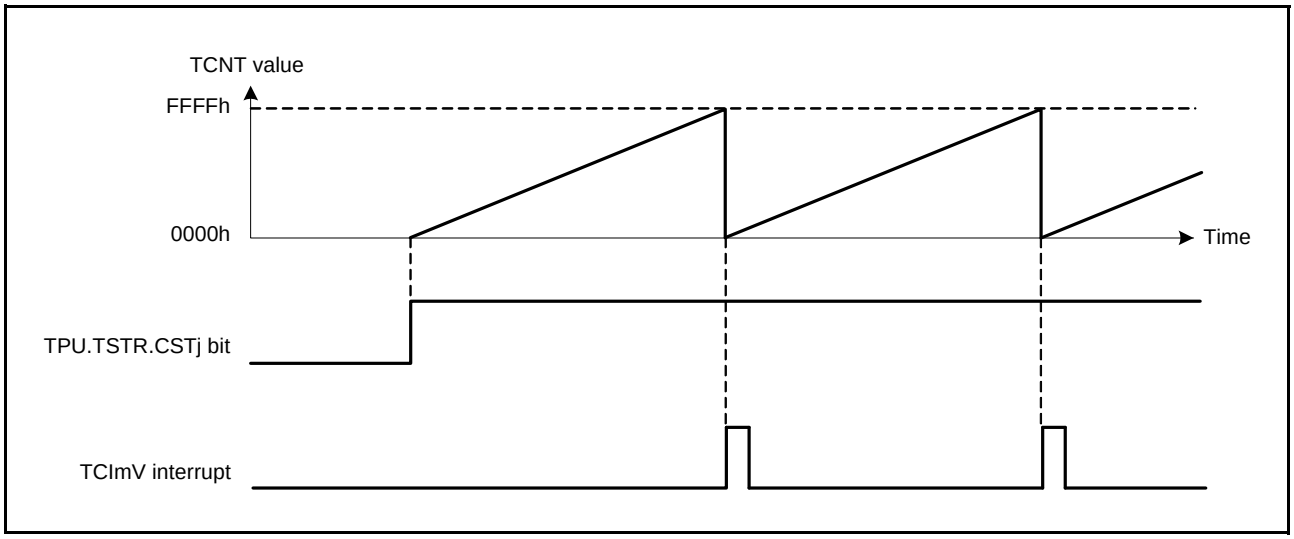


Figure 24.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT for the relevant channel performs periodic count operation. The TPUm.TGRy for setting the period is set as an output compare register, and counter clearing by compare match is selected by the TPUm.TCR.CCLR[2:0] bits. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPU.TSTR is set to 1. When the count value matches the TGRy value, TCNT is cleared to 0000h.
At this time, the TPU requests an interrupt. After a compare match, TCNT restarts counting up from 0000h.
Figure 24.4 shows periodic counter operation.

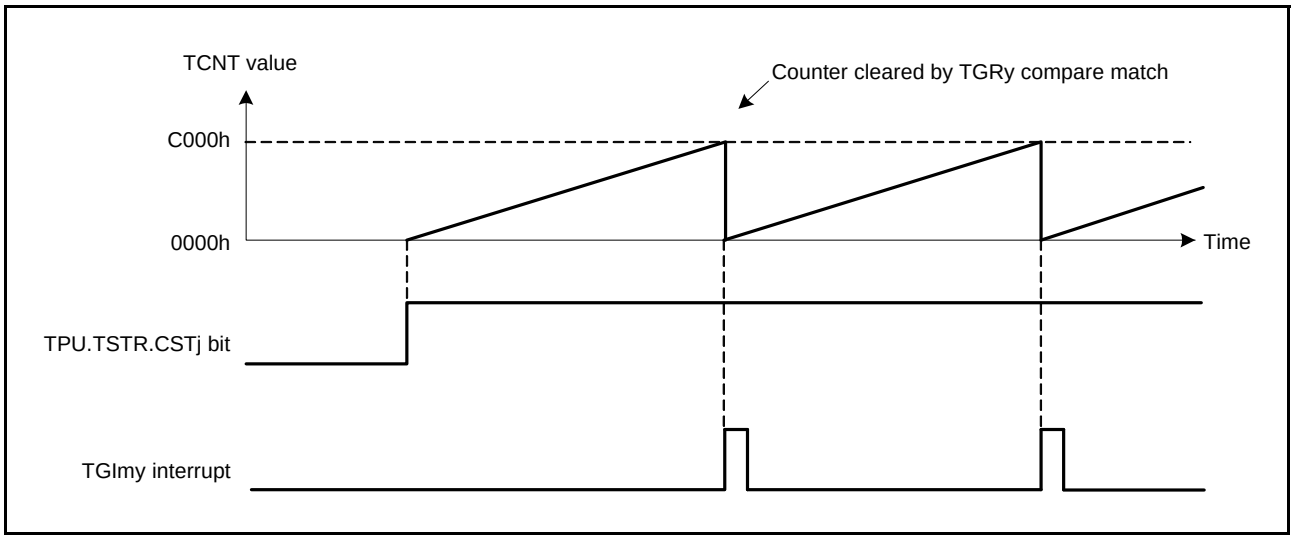


Figure 24.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform low, high, or toggle output from the corresponding output pin using a compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 24.5 shows an example of the setting procedure for waveform output by a compare match.

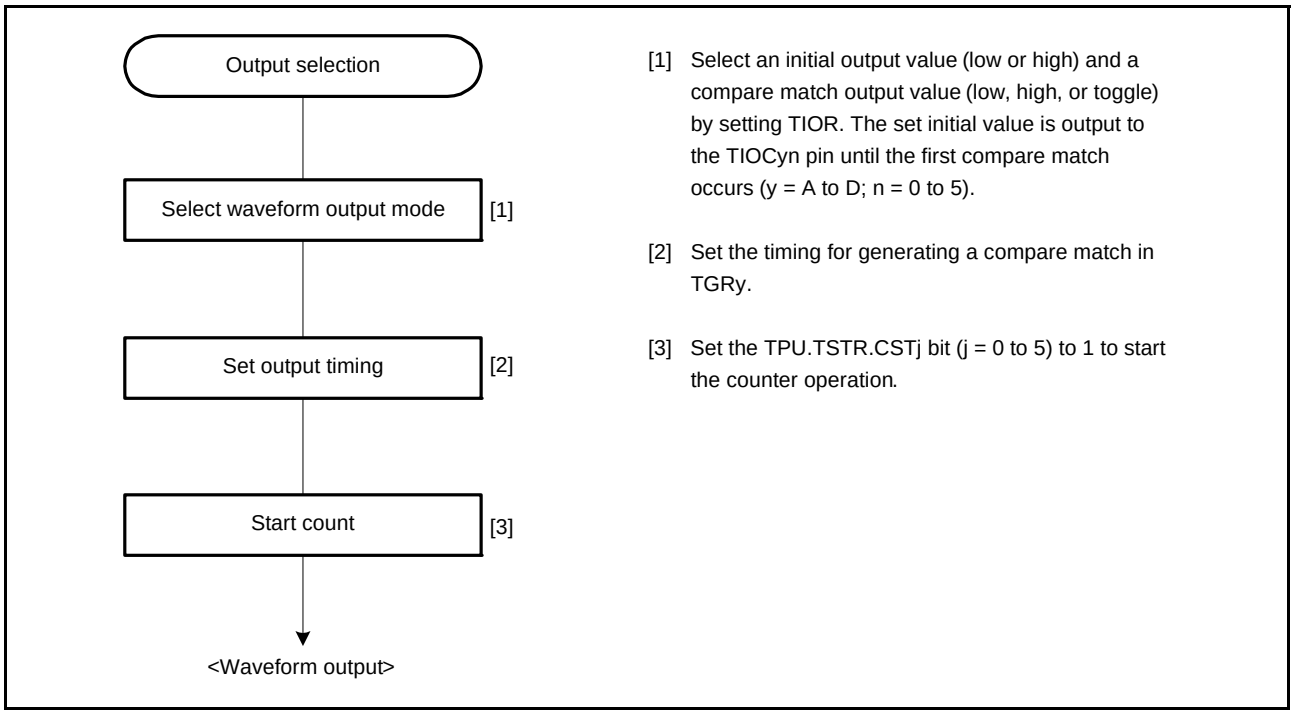


Figure 24.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 24.6 shows an example of low output/high output.
In this example, TPUm.TCNT has been set 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 set level and the pin level match, the pin level does not change.

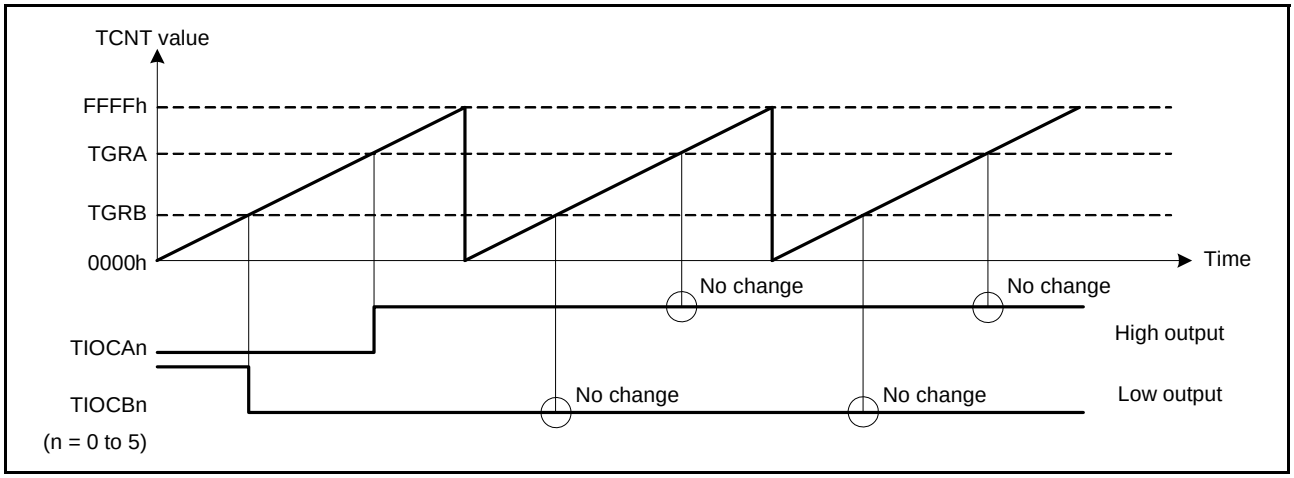


Figure 24.6 Example of Low-Output/High-Output Operation

Figure 24.7 shows an example of toggle output.
In this example, TPUm.TCNT has been set as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

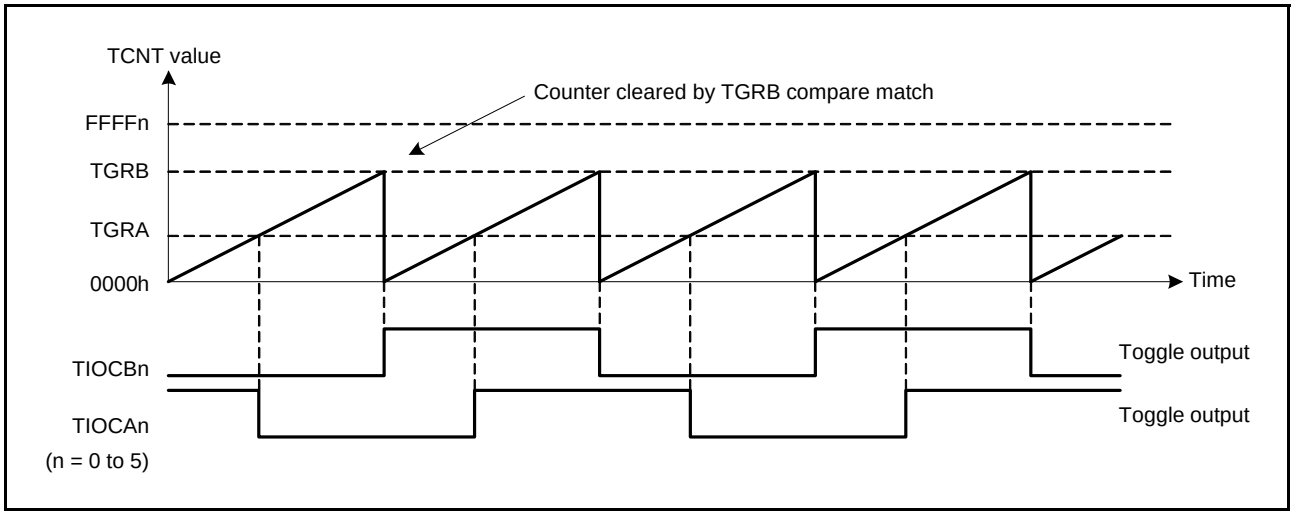


Figure 24.7 Example of Toggle Output Operation

(3) Input Capture Function

The TPUm.TCNT value can be transferred to TPUm.TGRy on detection of the TIOCyn pin (y = A to D; n = 0 to 5) input edge.
The rising edge, the falling edge, or both edges can be selected as the detection edge. It is also possible to specify the count clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 as the input capture source. Noise filtering can be applied to the input capture input.

Note: Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
Note: When another channel's count clock is used as the input capture input for TPU0 and TPU3, PCLK/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of setting procedure for input capture operation

Figure 24.8 shows an example of the setting procedure for input capture operation.

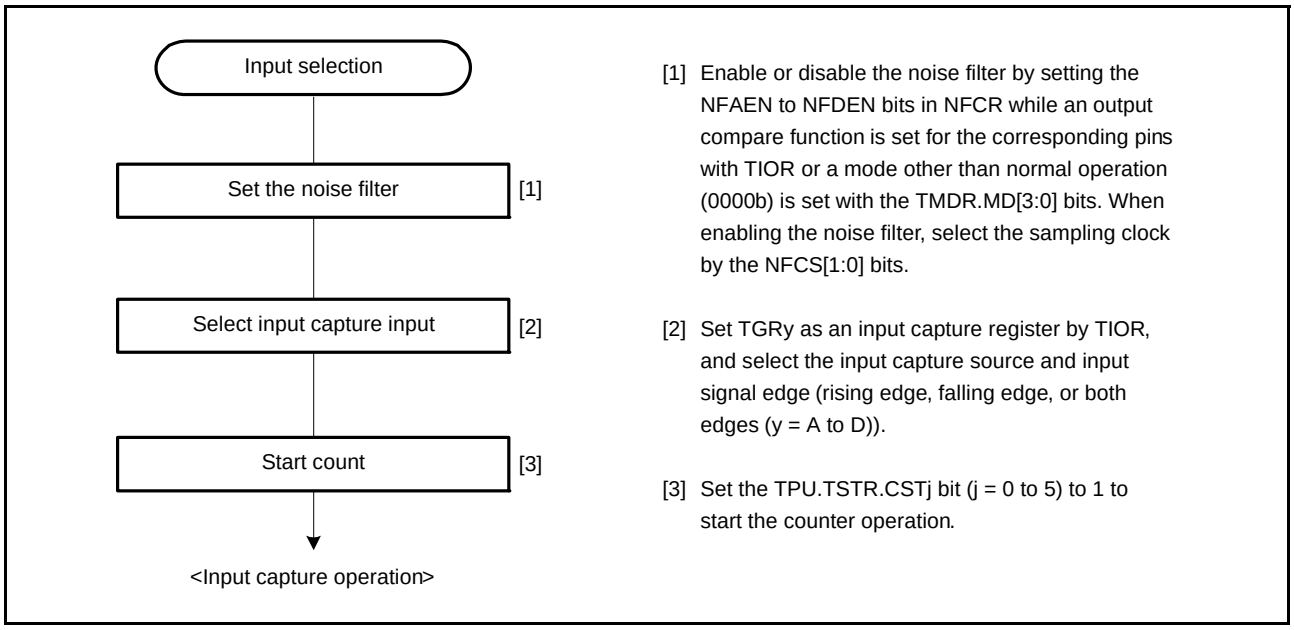


Figure 24.8 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 24.9 shows an example of input capture operation when the noise filter is stopped.
In this example, both rising and falling edges have been selected as the TIOCAn pin input capture input edge, the falling edge has been selected as the TIOCBn pin input capture input edge, and counter clearing by TPUm.TGRB input capture has been set for TPUm.TCNT.

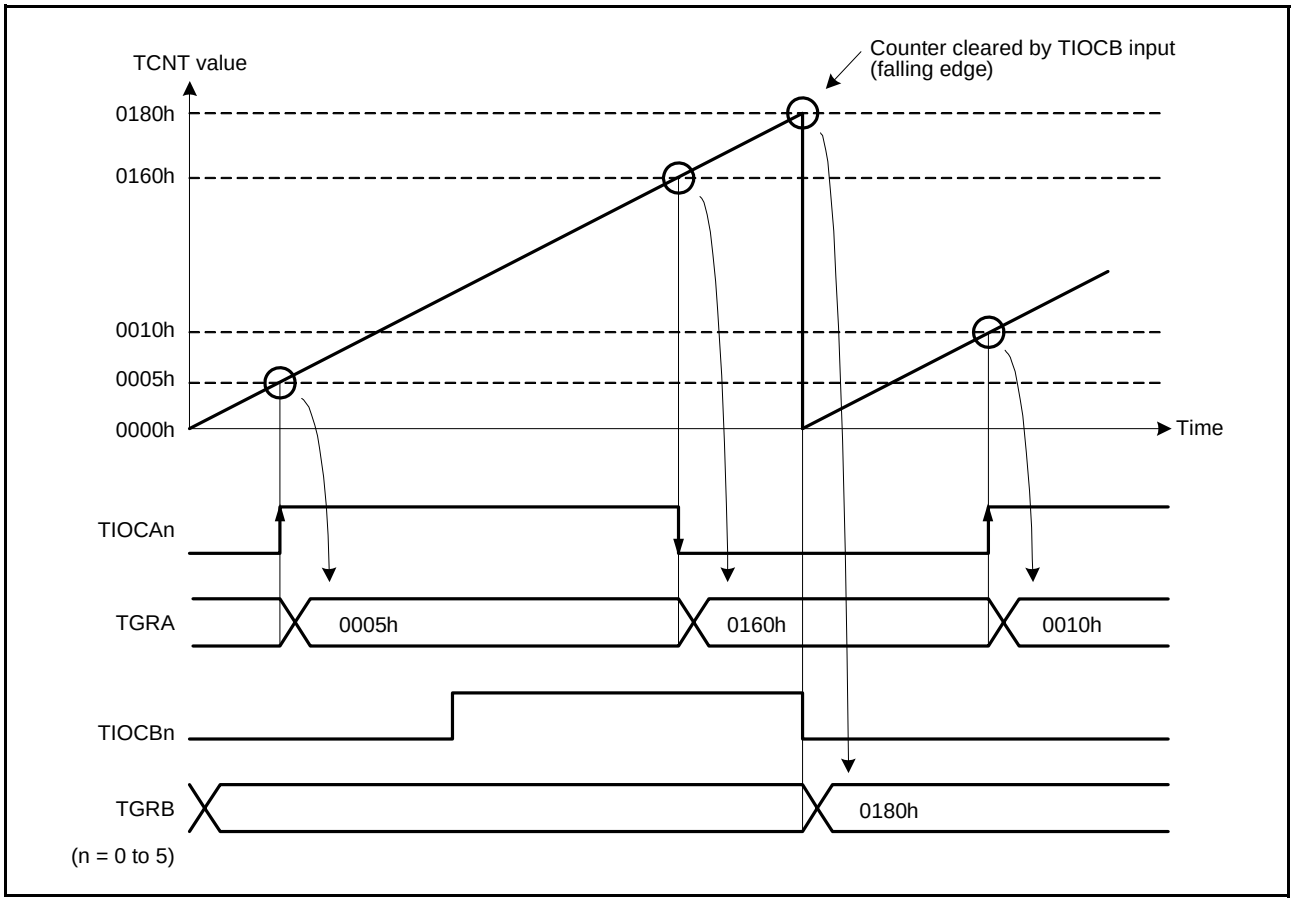


Figure 24.9 Example of Input Capture Operation (with Noise Filter Stopped)

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 + PCLK) due to noise filtering for the input capture input.

24.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT can be rewritten simultaneously (synchronous setting). Also, multiple TCNT can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TPUm.TCR.
Synchronous operation enables TPUm.TGRy to be incremented with respect to a single time base.
TPU0 to TPU5 can all be set for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 24.10 shows an example of the synchronous operation setting procedure.

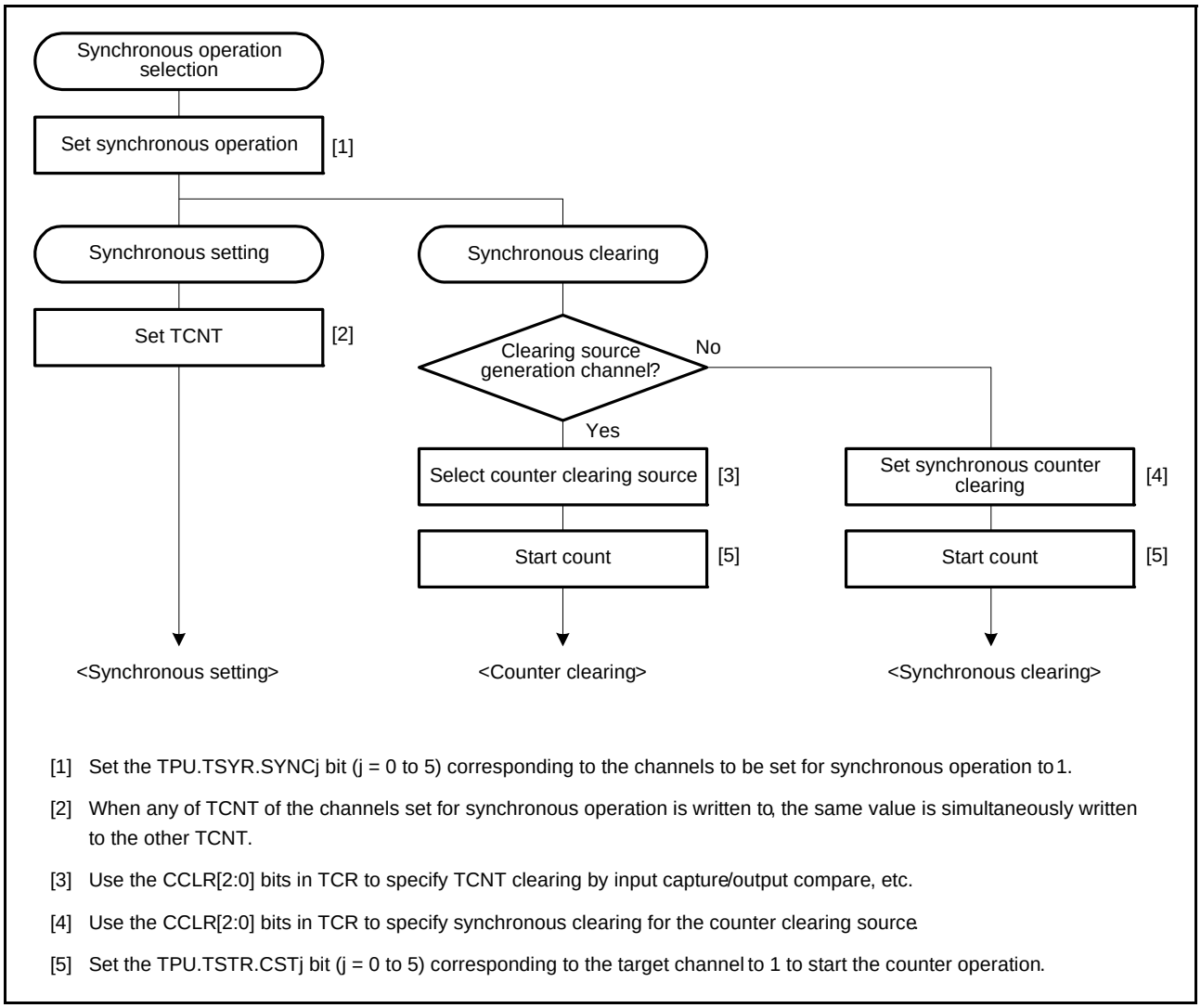


Figure 24.10 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 24.11 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been set for TPU0 to TPU2, TPU0.TGRB compare match has been set as the TPU0 counter clearing source, and synchronous clearing has been set for the TPU1 and TPU2 counter clearing source.
Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous setting and synchronous clearing by TPU0.TGRB compare match are performed for TPUm.TCNT of TPU0 to TPU2, and the data set in TPU0.TGRB is used as the PWM cycle.
For details on PWM modes, see section 24.3.5, PWM Modes.

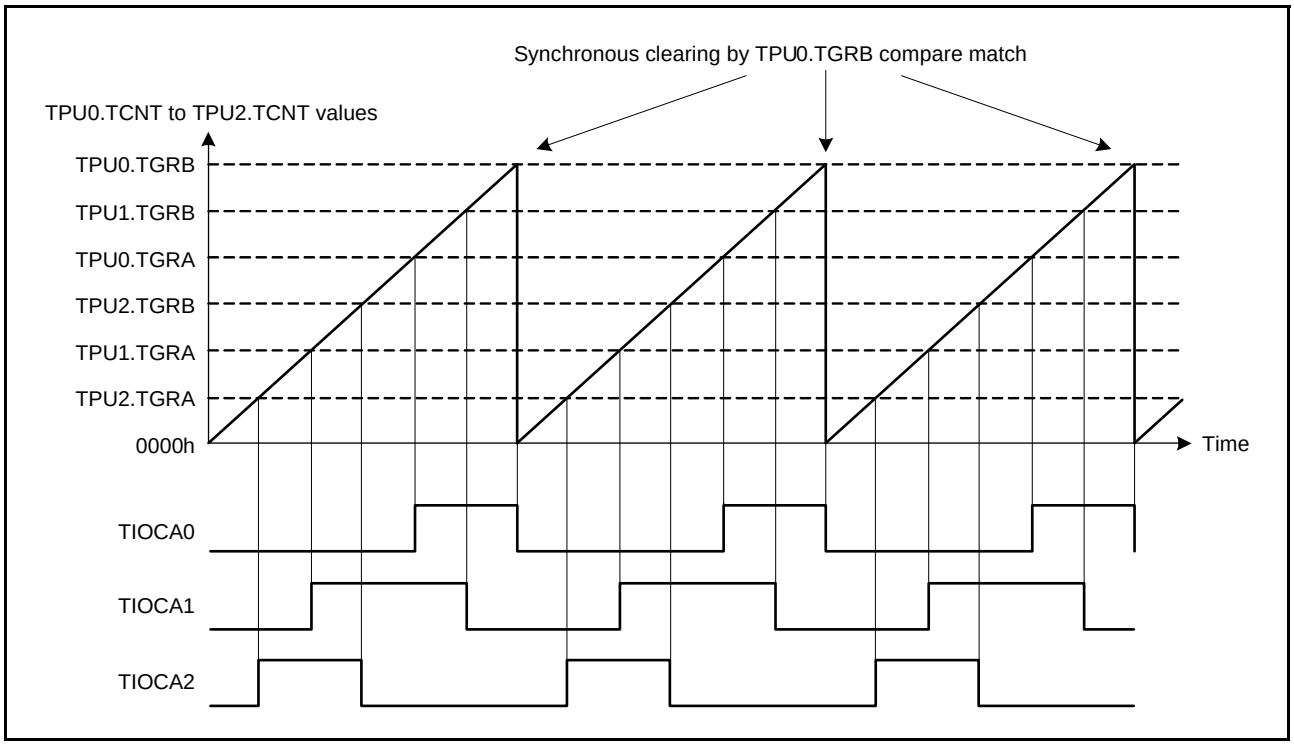


Figure 24.11 Example of Synchronous Operation

24.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3, enables TPUm.TGRC and TPUm.TGRD to be used as buffer registers. Buffer operation differs depending on whether TPUm.TGRy has been set as an input capture register or a compare match register.

Table 24.21 lists the register combinations used in buffer operation.

Table 24.21 Register Combinations

Channel	Timer General Register	Buffer Register
TPU0	TPU0.TGRA	TPU0.TGRC
	TPU0.TGRB	TPU0.TGRD
TPU3	TPU3.TGRA	TPU3.TGRC
	TPU3.TGRB	TPU3.TGRD

- When TPUm.TGRy 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 shown in Figure 24.12.

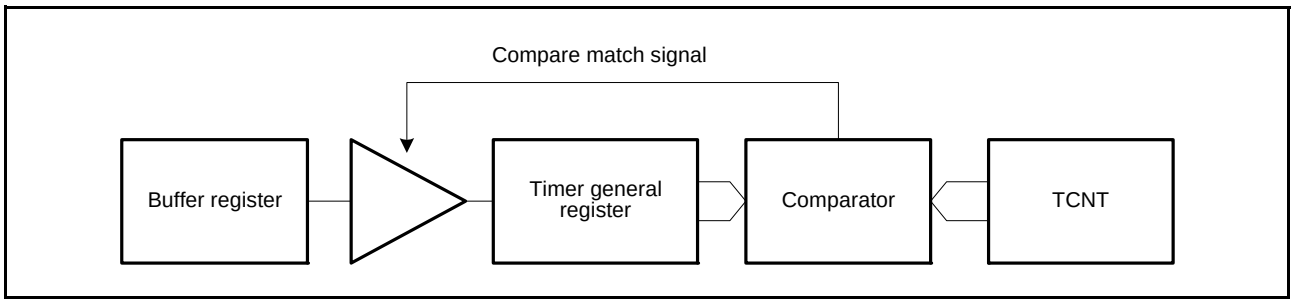


Figure 24.12 Compare Match Buffer Operation

- When TPUm.TGRy is an input capture register

When input capture occurs, the value in TPUm.TCNT is transferred to TGRy and the value previously held in TGRy is simultaneously transferred to the buffer register.

This operation is shown in Figure 24.13.

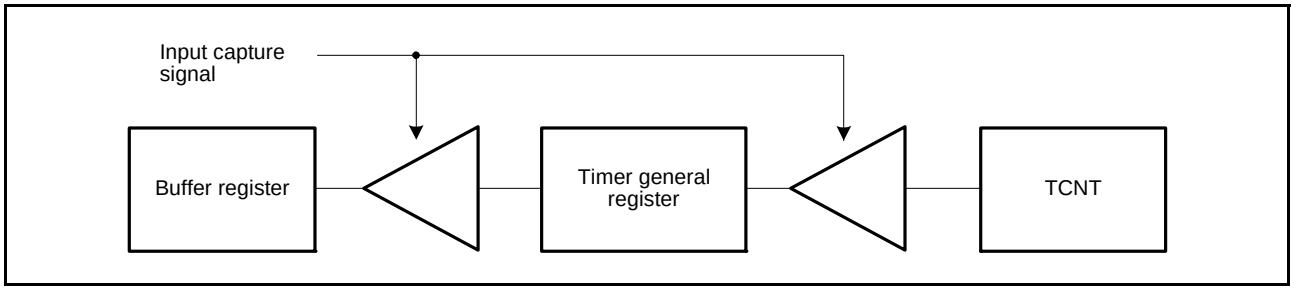


Figure 24.13 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 24.14 shows an example of the buffer operation setting procedure.

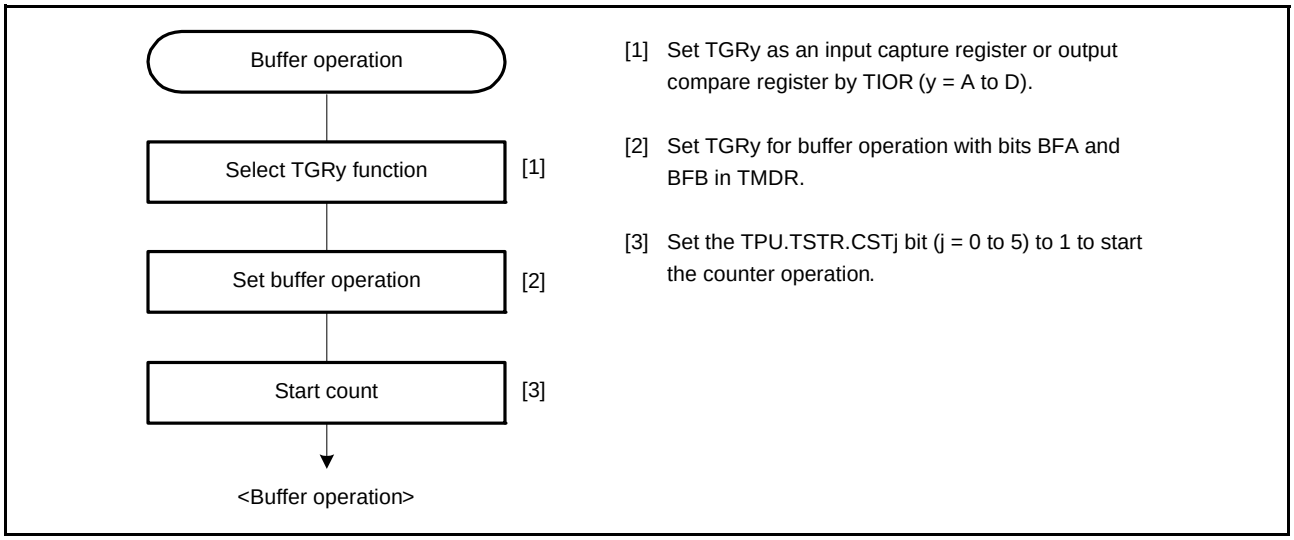


Figure 24.14 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 24.15 shows an operation example in which PWM mode 1 has been set for TPU0, and buffer operation has been set for TPU0.TGRA and TPU0.TGRC. The settings used in this example are TPU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B.
As buffer operation has been set, when compare match A occurs, the output changes and the TPU0.TGRC value is simultaneously transferred to TPU0.TGRA. This operation is repeated each time compare match A occurs.
For details on PWM modes, see section 24.3.5, PWM Modes.

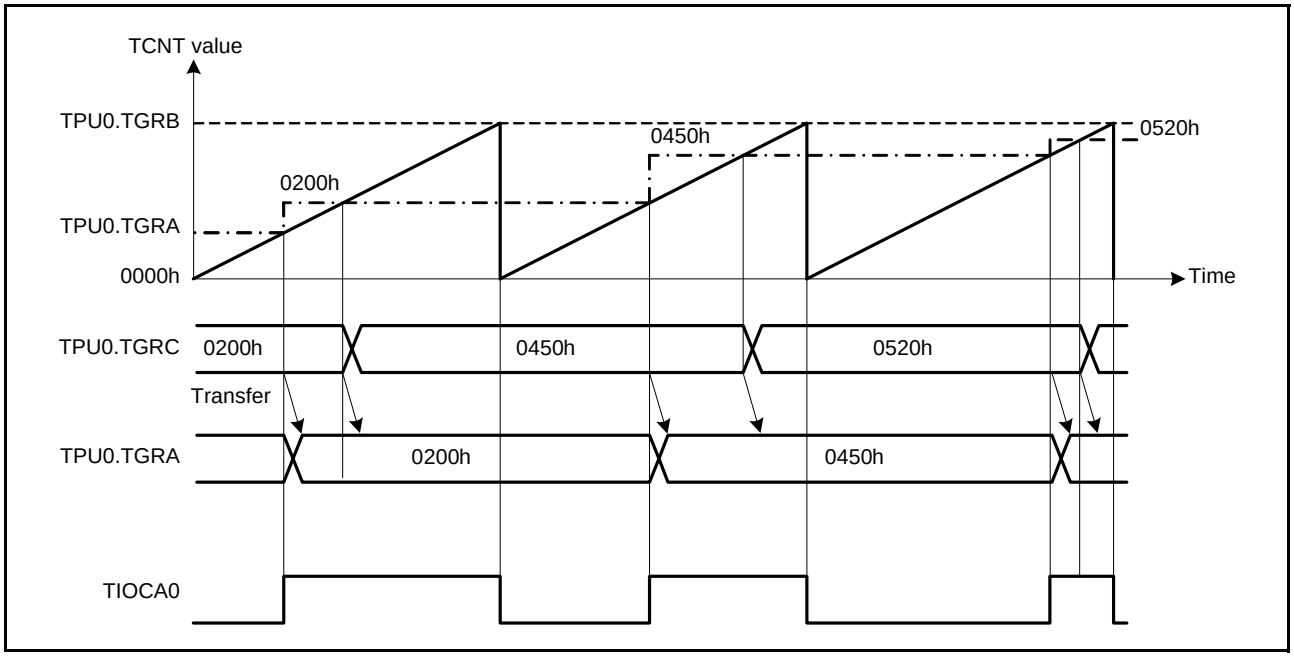


Figure 24.15 Example of Buffer Operation (1)

(b) When TPUm.TGRy is an input capture register

Figure 24.16 shows an operation example in which TPUm.TGRA has been set as an input capture register, and buffer operation has been set for the TGRA register and TPUm.TGRC.
Counter clearing by TGRA input capture has been set for TPUm.TCNT, and both rising and falling edges have been selected as the TIOCAn pin input capture input edge.
As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

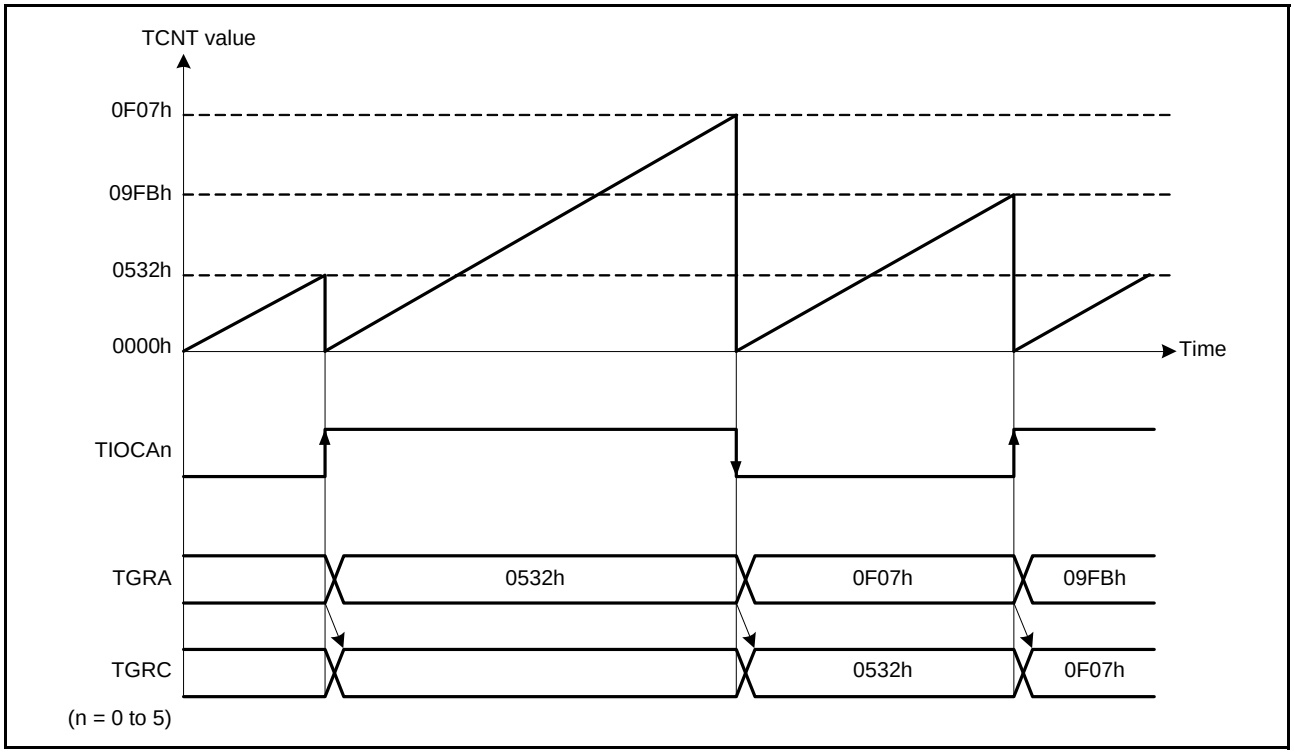


Figure 24.16 Example of Buffer Operation (2)

24.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.
This function works by counting the TPU1 (TPU4) count clock at overflow/underflow of TPU2.TCNT (TPU5.TCNT) as set by the TPSC[2:0] bits in TPU1.TCR (TPSC[2:0] bits in TPU4.TCR).
Underflow occurs only when the lower 16-bit TPUm.TCNT is in phase counting mode.
Table 24.22 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for TPU1 or TPU4, the count clock setting is invalid and the counter operates independently in phase counting mode.

Table 24.22 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT

(1) Example of Cascaded Operation Setting Procedure

Figure 24.17 shows an example of the setting procedure for cascaded operation.

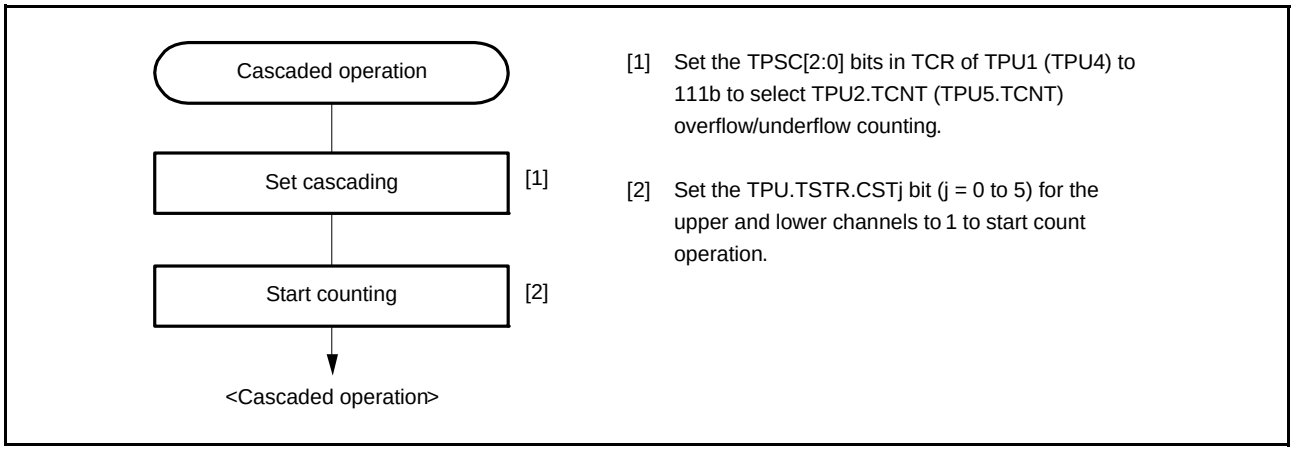


Figure 24.17 Cascaded Operation Setting Procedure

(2) Examples of Cascaded Operation

Figure 24.18 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, TPU1.TGRA and TPU2.TGRA have been set as input capture registers, and the rising edge of the TIOCA1 and TIOCA2 pins has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TPU1.TGRA, and the lower 16 bits to TPU2.TGRA.

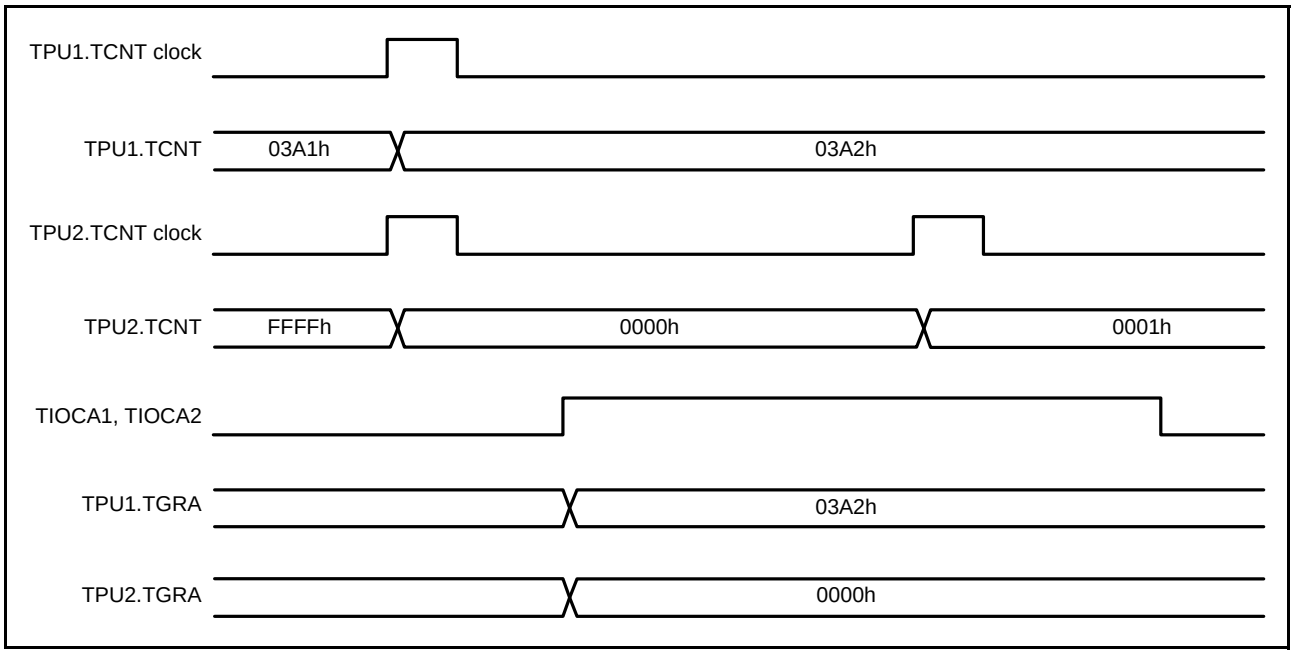


Figure 24.18 Example of Cascaded Operation (1)

Figure 24.19 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode 1 has been specified for TPU2.

TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

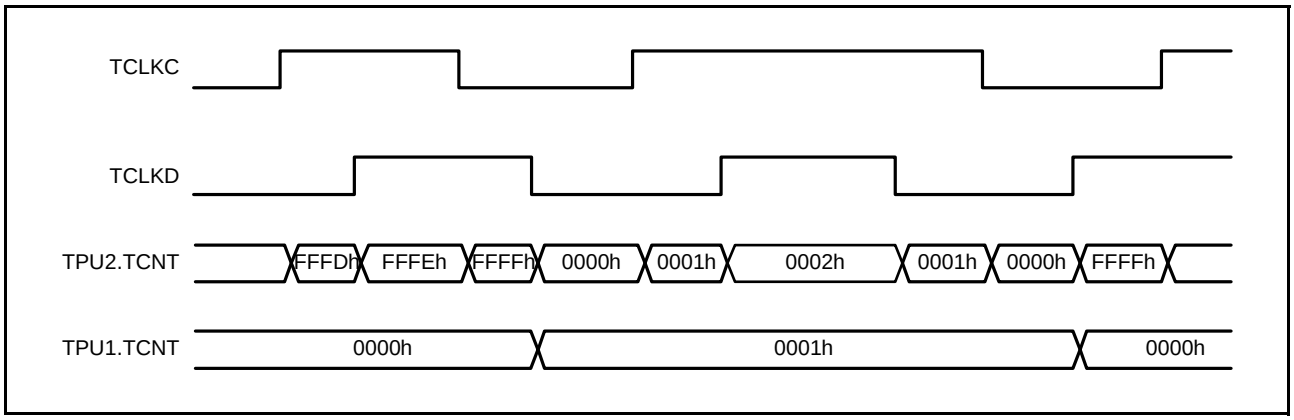


Figure 24.19 Example of Cascaded Operation (2)

24.3.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. low, high, or toggle output can be selected as the output level in response to compare match of each TPUm.TGRy.

Settings of TGRy registers can output a PWM waveform in the range of 0% to 100% duty cycle.

Specifying TGRy compare match as the counter clearing source enables the cycle to be set in that register. All channels can be set for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

1. PWM mode 1

   PWM waveform is generated from the TIOCAn and TIOCCn pins by pairing TPUm.TGRA with TPUm.TGRB and TPUm.TGRC with TPUm.TGRD. The outputs specified by the IOA[3:0] bits in TPUm.TIOR(H) and IOC[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches A and C, respectively. The outputs specified by the IOB[3:0] bits in TPUm.TIOR(H) and IOD[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRy registers are identical, the output value does not change even when a compare match occurs.

   In PWM mode 1, a maximum 8-phase PWM output is possible.

2. PWM mode 2

   PWM waveform is generated by using one TPUm.TGRy as the cycle register and the others as duty cycle registers. The output specified in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is performed by compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is the initial value set in TIORH, TIORL, or TIOR. If the set values of the cycle register and duty cycle register are identical, the output value does not change even when a compare match occurs.

   In PWM mode 2, a maximum 15-phase PWM waveform is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is listed in Table 24.23.

Table 24.23 PWM Output Registers and Output Pins

Channel	Register	Output Pin	
		PWM Mode 1	PWM Mode 2
TPU0	TPU0.TGRA	TIOCA0	TIOCA0
	TPU0.TGRB		TIOCB0
	TPU0.TGRC	TIOCC0	TIOCC0
	TPU0.TGRD		TIOCD0
TPU1	TPU1.TGRA	TIOCA1	TIOCA1
	TPU1.TGRB		TIOCB1
TPU2	TPU2.TGRA	TIOCA2	TIOCA2
	TPU2.TGRB		TIOCB2
TPU3	TPU3.TGRA	TIOCA3	TIOCA3
	TPU3.TGRB		TIOCB3
	TPU3.TGRC	TIOCC3	TIOCC3
	TPU3.TGRD		TIOCD3
TPU4	TPU4.TGRA	TIOCA4	TIOCA4
	TPU4.TGRB		TIOCB4
TPU5	TPU5.TGRA	TIOCA5	TIOCA5
	TPU5.TGRB		TIOCB5

Note: In PWM mode 2, PWM waveform output is not possible for the TPUm.TGRy register in which the cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 24.20 shows an example of the PWM mode setting procedure.

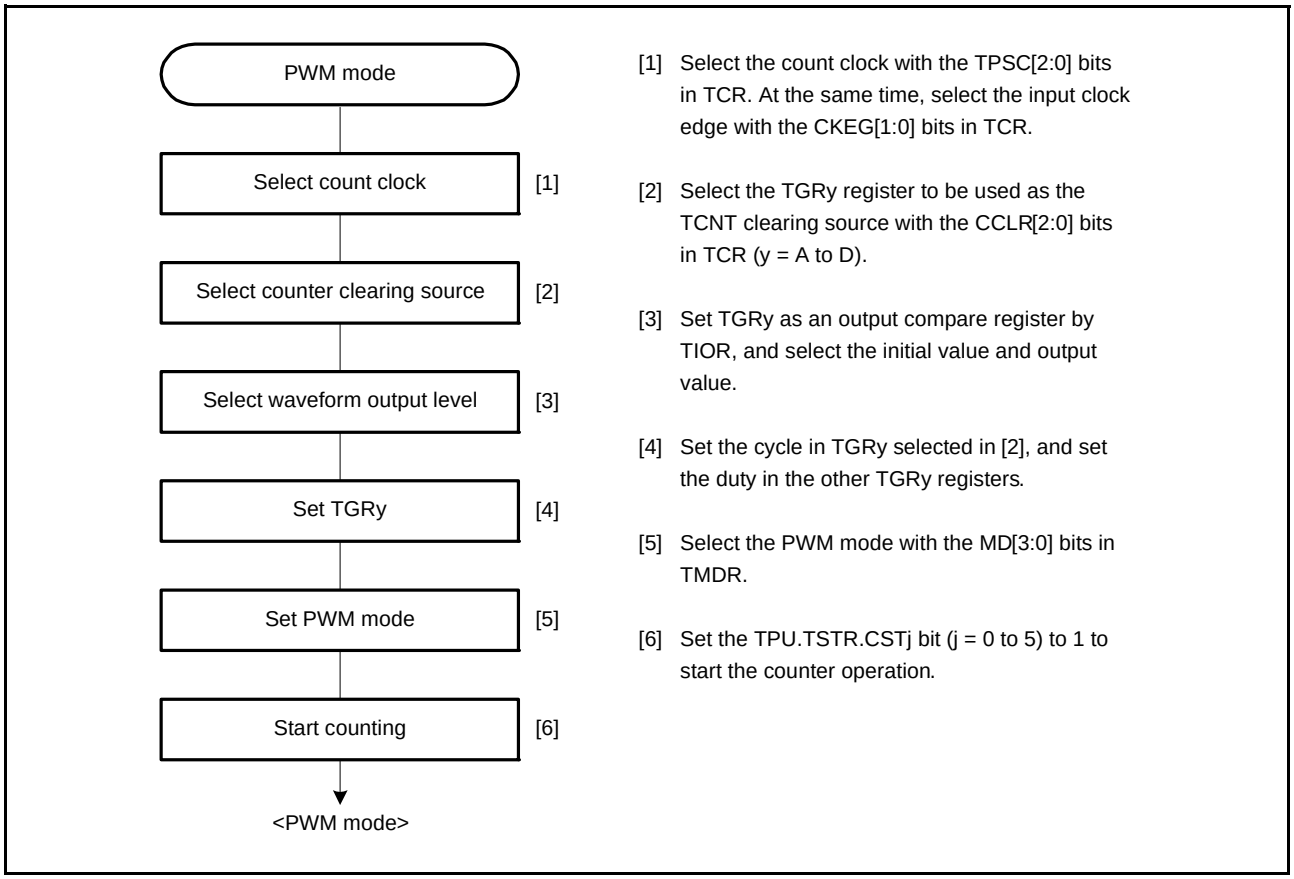


Figure 24.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 24.21 shows an example of PWM mode 1 operation.
In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, low is set for the TGRA initial output value and output value, and high is set as the TPUm.TGRB output value.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty cycle.

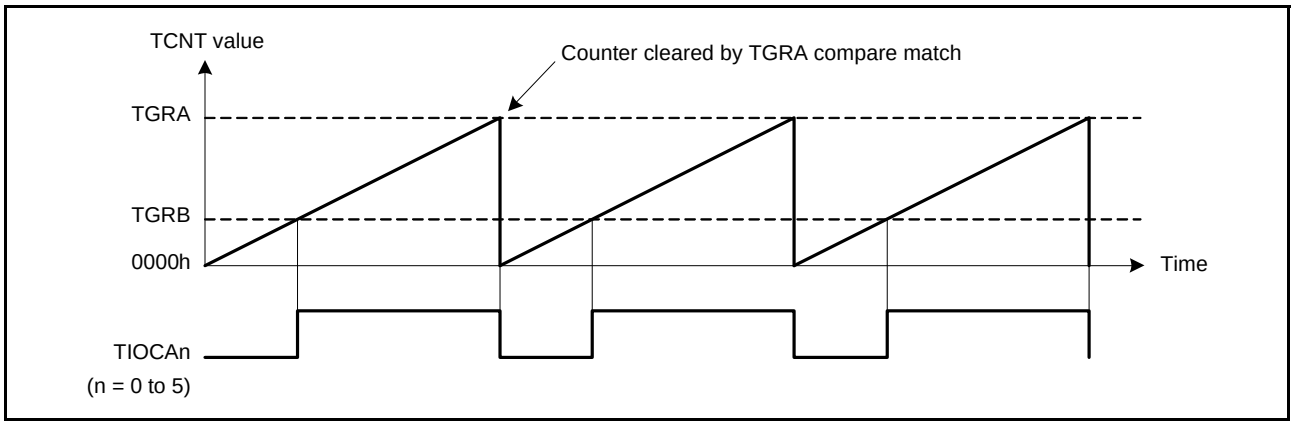


Figure 24.21 Example of PWM Mode Operation (1)

Figure 24.22 shows an example of PWM mode 2 operation.
In this example, synchronous operation is specified for TPU0 and TPU1, TPU1.TGRB compare match is set as the TPUm.TCNT clearing source, and low is set for the initial output value and high for the output value of the other TPUm.TGRy registers (TPU0.TGRA to TPU0.TGRD and TPU1.TGRA), to output a 5-phase PWM waveform.
In this case, the value set in TPU1.TGRB is used as the cycle, and the values set in the other TGRy registers are used as the duty cycle.

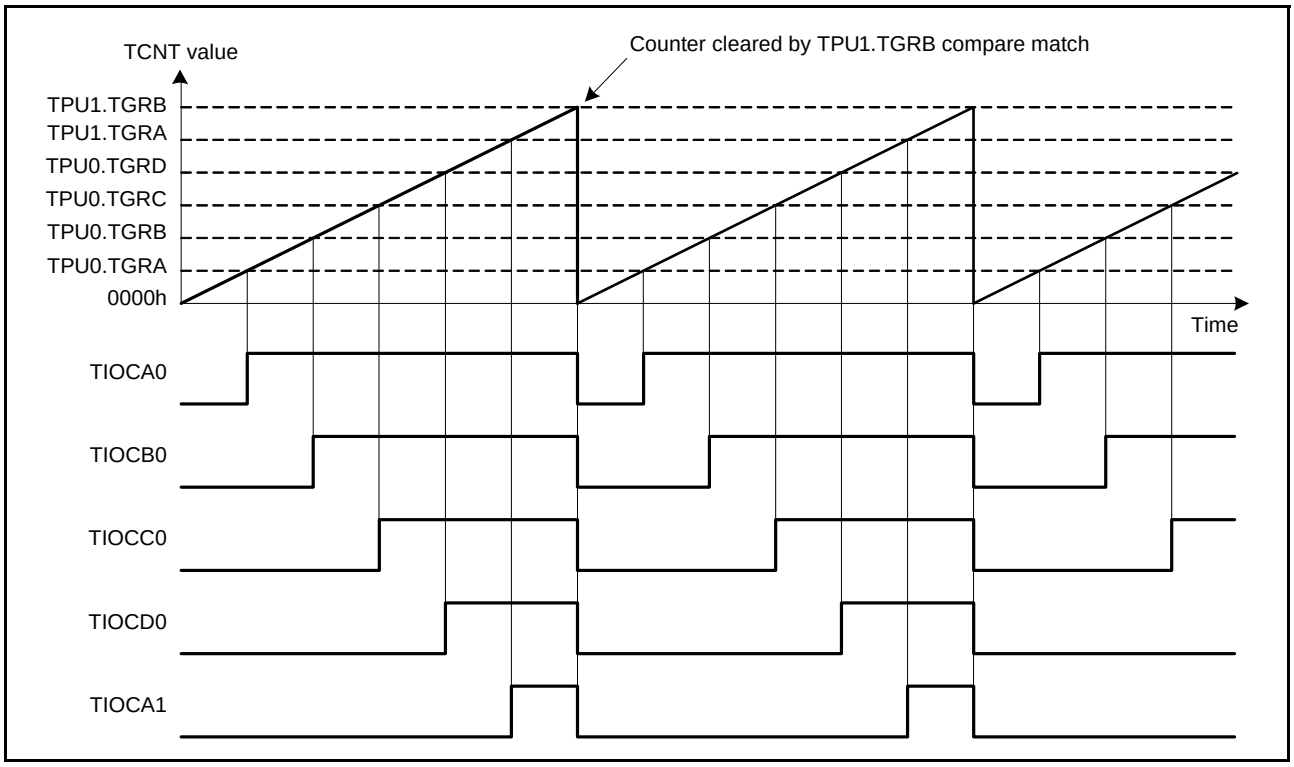


Figure 24.22 Example of PWM Mode Operation (2)

Figure 24.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

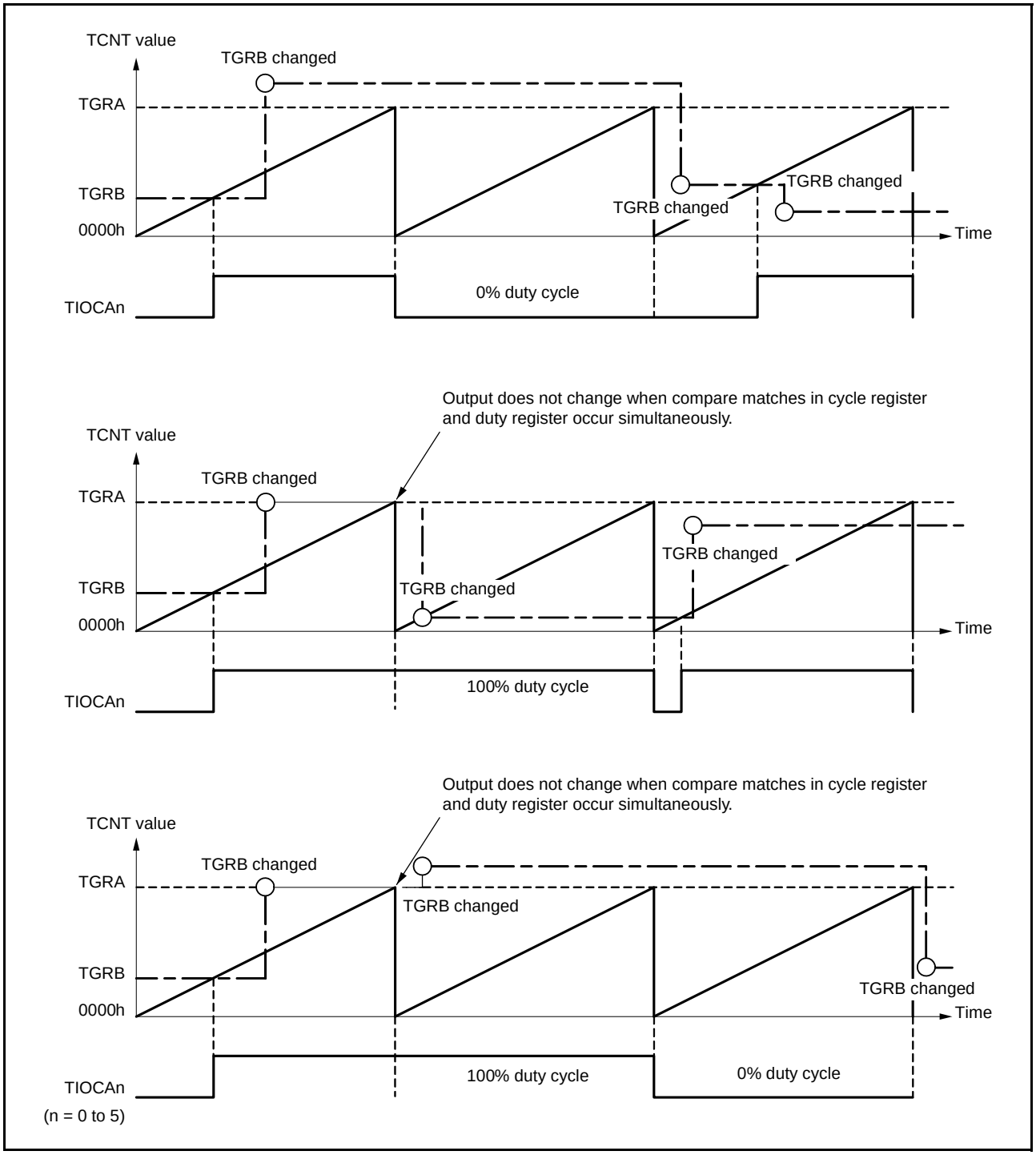


Figure 24.23 Example of PWM Mode Operation (3)

24.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected by the settings for channels 1, 2, 4, and 5, and TPUm.TCNT is incremented/decremented accordingly.
When phase counting mode is set, an external clock is selected as the count clock and TCNT operates as an up-/down-counter regardless of the setting of the TPSC[2:0] bits and CKEG[1:0] bits in TPUm.TCR. However, the lower 2 bits of the CCLR[2:0] bits in TPUm.TCR and the functions of TPUm.TIORH, TPUm.TIORL, TPUm.TIOR, TPUm.TIER, and TPUm.TGRy are valid, and therefore input capture/compare match and interrupt functions are available.
When an overflow occurs while TCNT is counting up, a TCIV interrupt request is generated; when an underflow occurs while TCNT is counting down, a TCIU interrupt request is generated. The TCFD bit in TPUm.TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.
In phase counting mode, the external clock pins TCLKA, TCLKB, TCLKC, and TCLKD can be used as 2-phase encoder pulse input.
Table 24.24 lists the correspondence between external clock pins and channels.

Table 24.24 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channel	**A-Phase**	**B-Phase**
When TPU1 or TPU5 is set to phase counting mode	TCLKA	TCLKB
When TPU2 or TPU4 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 24.24 shows an example of the phase counting mode setting procedure.

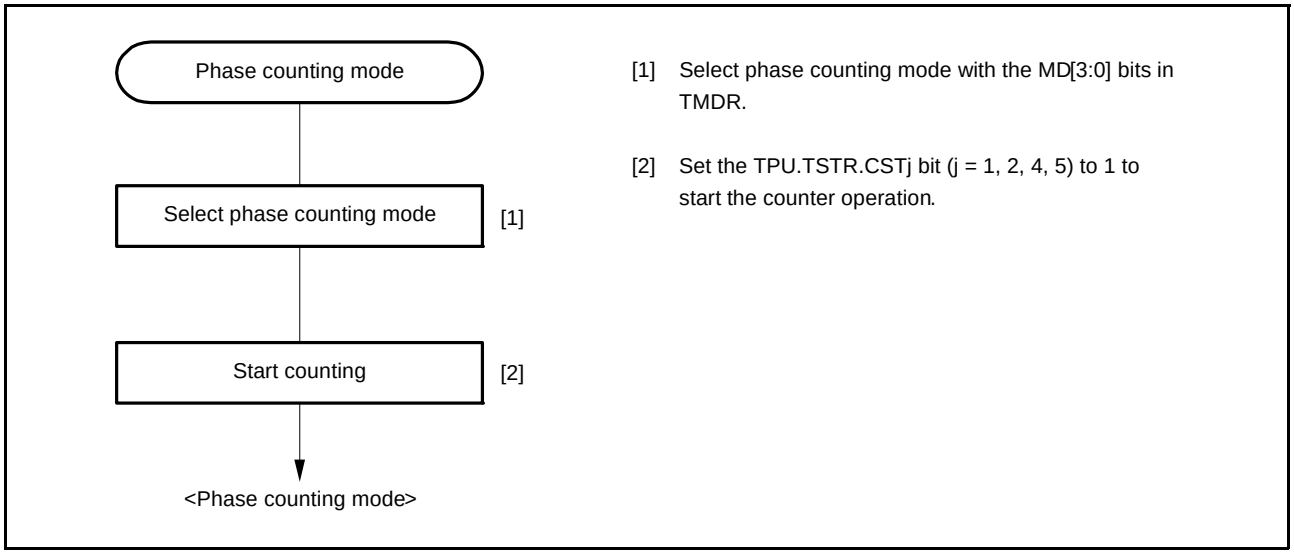


Figure 24.24 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TPUm.TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 24.25 shows an example of phase counting mode 1 operation, and Table 24.25 lists the TPUm.TCNT up-/down-count conditions.

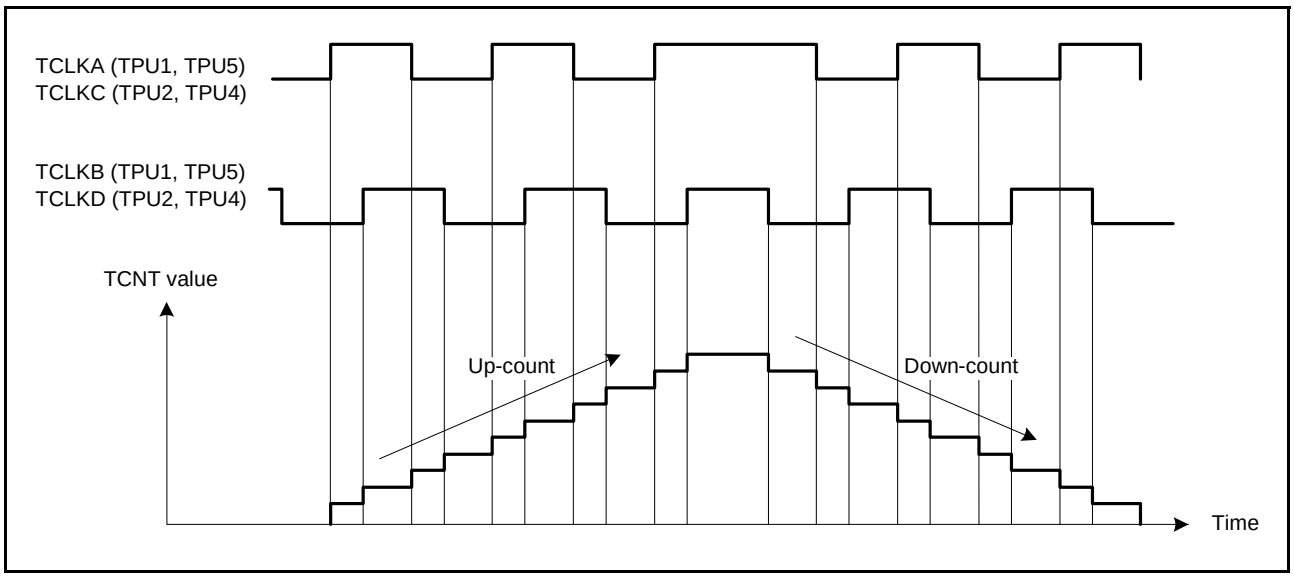


Figure 24.25 Example of Phase Counting Mode 1 Operation

Table 24.25 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	
↓	High	
High	↓	Down-count
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase counting mode 2

Figure 24.26 shows an example of phase counting mode 2 operation, and Table 24.26 lists the TPUm.TCNT up-/down-count conditions.

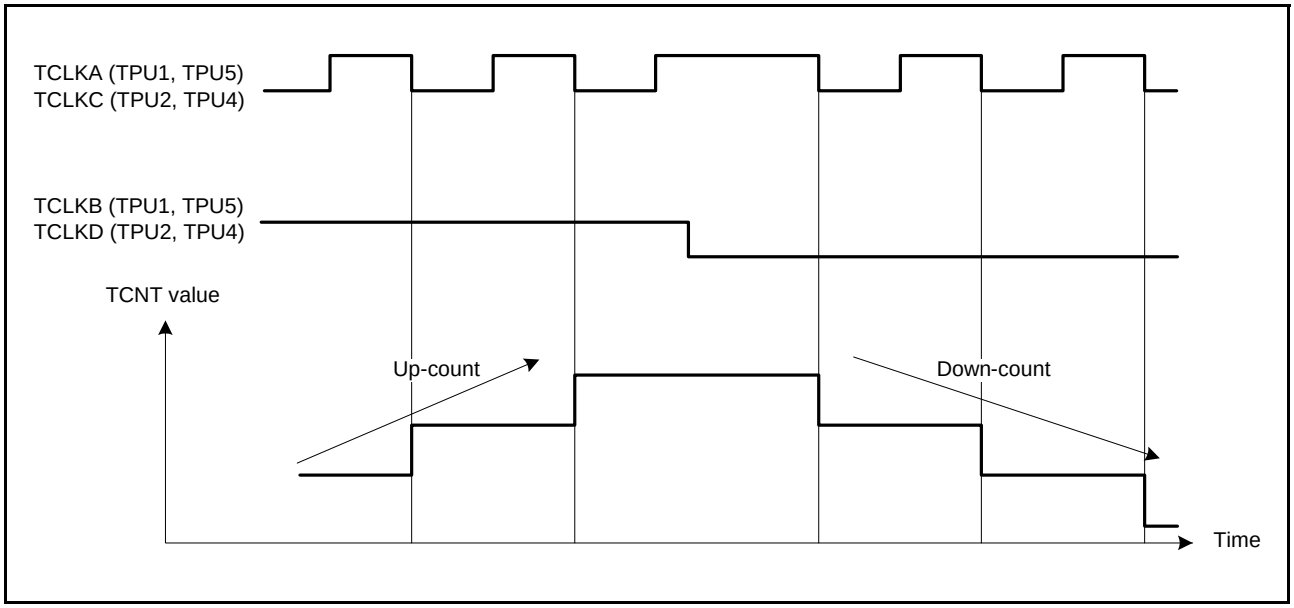


Figure 24.26 Example of Phase Counting Mode 2 Operation

Table 24.26 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Don't care
Low	↑	Don't care
↑	High	Don't care
↓	Low	Down-count

↑ : Rising edge

↓ : Falling edge

(c) Phase counting mode 3

Figure 24.27 shows an example of phase counting mode 3 operation, and Table 24.27 lists the TPUm.TCNT up-/down-count conditions.

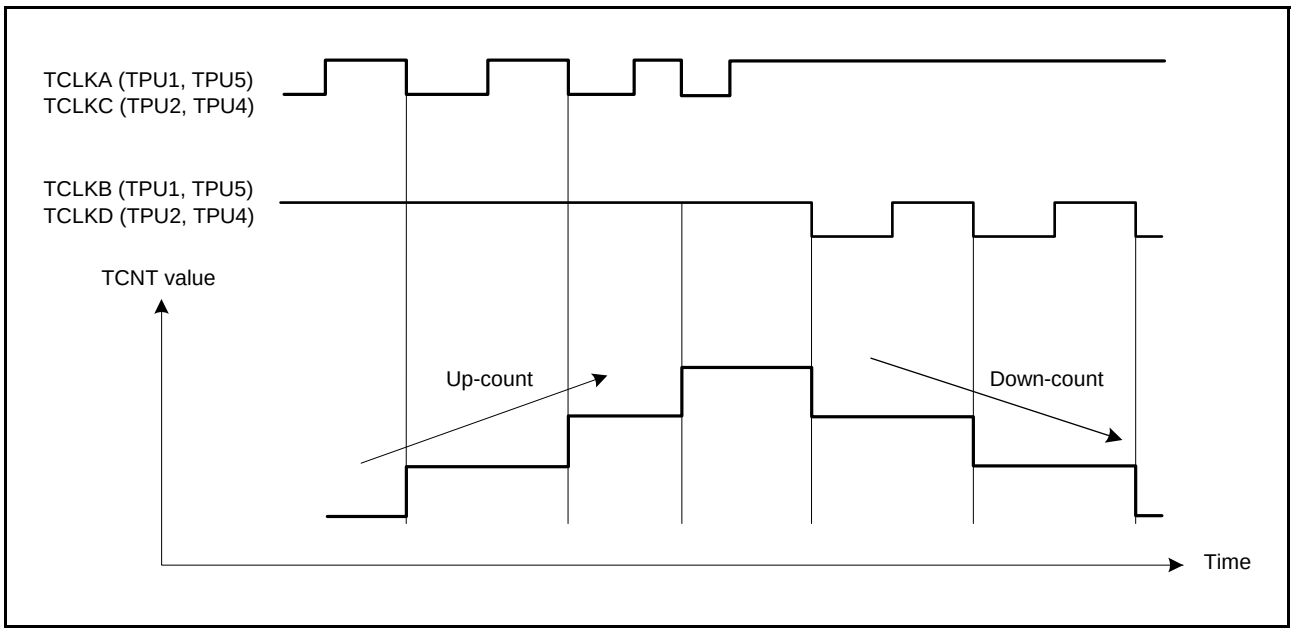


Figure 24.27 Example of Phase Counting Mode 3 Operation

Table 24.27 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Down-count
Low	↑	Don't care
↑	High	Don't care
↓	Low	Don't care

↑ : Rising edge

↓ : Falling edge

(d) Phase counting mode 4

Figure 24.28 shows an example of phase counting mode 4 operation, and Table 24.28 lists the TPUm.TCNT up-/down-count conditions.

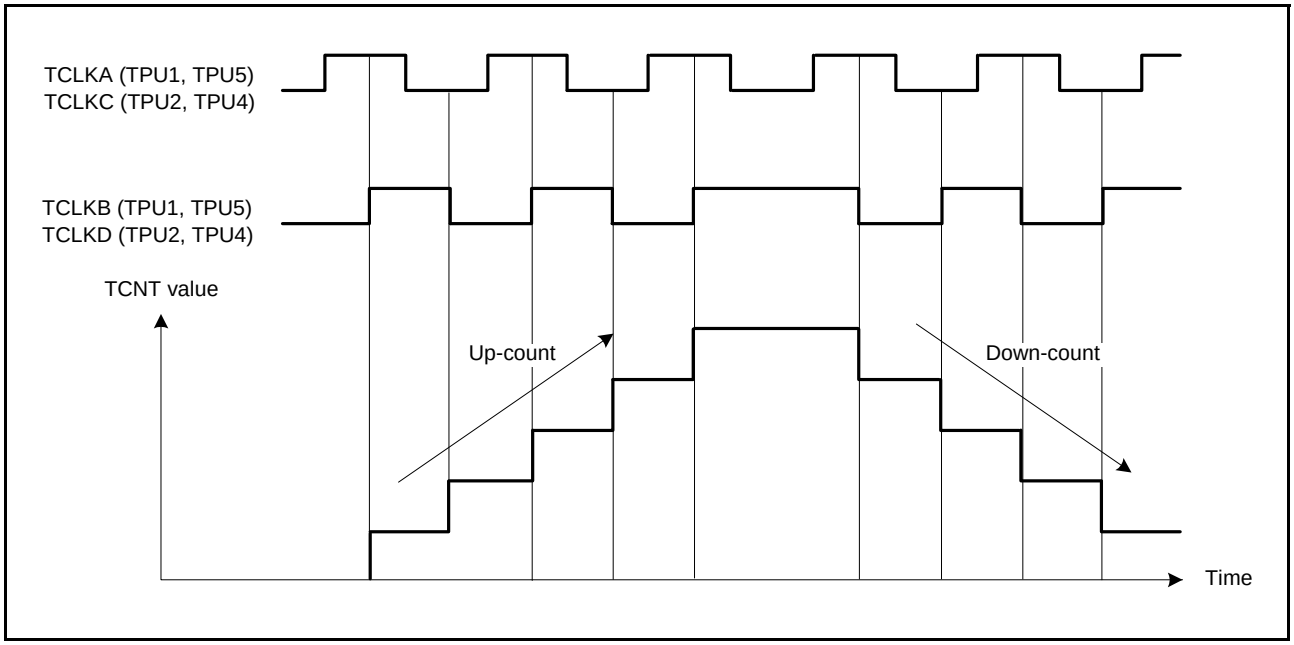


Figure 24.28 Example of Phase Counting Mode 4 Operation

Table 24.28 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	Don't care
↓	High	
High	↓	Down-count
Low	↑	
↑	High	Don't care
↓	Low	

↑ : Rising edge

↓ : Falling edge

24.3.6.1 Phase Counting Mode Application Example

Figure 24.29 shows an example in which phase counting mode is set for TPU1, and TPU1 is coupled with TPU0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

TPU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to the TCLKA and TCLKB pins.

TPU0 operates with TPU0.TCNT clearing by TPU0.TGRC compare match; TPU0.TGRA and TPU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. TPU0.TGRB is used for input capture, with TPU0.TGRB and TPU0.TGRD operating in buffer mode. The TPU1 count clock is specified as the TPU0.TGRB input capture source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TPU1.TGRA and TPU1.TGRB for TPU1 are specified for input capture, TPU0.TGRA and TPU0.TGRC compare matches are selected as the input capture source, and the up-/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

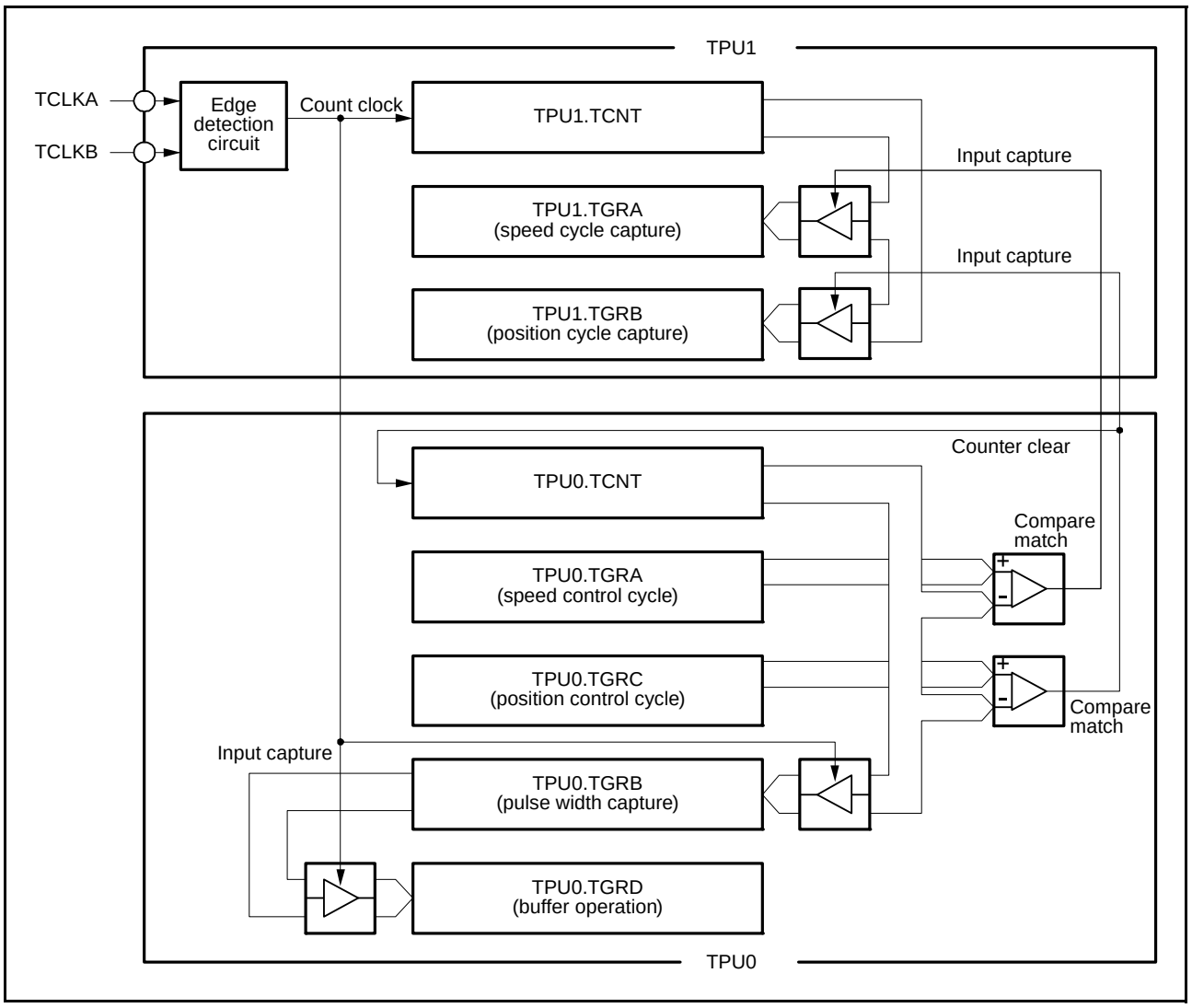


Figure 24.29 Phase Counting Mode Application Example

24.3.7 Noise Filters

Each pin for use in input capture by TPU is equipped with a noise filter. The noise filter samples the level on the pin three times at the selected sampling interval, conveys the level to the internal circuits if the samples match, and continues to convey that level until the other level is sampled from the pins three times in a row. The noise filter function can be enabled or disabled for each pin. Furthermore, sampling clock settings can be made for each channel.
Figure 24.30 is a timing chart for the noise filter.

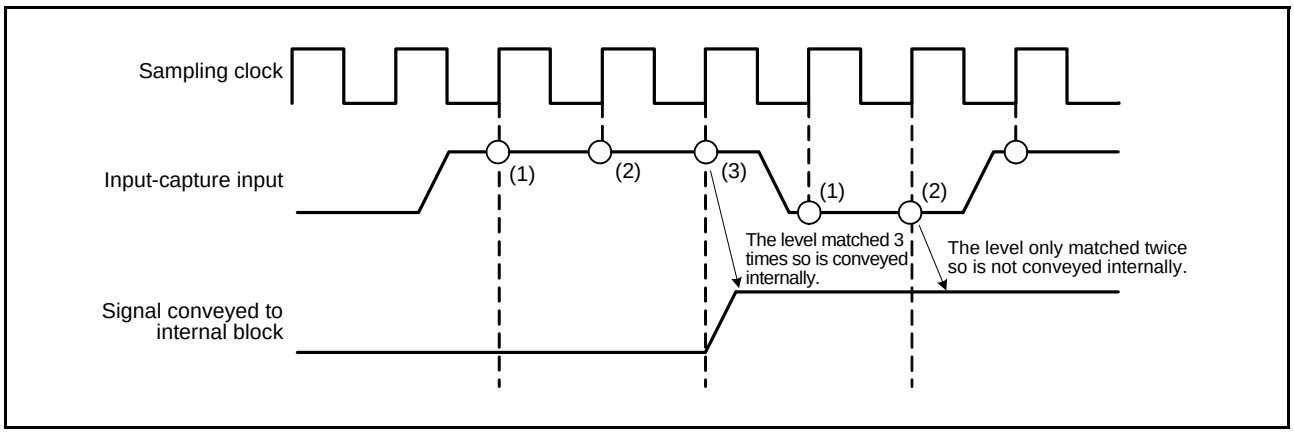


Figure 24.30 Timing Chart for the Noise Filter

24.4 Interrupt Sources

There are three kinds of TPU interrupt sources: TPUm.TGRy input capture/compare match, TPUm.TCNT overflow, and TPUm.TCNT underflow.

Relative channel priority levels can be changed by the interrupt controller, but the priority within a channel is fixed. For details, see section 14, Interrupt Controller (ICUb).

Table 24.29 lists the TPU interrupt sources.

Table 24.29 TPU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
TPU0	TGI0A	TPU0.TGRA input capture/compare match	Possible	Possible
	TGI0B	TPU0.TGRB input capture/compare match	Possible	Not possible
	TGI0C	TPU0.TGRC input capture/compare match	Possible	Not possible
	TGI0D	TPU0.TGRD input capture/compare match	Possible	Not possible
	TCI0V	TPU0.TCNT overflow	Not possible	Not possible
TPU1	TGI1A	TPU1.TGRA input capture/compare match	Possible	Possible
	TGI1B	TPU1.TGRB input capture/compare match	Possible	Not possible
	TCI1V	TPU1.TCNT overflow	Not possible	Not possible
	TCI1U	TPU1.TCNT underflow	Not possible	Not possible
TPU2	TGI2A	TPU2.TGRA input capture/compare match	Possible	Possible
	TGI2B	TPU2.TGRB input capture/compare match	Possible	Not possible
	TCI2V	TPU2.TCNT overflow	Not possible	Not possible
	TCI2U	TPU2.TCNT underflow	Not possible	Not possible
TPU3	TGI3A	TPU3.TGRA input capture/compare match	Possible	Possible
	TGI3B	TPU3.TGRB input capture/compare match	Possible	Not possible
	TGI3C	TPU3.TGRC input capture/compare match	Possible	Not possible
	TGI3D	TPU3.TGRD input capture/compare match	Possible	Not possible
	TCI3V	TPU3.TCNT overflow	Not possible	Not possible
TPU4	TGI4A	TPU4.TGRA input capture/compare match	Possible	Possible
	TGI4B	TPU4.TGRB input capture/compare match	Possible	Not possible
	TCI4V	TPU4.TCNT overflow	Not possible	Not possible
	TCI4U	TPU4.TCNT underflow	Not possible	Not possible
TPU5	TGI5A	TPU5.TGRA input capture/compare match	Possible	Possible
	TGI5B	TPU5.TGRB input capture/compare match	Possible	Not possible
	TCI5V	TPU5.TCNT overflow	Not possible	Not possible
	TCI5U	TPU5.TCNT underflow	Not possible	Not possible

Note: This table lists the initial state immediately after a reset. The relative channel priority levels can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested when the TGIEy bit (y = A, B, C, D) in TPUm.TIER is set to 1 by the occurrence of a TPUm.TGRy input capture/compare match on a channel. The TPU has 16 input capture/compare match interrupts, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

(2) Overflow Interrupt

An interrupt is requested when the TCIEV bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT overflow on a channel. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested when the TCIEU bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT underflow on a channel. The TPU has four underflow interrupts, one each for TPU1, TPU2, TPU4, and TPU5.

24.5 DTC Activation

The DTC can be activated by the TPUm.TGRy input capture/compare match interrupt of each channel. For details, see section 18, Data Transfer Controller (DTCa).
A total of 16 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

24.6 DMAC Activation

The DMAC can be activated by the TPUm.TGRA input capture/compare match interrupt of each channel. For details, see section 17, DMA Controller (DMACA).
A total of six TPUm.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

24.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUm.TGRA input capture/compare match for each channel.
When the TTGE bit in TPUm.TIER is set to 1, the TPU requests the A/D converter to start A/D conversion by the occurrence of a TPUm.TGRA input capture/compare match on a particular channel.

24.8 Operation Timing

24.8.1 Input/Output Timing

(1) TPUm.TCNT Count Timing

Figure 24.31 shows TPUm.TCNT count timing in internal clock operation, and Figure 24.32 shows TCNT count timing in external clock operation.

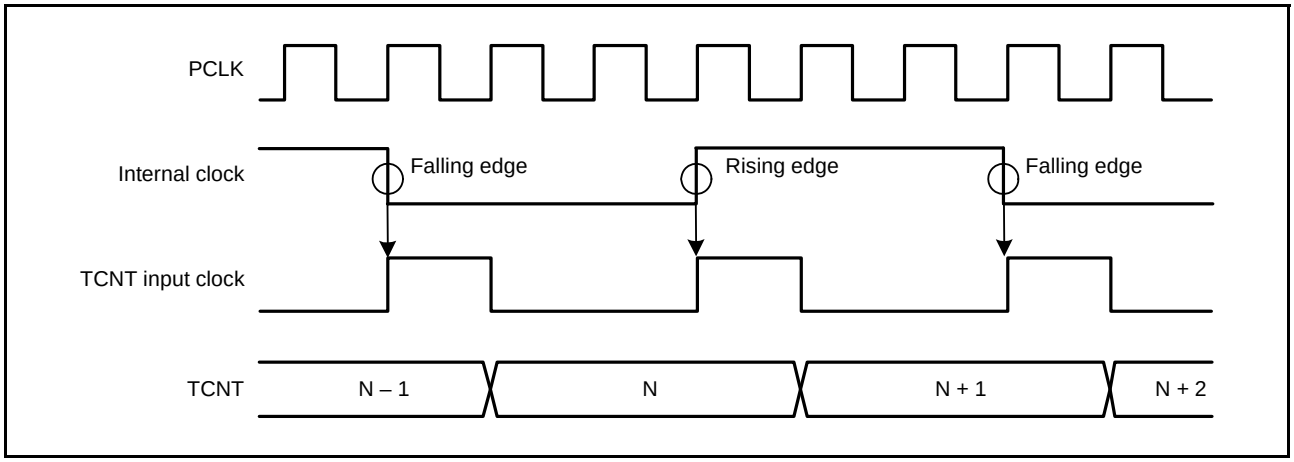


Figure 24.31 Count Timing in Internal Clock Operation

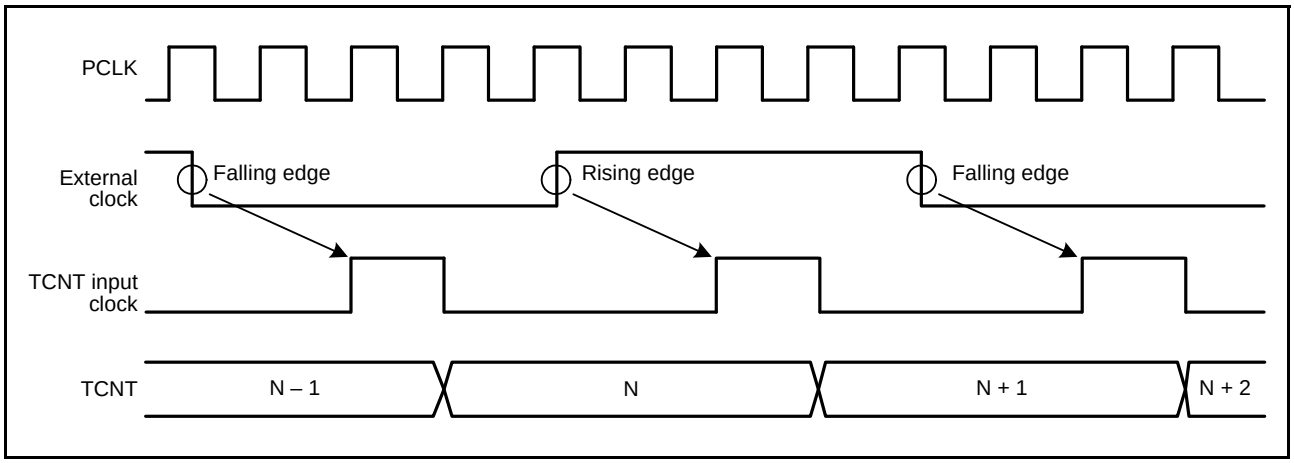


Figure 24.32 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TPUm.TCNT and TPUm.TGRy match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is output to the output compare output pin TIOCyn (y = A to D; n = 0 to 5). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 24.33 shows output compare output timing.

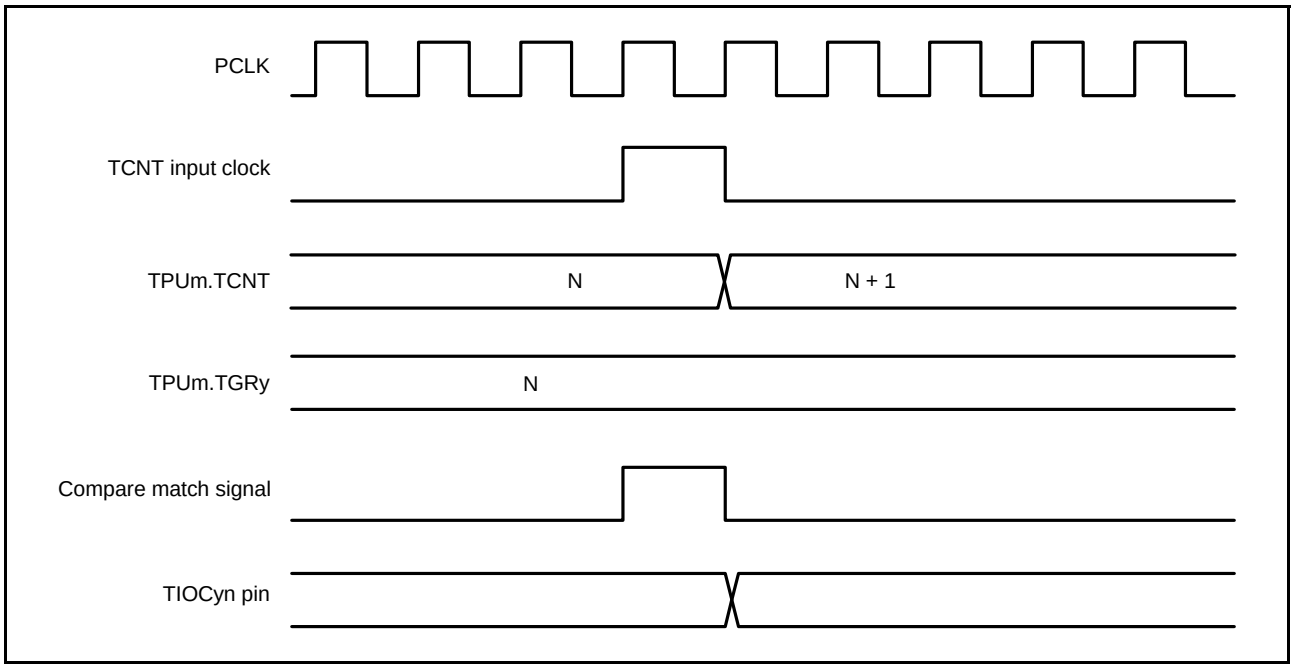


Figure 24.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 24.34 shows input capture signal timing.

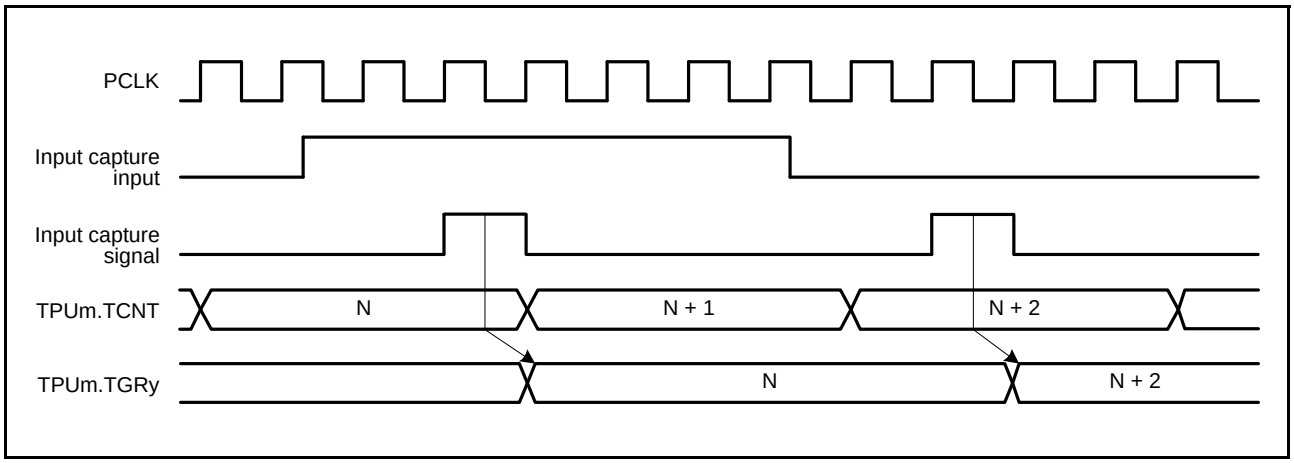


Figure 24.34 Input Capture Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 24.35 shows the timing when counter clearing by compare match occurrence is specified, and Figure 24.36 shows the timing when counter clearing by input capture occurrence is specified.

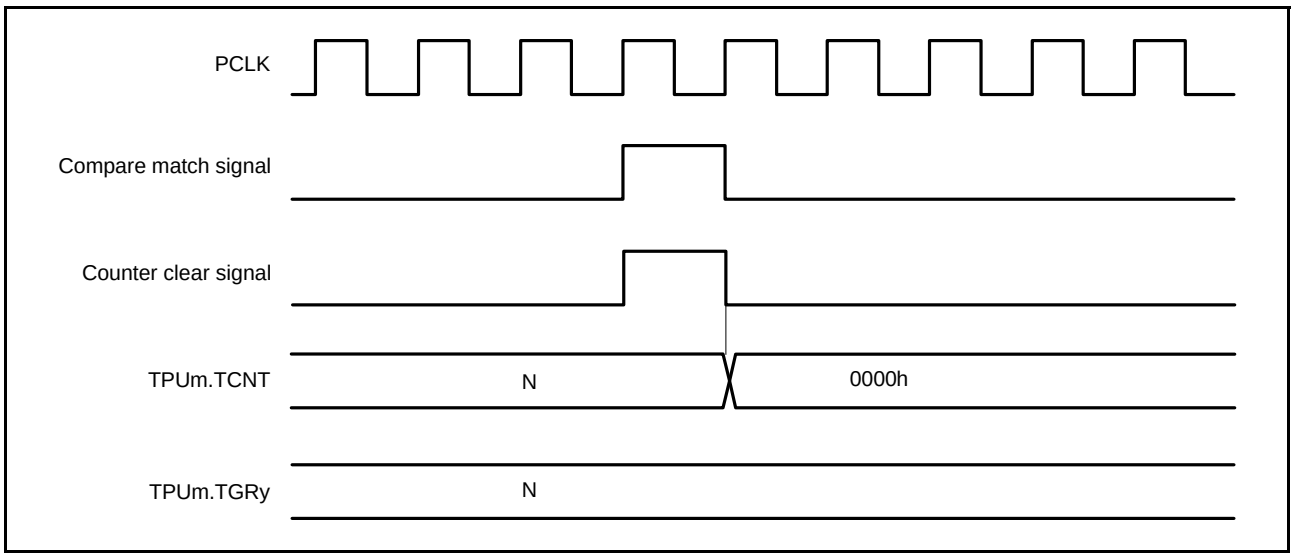


Figure 24.35 Counter Clear Timing (Compare Match)

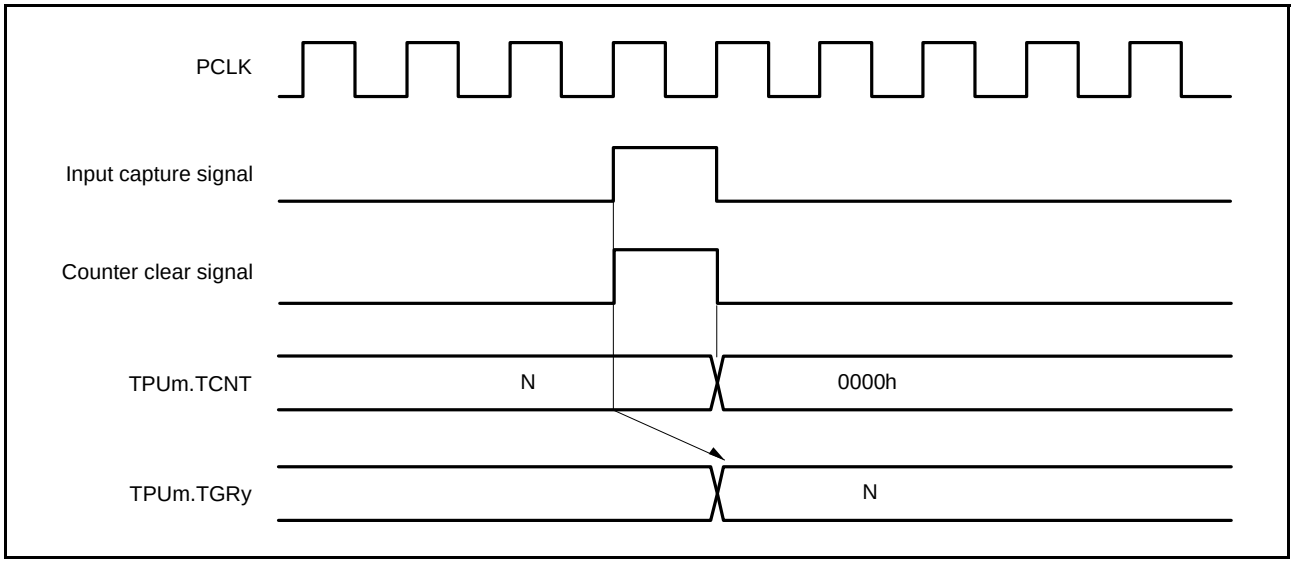


Figure 24.36 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figure 24.37 and Figure 24.38 show the timings in buffer operation.

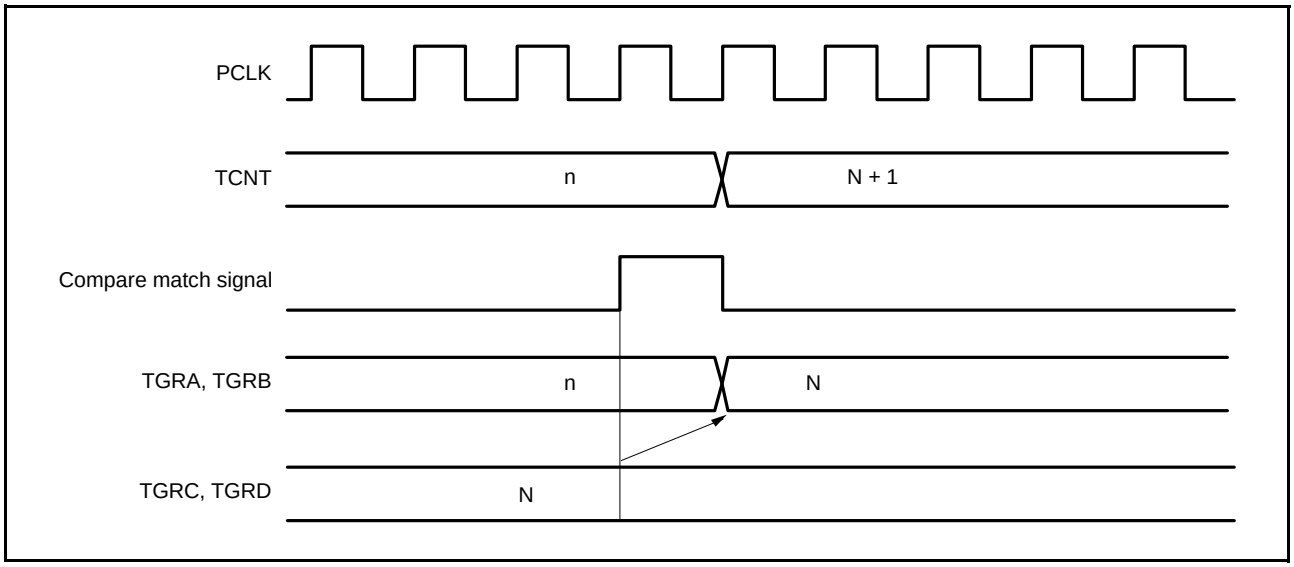


Figure 24.37 Buffer Operation Timing (Compare Match)

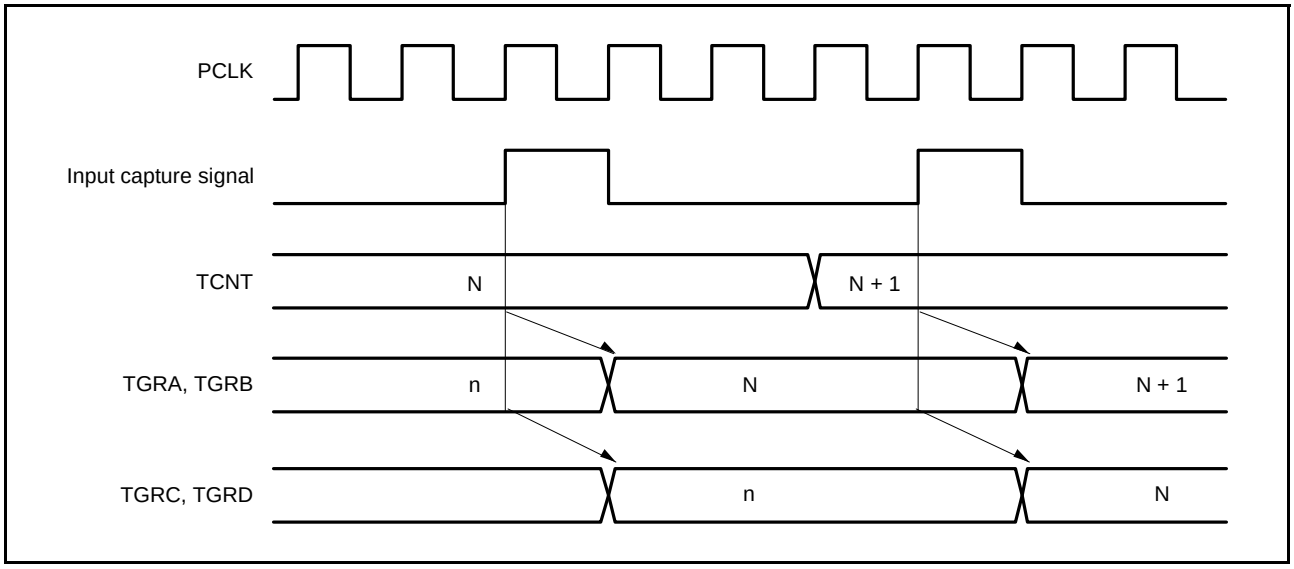


Figure 24.38 Buffer Operation Timing (Input Capture)

24.8.2 Interrupt Signal Timing

(1) Timing of Interrupt Signal Setting on Compare Match

Figure 24.39 shows the timing for setting the interrupt signal by compare match occurrence.

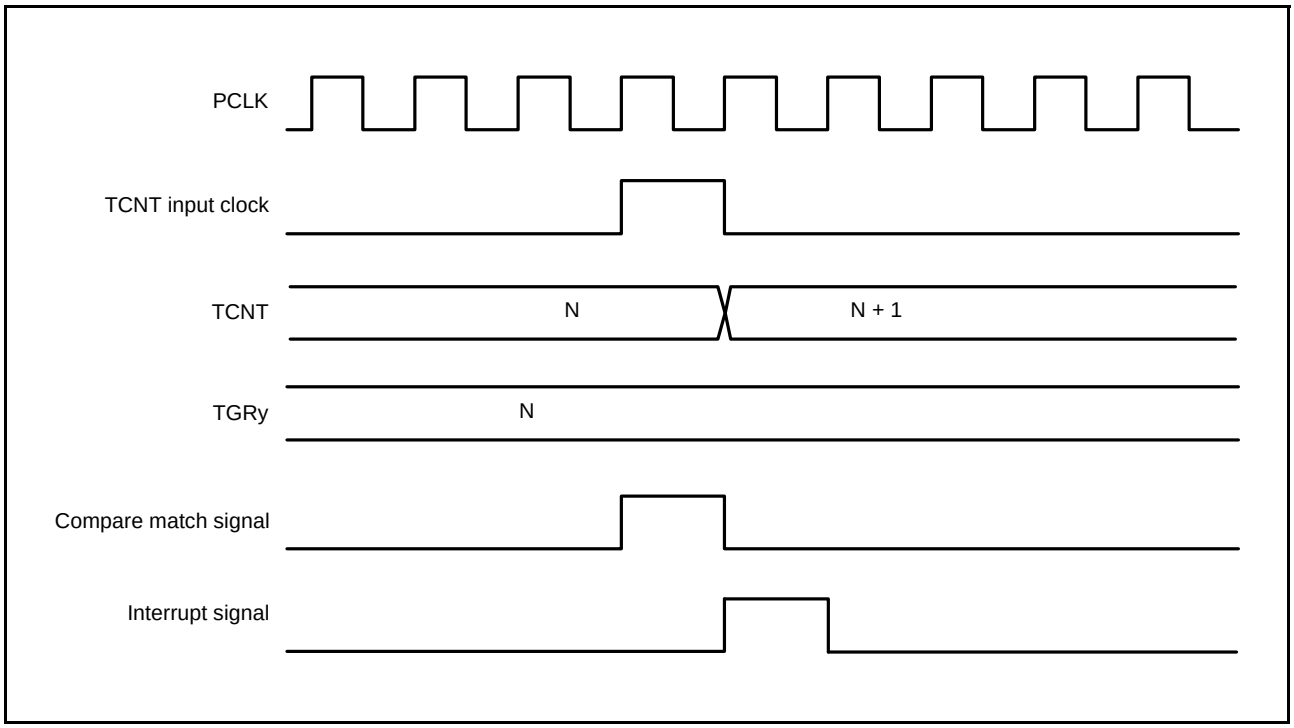


Figure 24.39 TGImy Interrupt Timing (Compare Match)

(2) Timing of Interrupt Signal Setting on Input Capture

Figure 24.40 shows the timing for setting the interrupt signal by input capture occurrence.

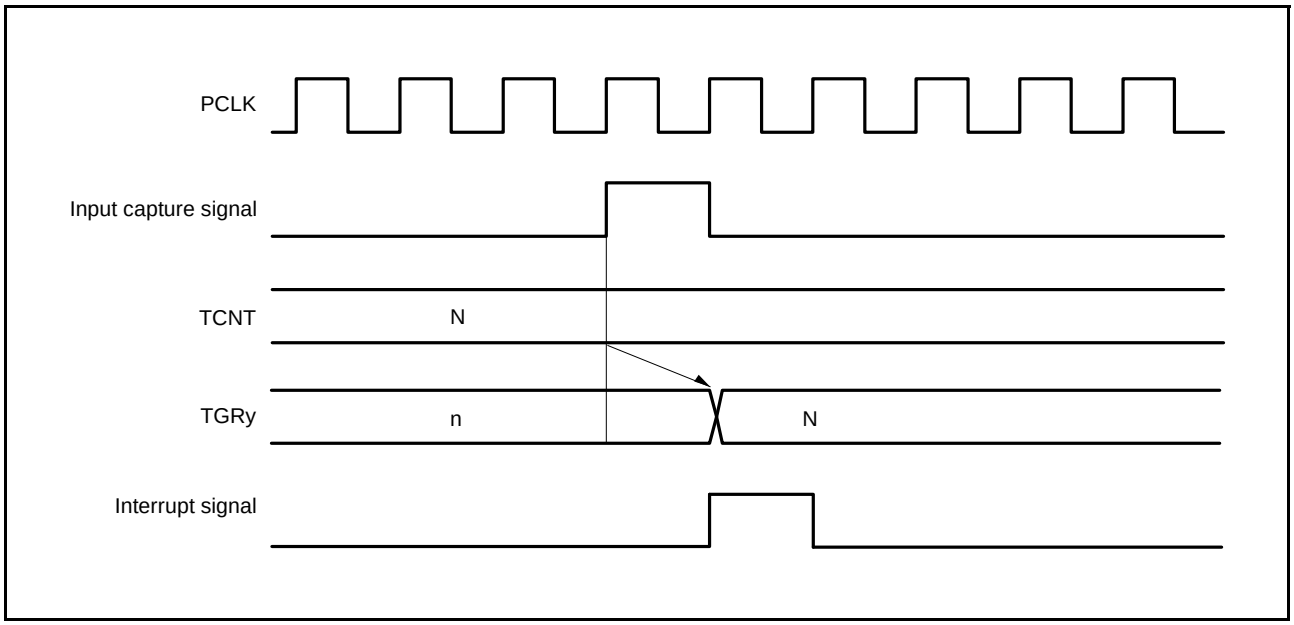


Figure 24.40 TGImy Interrupt Timing (Input Capture)

(3) Timing of TCImV/TCImU Interrupt Signal Setting

Figure 24.41 shows the timing for generating the TCImV interrupt signal by overflow occurrence.
Figure 24.42 shows the timing for generating the TCImU interrupt signal by underflow occurrence.

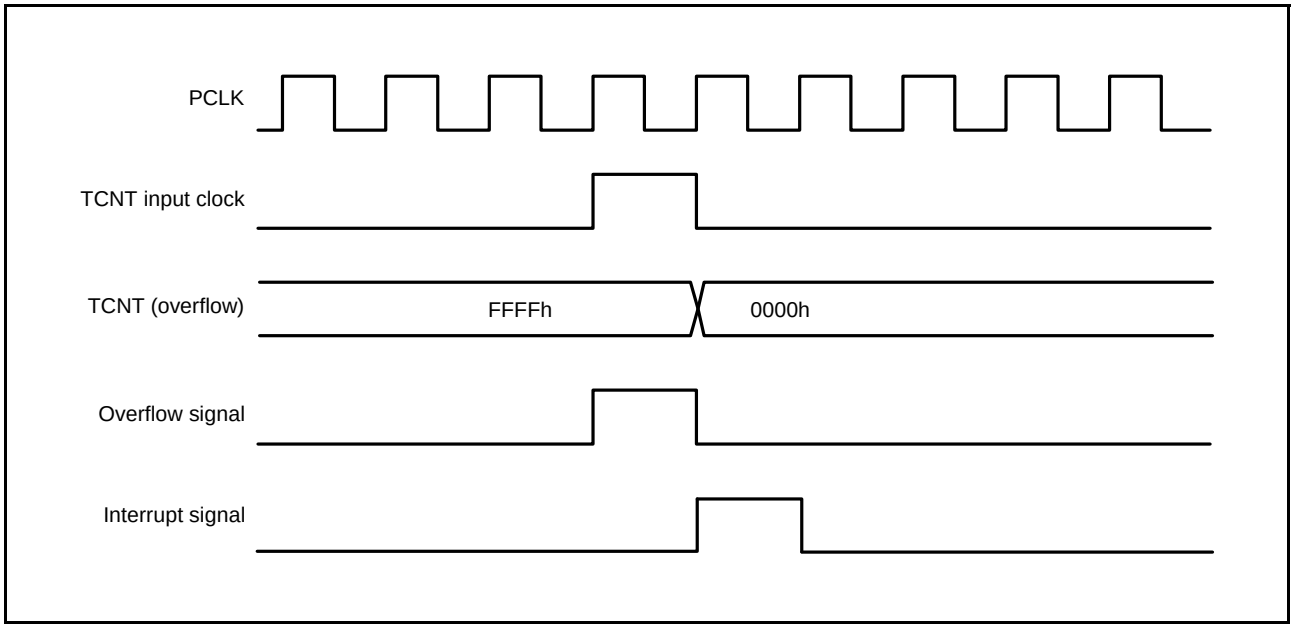


Figure 24.41 TCImV Interrupt Setting Timing

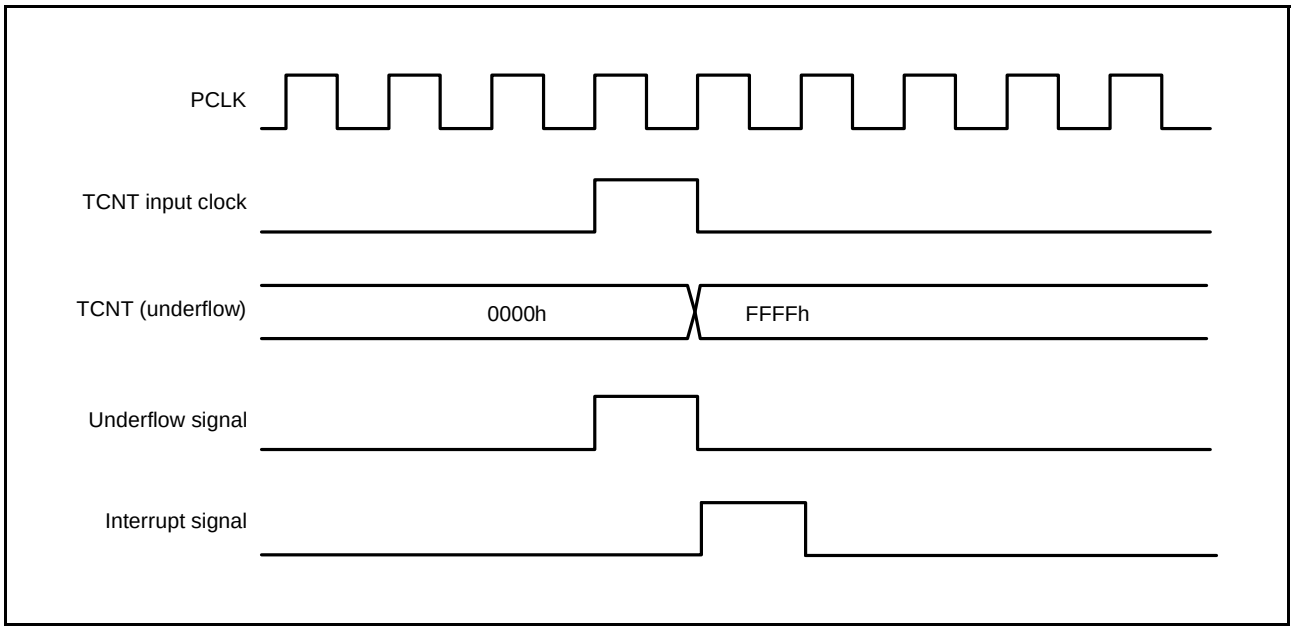


Figure 24.42 TCImU Interrupt Setting Timing

24.9 Usage Notes

24.9.1 Module Stop Function Setting

Operation of the TPU can be disabled or enabled using the module stop control register. The TPU does not operate with the initial setting. Register access is enabled by releasing the module stop state. For details, see section 11, Low Power Consumption.

24.9.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles in the case of single-edge detection, and at least 2.5 PCLK cycles in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 24.43 shows the input clock conditions in phase counting mode.

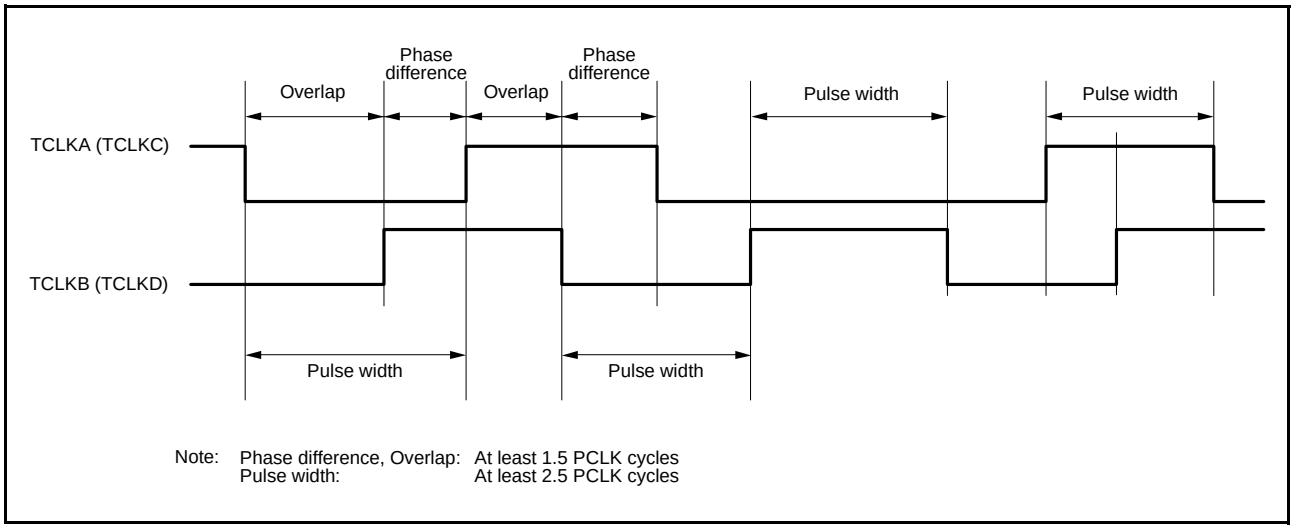


Figure 24.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

24.9.3 Notes on Cycle Setting

When counter clearing by compare match is set, TPUm.TCNT is cleared in the final state in which it matches the TPUm.TGRy value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{f_{TCNT\_CLK}}{(N + 1)}$$

f: Counter frequency
f_{TCNT_CLK}: Count clock frequency
N: TGRy set value

24.9.4 Conflict between TPUm.TCNT Write and Clear Operations

If the counter clearing signal is generated in a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 24.44 shows the timing in this case.

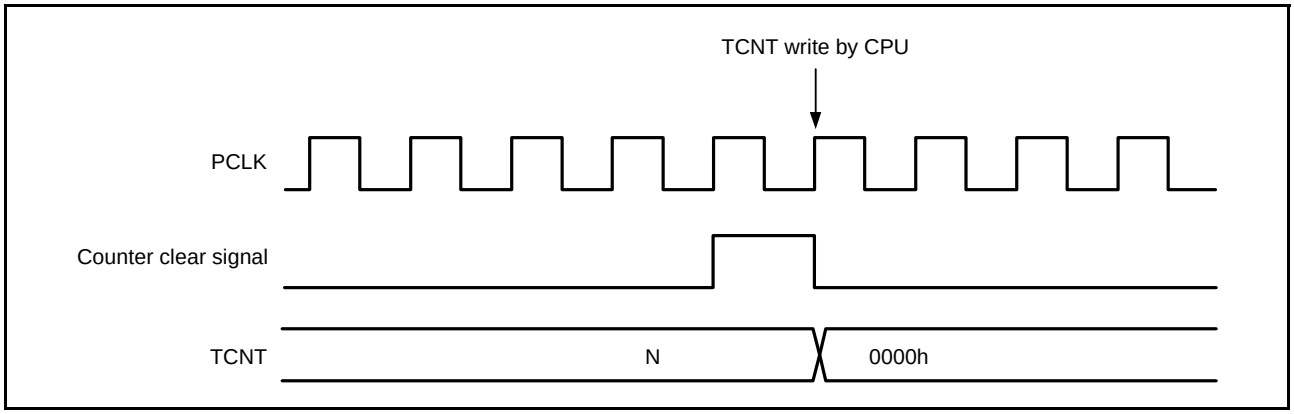


Figure 24.44 Conflict between TPUm.TCNT Write and Clear Operations

24.9.5 Conflict between TPUm.TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 24.45 shows the timing in this case.

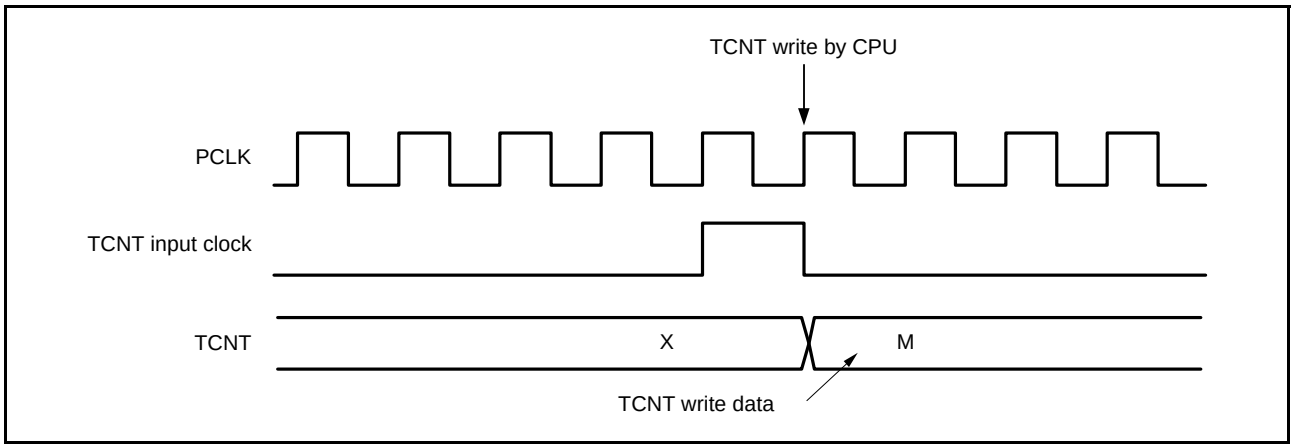


Figure 24.45 Conflict between TPUm.TCNT Write and Increment Operations

24.9.6 Conflict between TPUm.TGRy Write and Compare Match

If a compare match occurs in a TGRy write cycle, the TGRy write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.
Figure 24.46 shows the timing in this case.

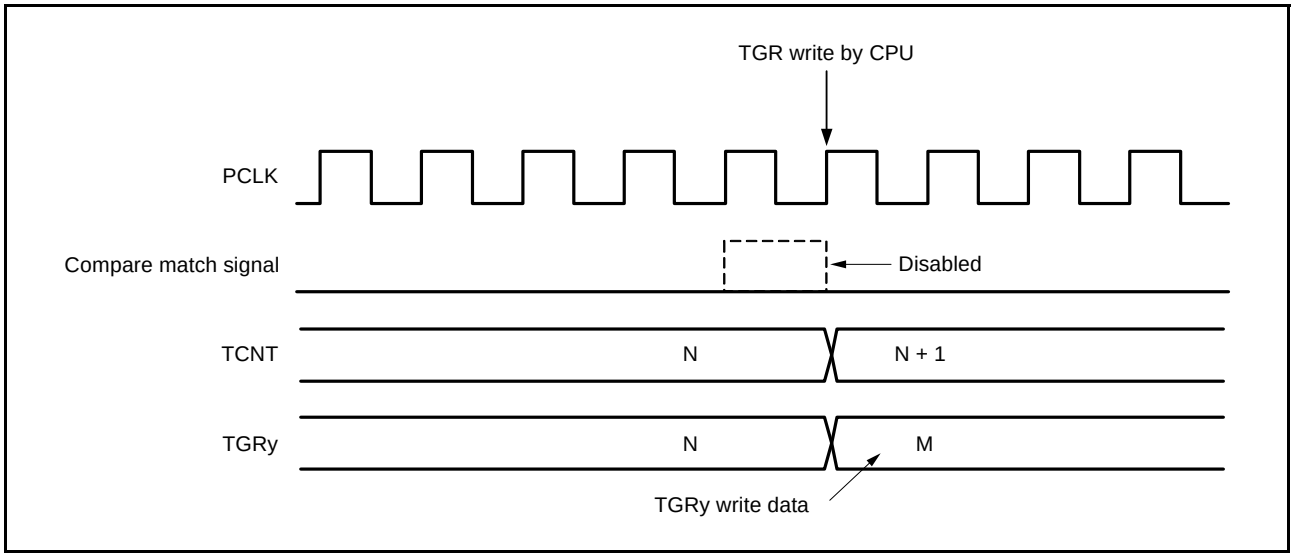


Figure 24.46 Conflict between TPUm.TGRy Write and Compare Match

24.9.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in a TPUm.TGRy write cycle, the data transferred to TGRy by the buffer operation will be the data before writing.
Figure 24.47 shows the timing in this case.

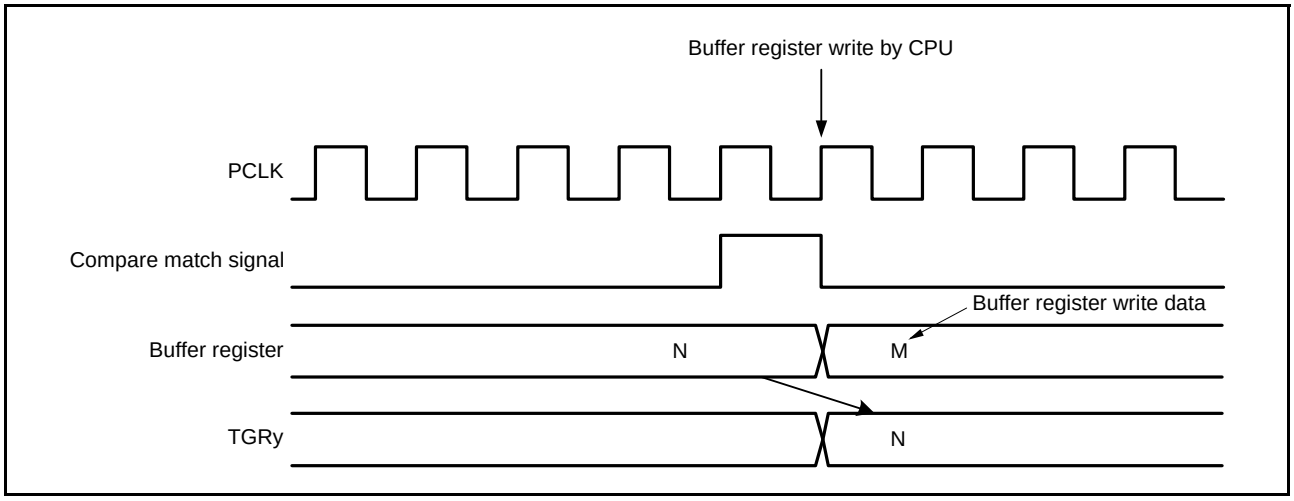


Figure 24.47 Conflict between Buffer Register Write and Compare Match

24.9.8 Conflict between TPUm.TGRy Read and Input Capture

If the input capture signal is generated in a TGRy read cycle, the data that is read will be the data before input capture transfer.

Figure 24.48 shows the timing in this case.

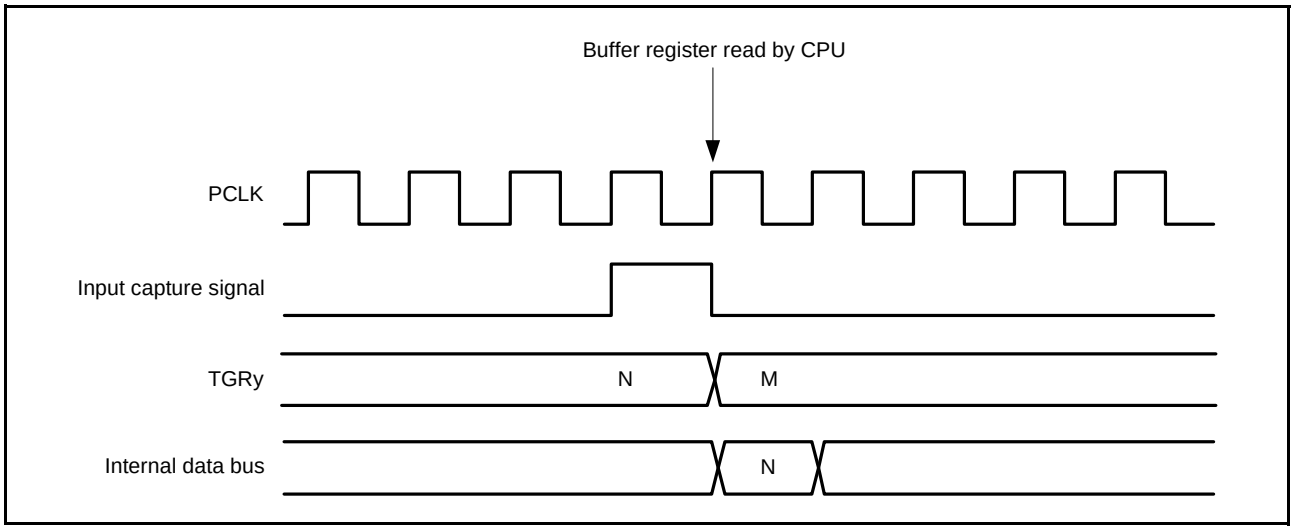


Figure 24.48 Conflict between TPUm.TGRy Read and Input Capture

24.9.9 Conflict between TPUm.TGRy Write and Input Capture

If the input capture signal is generated in a TGRy write cycle, the input capture operation takes precedence and the write to TGRy is not performed. Figure 24.49 shows the timing in this case.

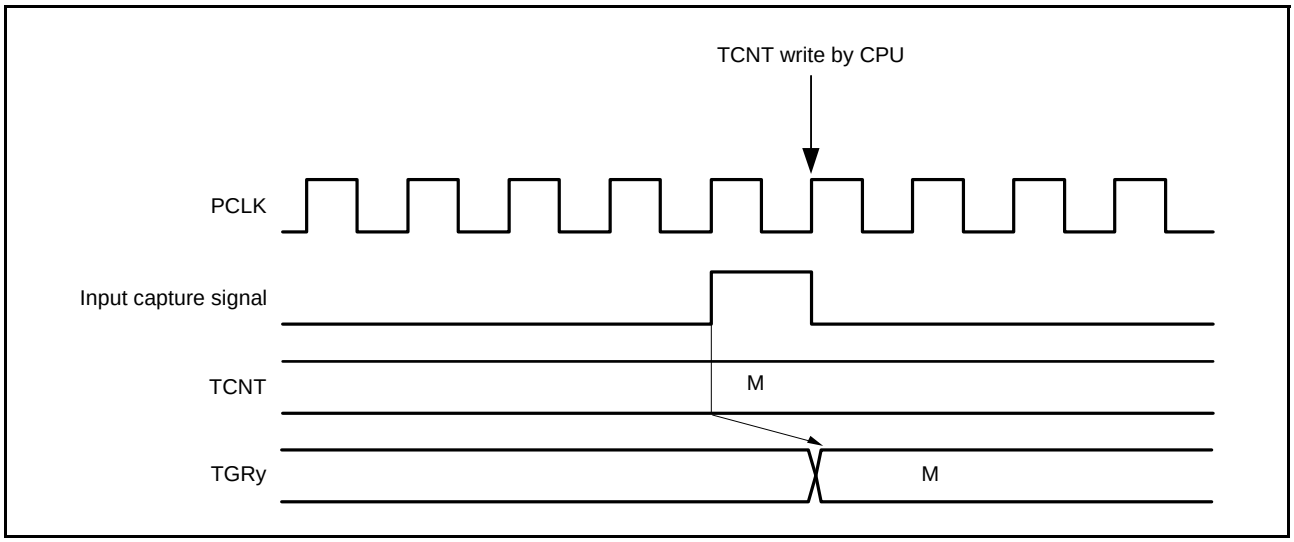


Figure 24.49 Conflict between TPUm.TGRy Write and Input Capture

24.9.10 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 24.50 shows the timing in this case.

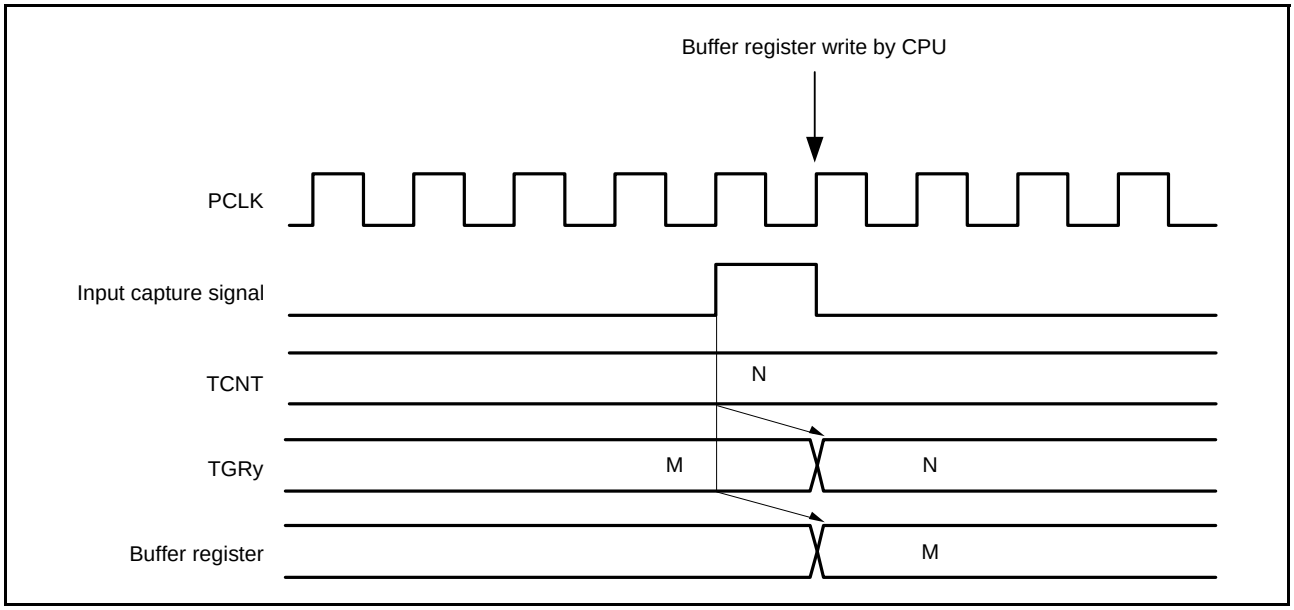


Figure 24.50 Conflict between Buffer Register Write and Input Capture

24.9.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, TPUm.TCNT is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.
Figure 24.51 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.

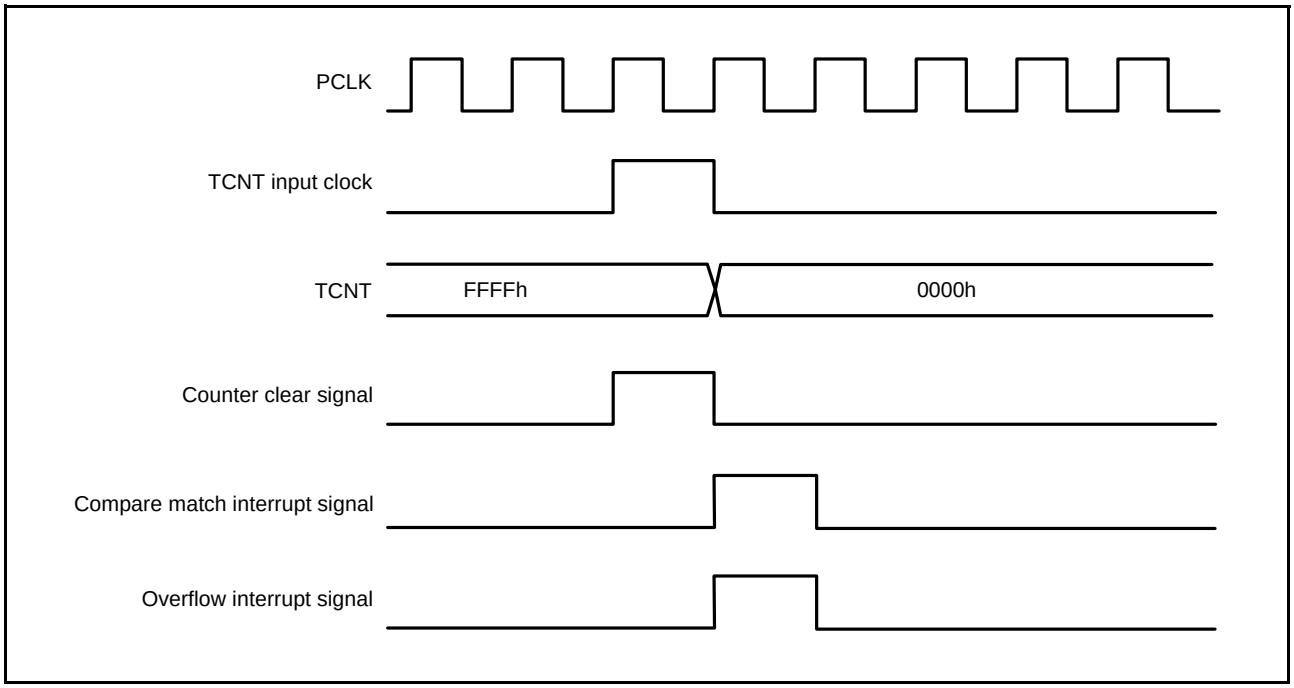


Figure 24.51 Conflict between Overflow and Counter Clearing

24.9.12 Conflict between TPUm.TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in a TCNT write cycle, the TCNT write takes precedence. Figure 24.52 shows the operation timing when there is conflict between TCNT write and overflow.

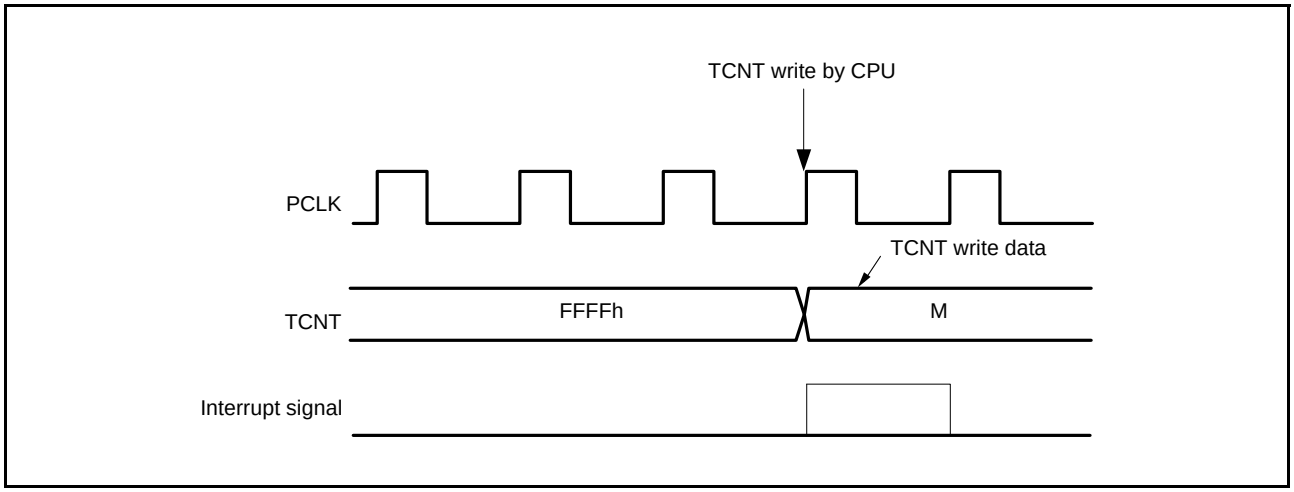


Figure 24.52 Conflict between TPUm.TCNT Write and Overflow

24.9.13 Multiplexing of I/O Pins

In this MCU, the TCLKA input pin is multiplexed with the TIOCB5 I/O pin, the TCLKB input pin with the TIOCB2 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, the TCLKD input pin with the TIOCB0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, and the TCLKD input pin with the TIOCD3 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

24.9.14 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0000h, PCLK/1 is set as the count clock, and compare match is set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 24.53 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

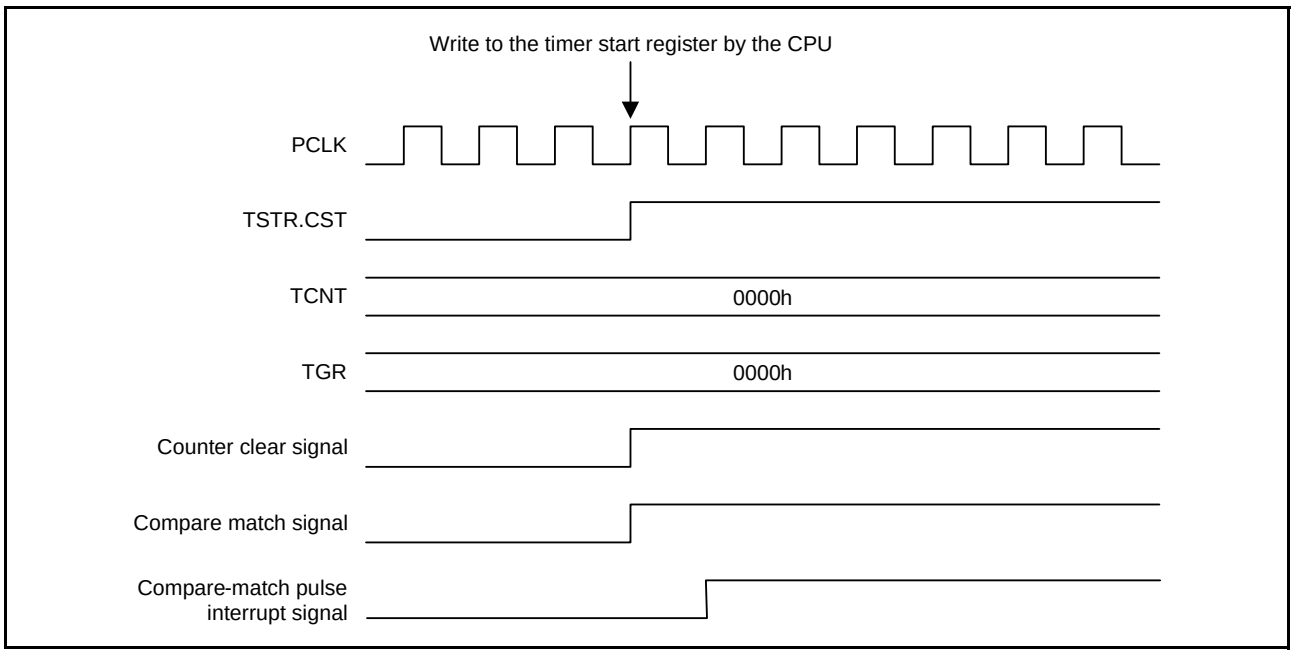


Figure 24.53 Continuous Output of Compare-Match Pulse Interrupt Signal

24.9.15 Continuous Output of Input-Capture Pulse Interrupt Signal

When input-capture signal is set on both edges and when the pulse width of the input-capture input equals to one PCLK cycle detected by internal sampling, input capture is generated continuously on the rising and falling edges. Therefore, an input-capture pulse interrupt signal is output continuously to form a flat signal level.
When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 24.54 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

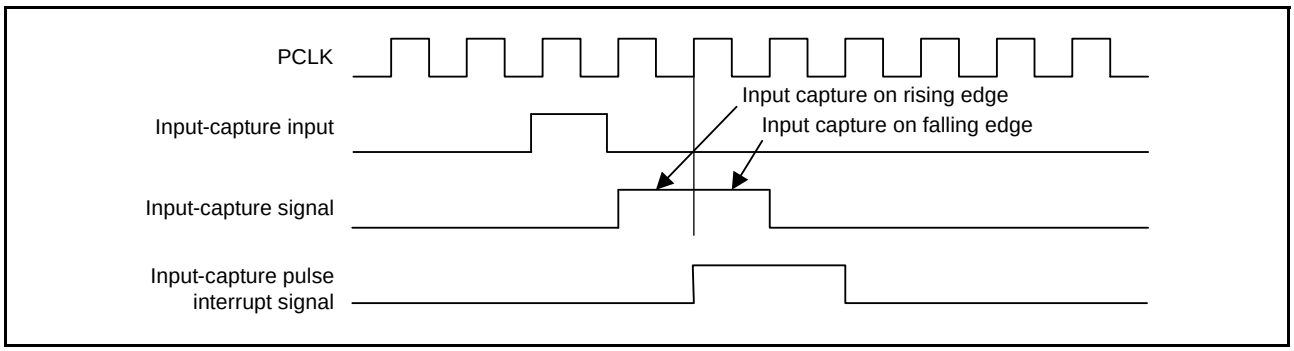


Figure 24.54 Continuous Output of Input-Capture Pulse Interrupt Signal

24.9.16 Continuous Output of Underflow Pulse Interrupt Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal and an underflow interrupt signal are output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 24.55 shows an operation timing when the underflow pulse interrupt signal is output continuously.

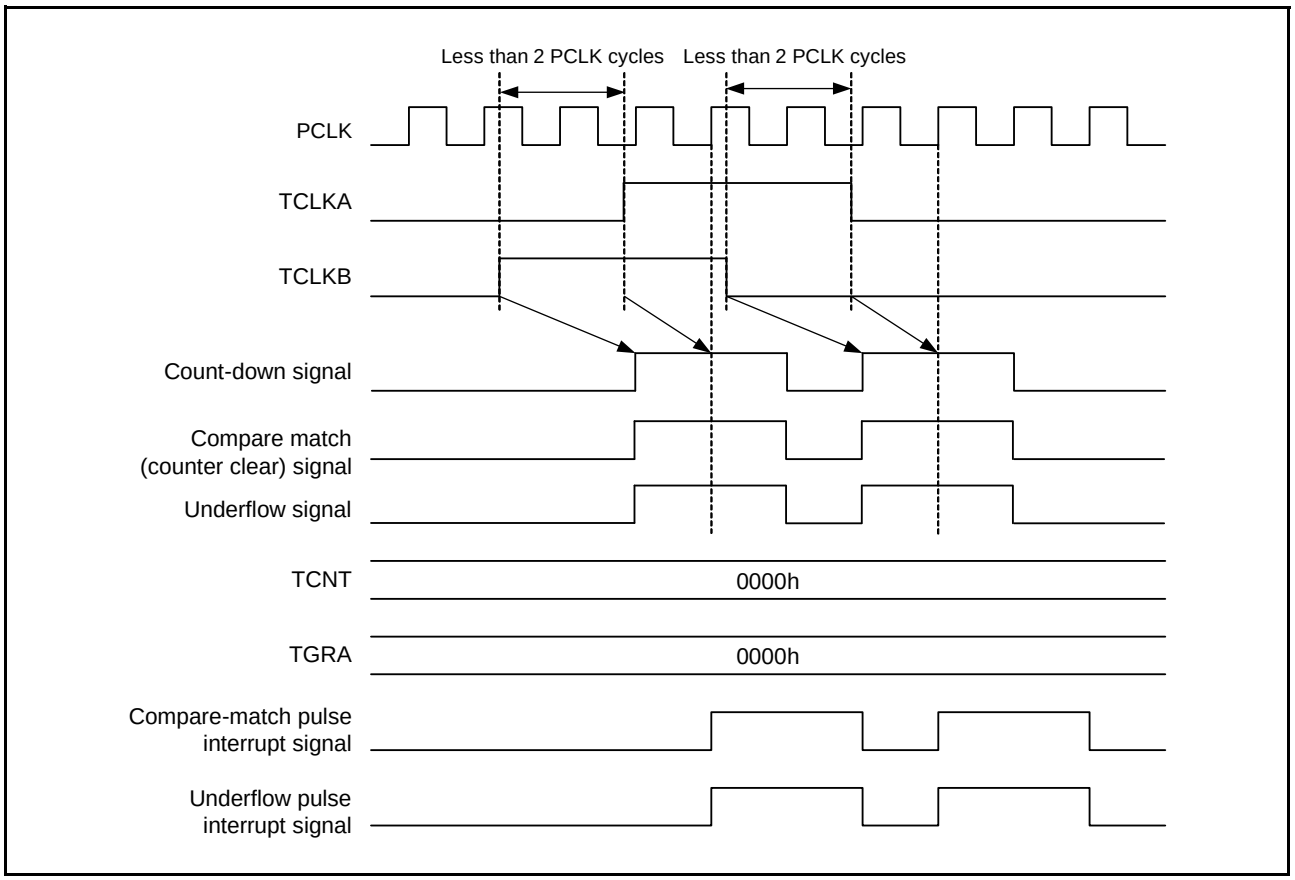


Figure 24.55 Continuous Output of Underflow Pulse Interrupt Signal

25. Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) generates pulse outputs by using multi-function timer pulse unit 2 (MTU) as a time base.

This MCU has one PPG unit, which controls up to 16 pulse output pins. The pulse output from the PPG is divided into 4-bit groups that can operate all simultaneously and independently.

25.1 Overview

Table 25.1 lists the specifications of the PPG and Table 25.2 lists PPG functions.

Figure 25.1 shows block diagrams of the PPG.

Table 25.1 Specifications of PPG

Item	Specifications
Number of output bits	Up to 16 bits
Pulse output	• One unit, capable of output through four pin groups • Output trigger signals are selectable. • Non-overlapping operation is possible. • Inverted output is selectable.
Output data transfer	Can operate together with the DTC and DMAC (when MTU interrupt is in use)
Power consumption reducing function	Module-stop state can be set.

Table 25.2 List of PPG Functions

Item			PPG0
PPG output trigger	MTU channels 0 to 3 (MTU0 to MTU3)	Compare match	○
		Input capture	○
Non-overlapping operation			○
Output data transfer	DTC		○
	DMAC		○
Selecting inverted output			○
Setting the module-stop state*1			The MSTPA11 bit in MSTPCRA

○: Possible

—: Not possible

Note 1. For details, see section 11, Low Power Consumption.

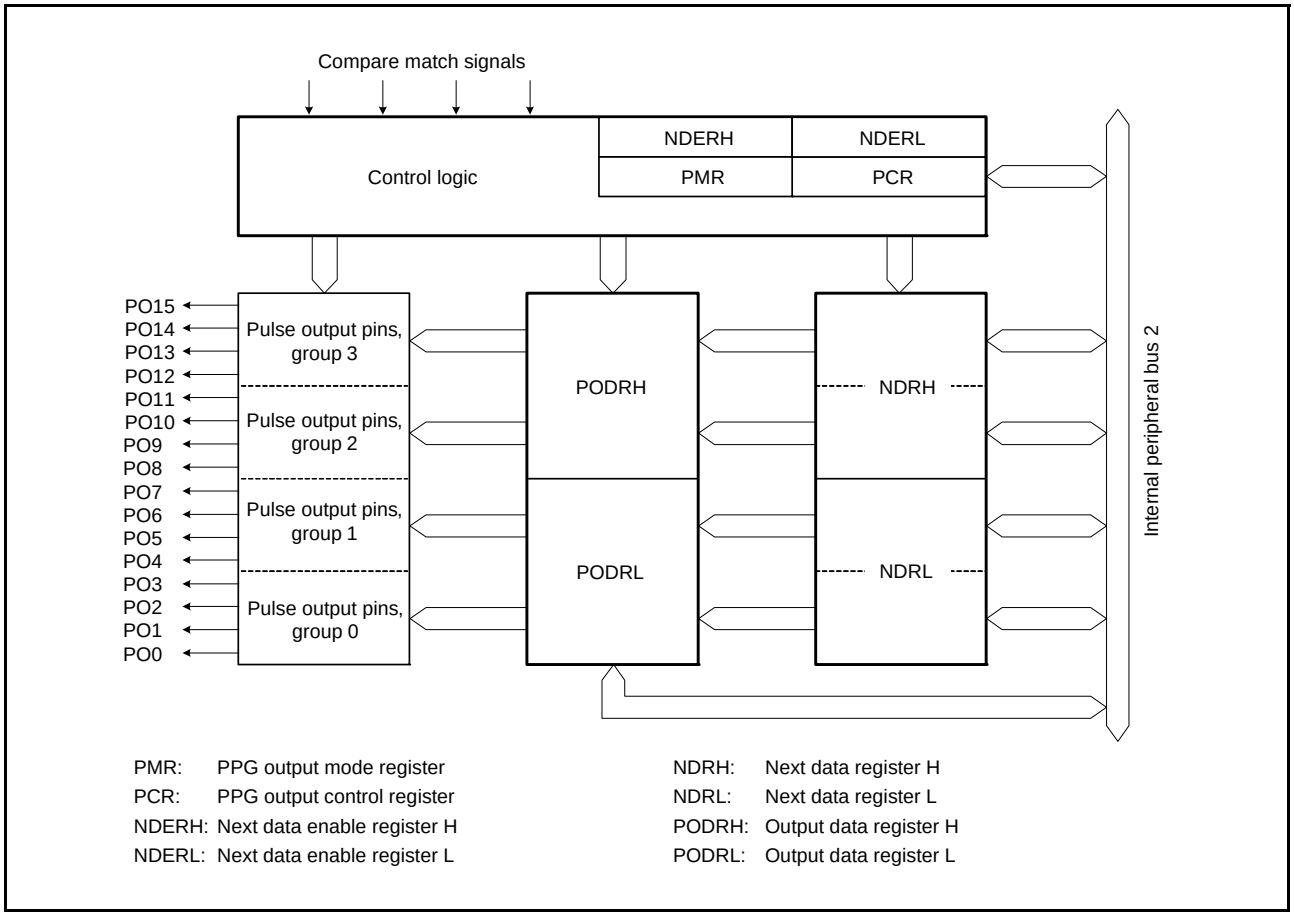


Figure 25.1 Block Diagram of PPG0

Table 25.3 lists the pin configuration of the PPG.

Table 25.3 Pin Configuration of PPG

Unit	Pin Name	I/O	Function
PPG0	PO0	Output	Group 0 pulse output
	PO1	Output	
	PO2	Output	
	PO3	Output	
	PO4	Output	Group 1 pulse output
	PO5	Output	
	PO6	Output	
	PO7	Output	
	PO8	Output	Group 2 pulse output
	PO9	Output	
	PO10	Output	
	PO11	Output	
	PO12	Output	Group 3 pulse output
	PO13	Output	
	PO14	Output	
	PO15	Output	

25.2 Register Descriptions

25.2.1 Next Data Enable Registers H (NDERH) Next Data Enable Registers L (NDERL)

Address(es): 0008 81E8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERH	NDER 15	NDER 14	NDER 13	NDER 12	NDER 11	NDER 10	NDER9	NDER8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81E9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER8	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER9	Next Data Transfer Enable		R/W
b2	NDER 10	Next Data Transfer Enable		R/W
b3	NDER 11	Next Data Transfer Enable		R/W
b4	NDER 12	Next Data Transfer Enable		R/W
b5	NDER 13	Next Data Transfer Enable		R/W
b6	NDER 14	Next Data Transfer Enable		R/W
b7	NDER 15	Next Data Transfer Enable		R/W

PPG0.NDERH selects the pins (PO15 to PO8) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 15 to 8)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRH to the bit in PPG0.PODRH.

- PPG0.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER0	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER1	Next Data Transfer Enable		R/W
b2	NDER2	Next Data Transfer Enable		R/W
b3	NDER3	Next Data Transfer Enable		R/W
b4	NDER4	Next Data Transfer Enable		R/W
b5	NDER5	Next Data Transfer Enable		R/W
b6	NDER6	Next Data Transfer Enable		R/W
b7	NDER7	Next Data Transfer Enable		R/W

PPG0.NDERL selects the pins (PO7 to PO0) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 7 to 0)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRL to the bit in PPG0.PODRL.

25.2.2 Output Data Registers H (PODRH) Output Data Registers L (PODRL)

Address(es): 0008 81EAh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81EBh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD8	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 15 to 8)	R/W
b1	POD9	Output Data Register		R/W
b2	POD10	Output Data Register		R/W
b3	POD11	Output Data Register		R/W
b4	POD12	Output Data Register		R/W
b5	POD13	Output Data Register		R/W
b6	POD14	Output Data Register		R/W
b7	POD15	Output Data Register		R/W

PPG0.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERH, the output trigger transfers the values in PPG0.NDRH to this register.

PODi Bit (Output Data Register) (i = 15 to 8)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERH register are transferred from the PPG0.NDRH register to this register. Writing from the CPU is impossible while any of the NDERi (i = 15 to 8) bits in PPG0.NDERH is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERH register is 00h.

- PPG0.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD0	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 7 to 0)	R/W
b1	POD1	Output Data Register		R/W
b2	POD2	Output Data Register		R/W
b3	POD3	Output Data Register		R/W
b4	POD4	Output Data Register		R/W
b5	POD5	Output Data Register		R/W
b6	POD6	Output Data Register		R/W
b7	POD7	Output Data Register		R/W

PPG0.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERL, the output trigger transfers the values in PPG0.NDRL to this register.

PODi Bit (Output Data Register) (i = 7 to 0)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERL register are transferred from the PPG0.NDRL register to this register. Writing from the CPU is impossible while any of the NDERi (i = 7 to 0) bits in PPG0.NDERL is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERL register is 00h.

25.2.3 Next Data Registers H (NDRH) Next Data Registers L (NDRL)

Address(es): 0008 81ECh, 0008 81EEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81EDh, 0008 81EFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDRH

PPG0.NDRH stores the next data for pulse output. The PPG0.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 2 and 3 have the same output trigger

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 2 and 3: 0008 81ECh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b4	NDR12	Next Data Register		R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

Note: The address (0008 81EEh) to which PPG0.NDRH address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 2 and 3 have different output triggers

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 3: 0008 81ECh

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	NDR12	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

- Pulse output group 2: 0008 81EEh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

- PPG0.NDRL

PPG0.NDRL stores the next data for pulse output. The PPG0.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 0 and 1 have the same output trigger

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 0 and 1: 0008 81EDh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b4	NDR4	Next Data Register		R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

Note: The address (0008 81EFh) to which PPG0.NDRL address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 0 and 1 have different output triggers

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 1: 0008 81EDh

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	NDR4	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

- Pulse output group 0: 0008 81EFh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

25.2.4 PPG Output Control Register (PCR)

Address(es): PPG0.PCR 0008 81E6h

	b7	b6	b5	b4	b3	b2	b1	b0
	G3CMS[1:0]		G2CMS[1:0]		G1CMS[1:0]		G0CMS[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

- PPG0.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 0 Compare Match Select	b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b3, b2	G1CMS[1:0]	Group 1 Compare Match Select	b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b5, b4	G2CMS[1:0]	Group 2 Compare Match Select	b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b7, b6	G3CMS[1:0]	Group 3 Compare Match Select	b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W

PPG0.PCR selects pulse output trigger signals on a group-by-group basis. For details on output trigger selection, see section 25.2.5, PPG Output Mode Register (PMR).

25.2.5 PPG Output Mode Register (PMR)

Address(es): PPG0.PMR 0008 81E7h

	b7	b6	b5	b4	b3	b2	b1	b0
	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Value after reset:	1	1	1	1	0	0	0	0

- PPG0.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 0 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b1	G1NOV	Group 1 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b2	G2NOV	Group 2 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b3	G3NOV	Group 3 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b4	G0INV	Group 0 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 1 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 2 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 3 Output Polarity Change	0: Inverted output 1: Direct output	R/W

PPG0.PMR selects the pulse output mode of the PPG on a group-by-group basis.

While inverted output is selected, a low-level pulse is output when the values in PPG0.PODRH and PPG0.PODRL are 1, and a high-level pulse is output when the values in PPG0.PODRH and PPG0.PODRL are 0.

In addition, when non-overlapping operation is selected, the PPG updates its output values on compare match A or B in an MTU or TPU channel that functions as an output trigger.

For details, see section 25.3.4, Non-Overlapping Pulse Output.

25.3 Operation

Figure 25.2 shows a schematic diagram of the PPG.
PPG pulse output is enabled when the corresponding bits in PPG0.NDERH and PPG0.NDERL are set to 1 (data transfer is enabled).
An initial output value is determined by the initial settings in the corresponding PPG0.PODRH and PPG0.PODRL.
When the compare match event selected in PPG0.PCR occurs, the output values are updated by transfer of the values in the corresponding PPG0.NDRH and PPG0.NDRL to PPG0.PODRH and PPG0.PODRL, respectively.
Consecutive output of up to 16 bits of data is possible by writing new output data to PPG0.NDRH and PPG0.NDRL before the next compare match.

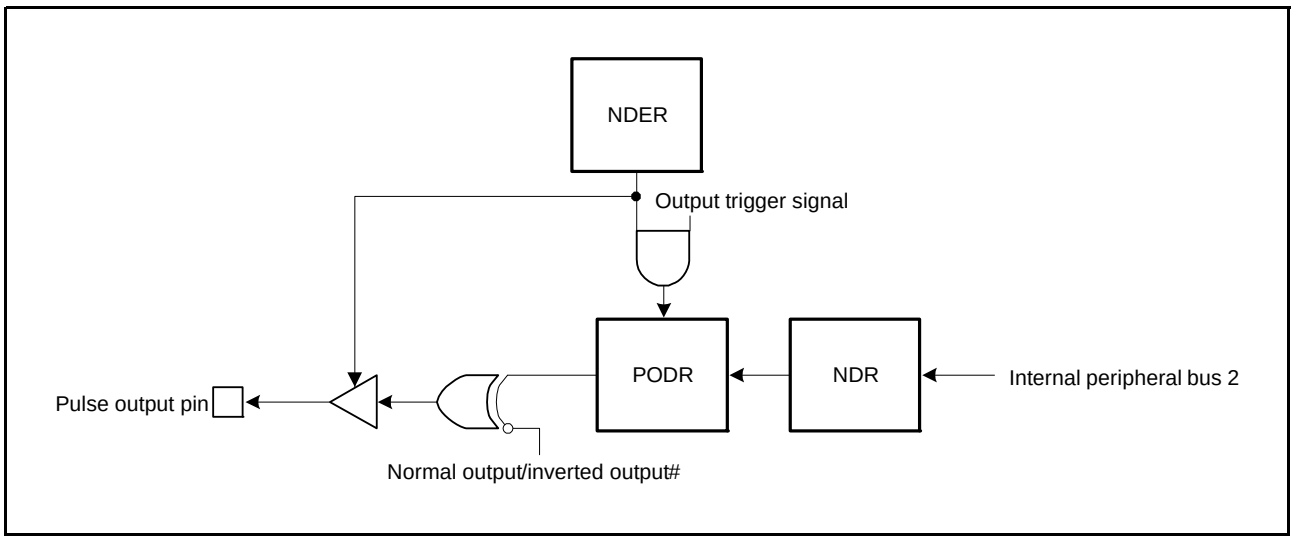


Figure 25.2 Schematic Diagram of PPG

25.3.1 Output Timing

When the selected compare match event occurs while pulse output is enabled, the values in PPG0.NDRH and PPG0.NDRL are transferred to PPG0.PODRH and PPG0.PODRL, respectively, and then output on the corresponding pins.

Figure 25.3 shows the timing of the above operation. In this case, the timing when compare match A triggers normal output from groups 2 and 3 is shown.

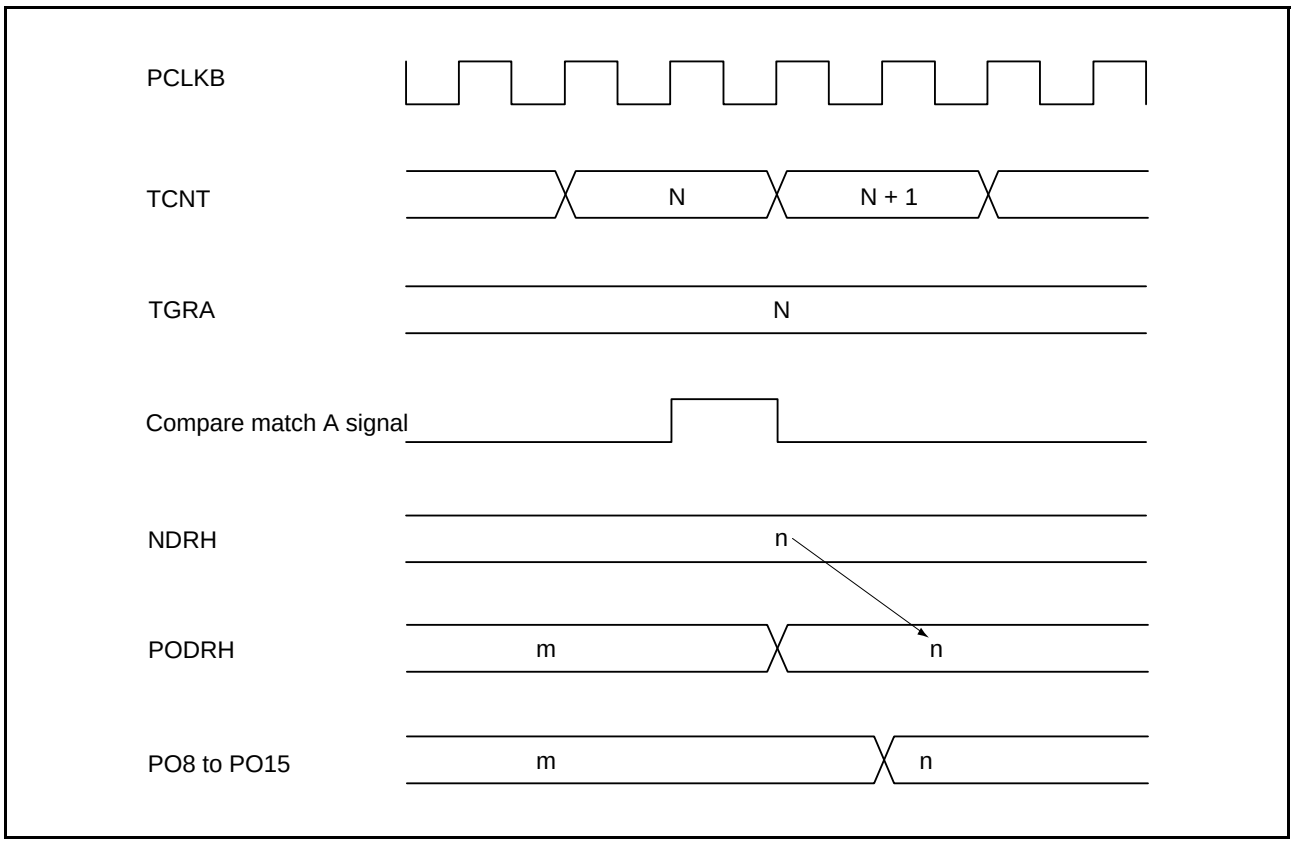


Figure 25.3 Timing of Transfer and Output of the Values in NDR (Example)

25.3.2 Sample Setup Procedure for Normal Pulse Output

Figure 25.4 shows sample procedures for setting normal pulse output.

(1) PPG0 Setting

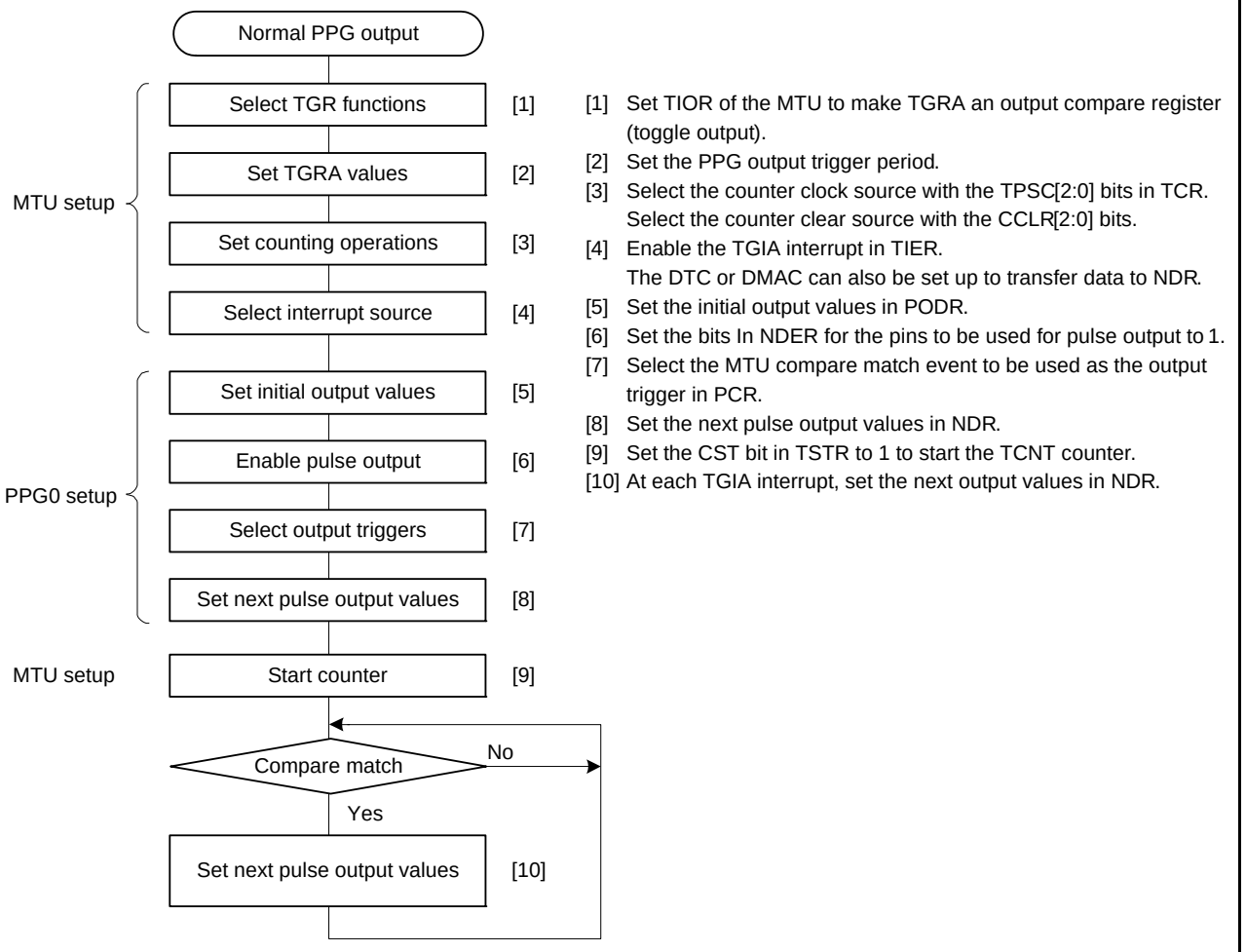


Figure 25.4 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting)

25.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 25.5 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.

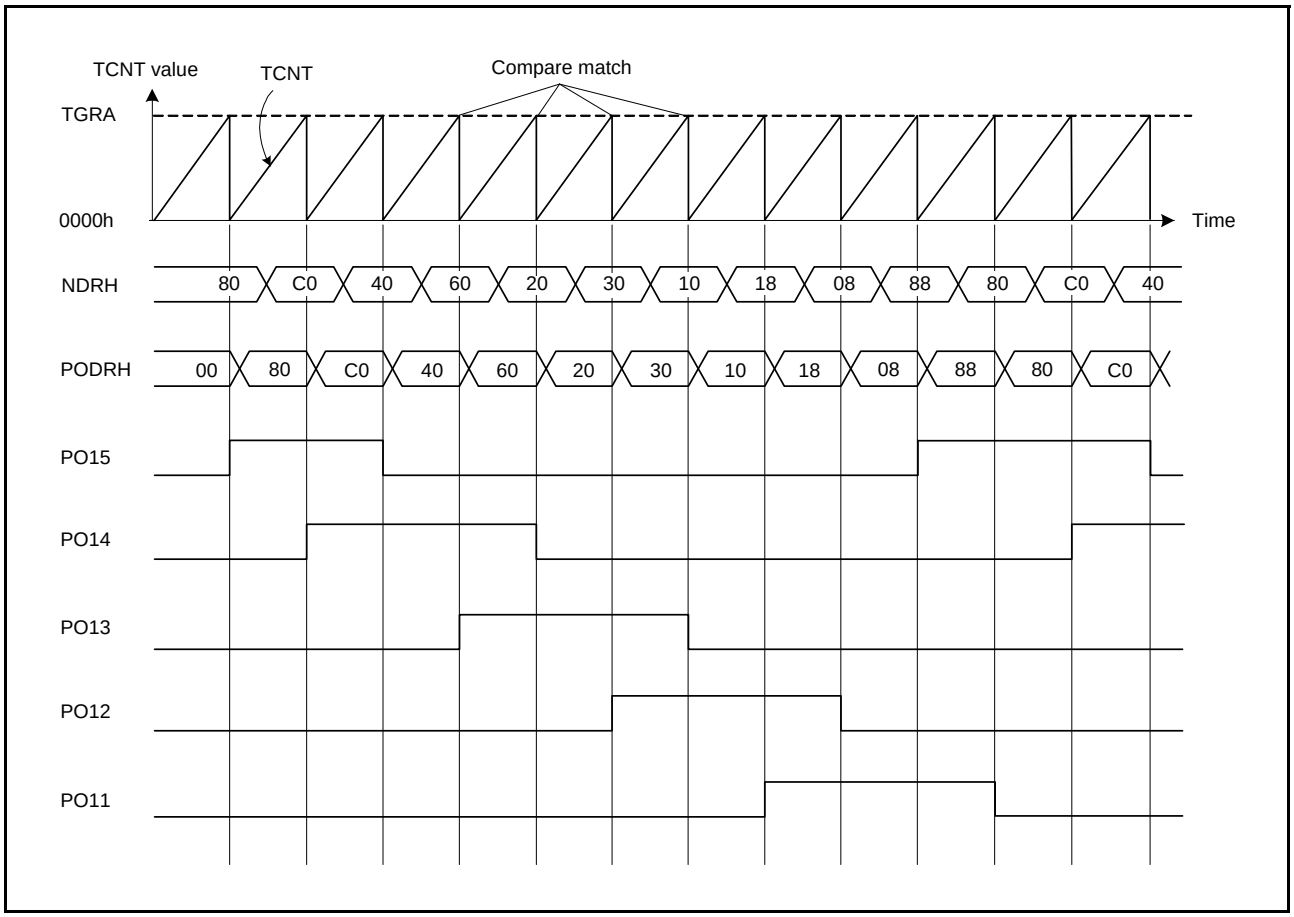


Figure 25.5 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

1. Set an output compare register of the MTUn.TGRA (n = 0 to 3) of MTU so that the corresponding compare match signal is the output trigger. Set a cycle in TGRA so that the counter will be cleared by compare match A. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIAn) interrupt.
2. Write F8h to PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers. Write output data 80h to PPG0.NDRH.
3. The timer counter in the MTU starts. When compare match A occurs, the values in PPG0.NDRH are transferred to PPG0.PODRH and output. The TGIAn interrupt handling routine writes the next output data C0h to PPG0.NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing 40h, 60h, 20h, 30h, 10h, 18h, 08h, 88h... at successive TGIAn interrupts.
   If the DTC or DMAC is set for activation by the TGIAn interrupt, pulse output can be obtained without imposing a load on the CPU.

25.3.4 Non-Overlapping Pulse Output

During non-overlapping operation, data transfer from PPG0.NDRH and PPG0.NDRL to PPG0.PODRH and PPG0.PODRL is performed as follows.

- On compare match A, the values in PPG0.NDRH and PPG0.NDRL are transferred to PPG0.PODRH and PPG0.PODRL.
- On compare match B, data transfer proceeds for bits in PPG0.NDRH and PPG0.NDRL that have the value 0. It does not proceed for bits having the value 1.

Figure 25.6 shows the non-overlapping pulse output operation.

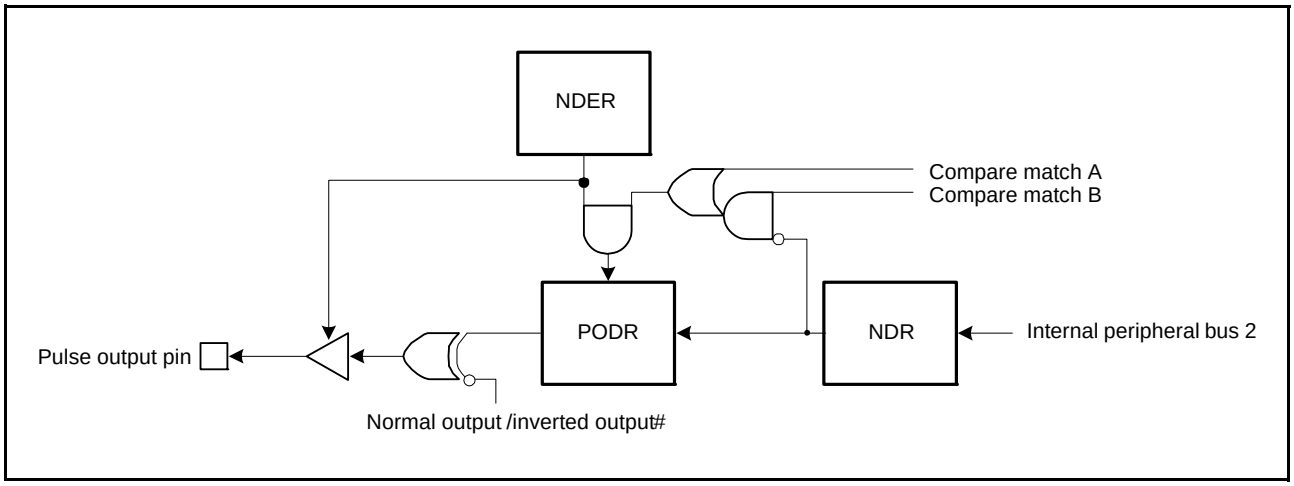


Figure 25.6 Non-Overlapping Pulse Output

Therefore, compare match B before compare match A allows 0-valued data to be transferred in advance of 1-valued data. Do not change the values in PPG0.NDRH and PPG0.NDRL during the interval from compare match B to compare match A (the non-overlap margin).

To transfer 0-valued data in advance of 1-valued data, write the next data to PPG0.NDRH and PPG0.NDRL from within the TGIA interrupt handling routine or by using a TGIA interrupt to activate transfer by the DTC or DMAC. In any case, the next data must be written before the next compare match B occurs.

Figure 25.7 shows the timing of the above procedure.

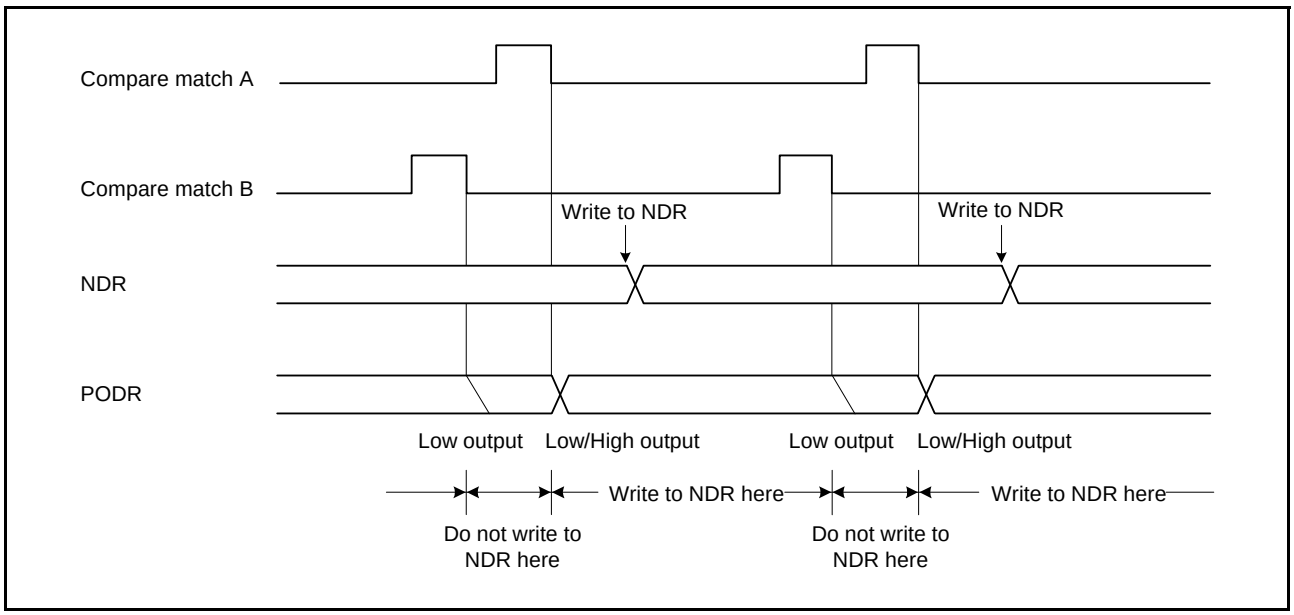
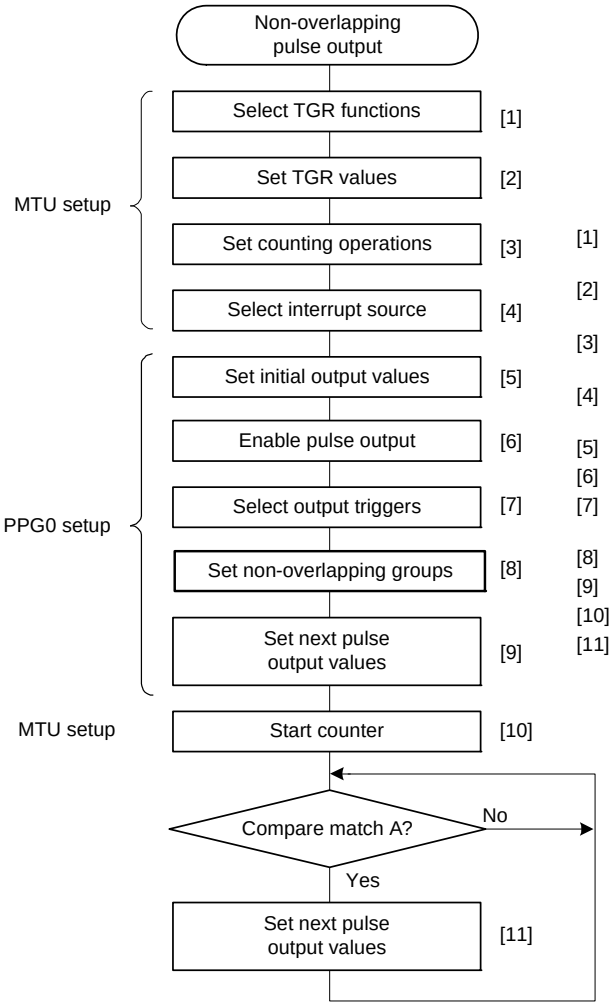


Figure 25.7 Non-Overlapping Operation and Write Timing to PPG0.NDRH and PPG0.NDRL

25.3.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 25.8 shows sample procedures for setting up non-overlapping pulse outputs.

(1) PPG0 Setting



[1] Set TIOR of the MTU to make TGRA and TGRB output compare registers (toggle output).
[2] Set the pulse output trigger period in TGRB and the non-overlap period in TGRA.
[3] Select the counter clock source with the TPSC[2:0] bits in TCR. Select the counter clear source with the CCLR[2:0] bits.
[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.
[5] Set the initial output values in PODR.
[6] Set the bits in NDER for the pins to be used for pulse output to 1.
[7] Select the MTU compare match event to be used as the pulse output trigger in PCR.
[8] In PMR, select the groups that will operate in non-overlap mode.
[9] Set the next pulse output values in NDR.
[10] Set the CST bit in TSTR to 1 to start the TCNT counter.
[11] At each TGIA interrupt, set the next output values in NDR.

Figure 25.8 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting)

25.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 25.9 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.

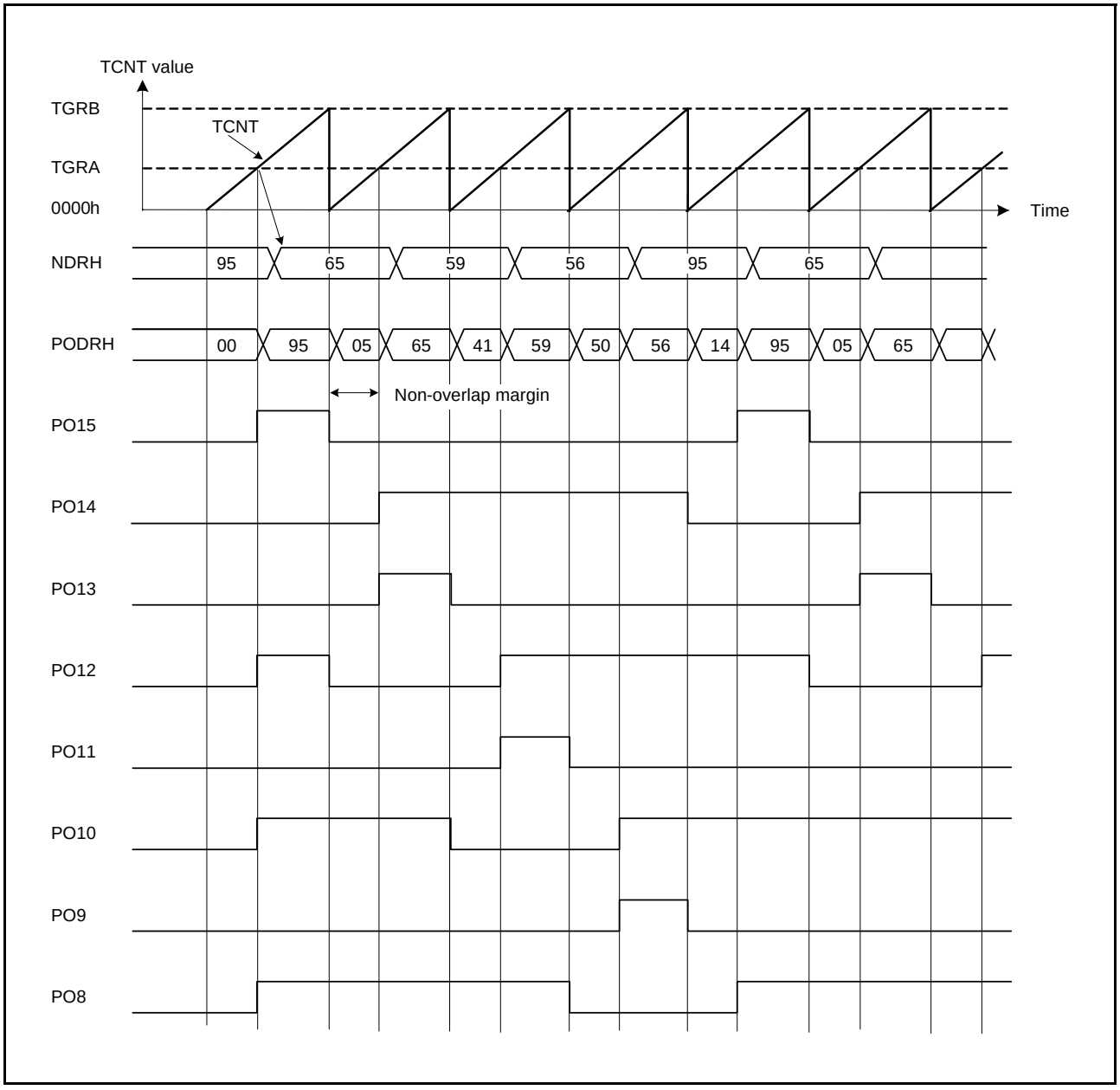


Figure 25.9 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

1. Set output compare registers of the MTUn.TGRA and MTUn.TGRB (n = 0 to 3) of MTU so that the corresponding compare match signals are the output triggers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIAn) interrupt.
2. Write FFh in PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers.
   Set the G3NOV and G2NOV bits in PPG0.PMR to 1 to select non-overlapping outputs. Write output data 95h in PPG0.NDRH.
3. The timer counter in the MTU starts. When a compare match with TGRB occurs, outputs change from high to low. When a compare match with TGRA occurs, outputs change from low to high (the change from low to high is delayed by the value set in TGRA).
   The TGIAn interrupt handling routine writes the next output data 65h in PPG0.NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing 59h, 56h, 95h... at successive TGIAn interrupts.
   If the DTC or DMAC is set for activation by the TGIAn interrupt, pulse output can be obtained without imposing a load on the CPU.

25.3.7 Inverted Pulse Output

When the G3INV, G2INV, G1INV, and G0INV bits in PPG0.PMR are cleared to 0, the values that are the inverse of the respective values in PPG0.PODRH and PPG0.PODRL can be output.

Figure 25.10 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 25.9.

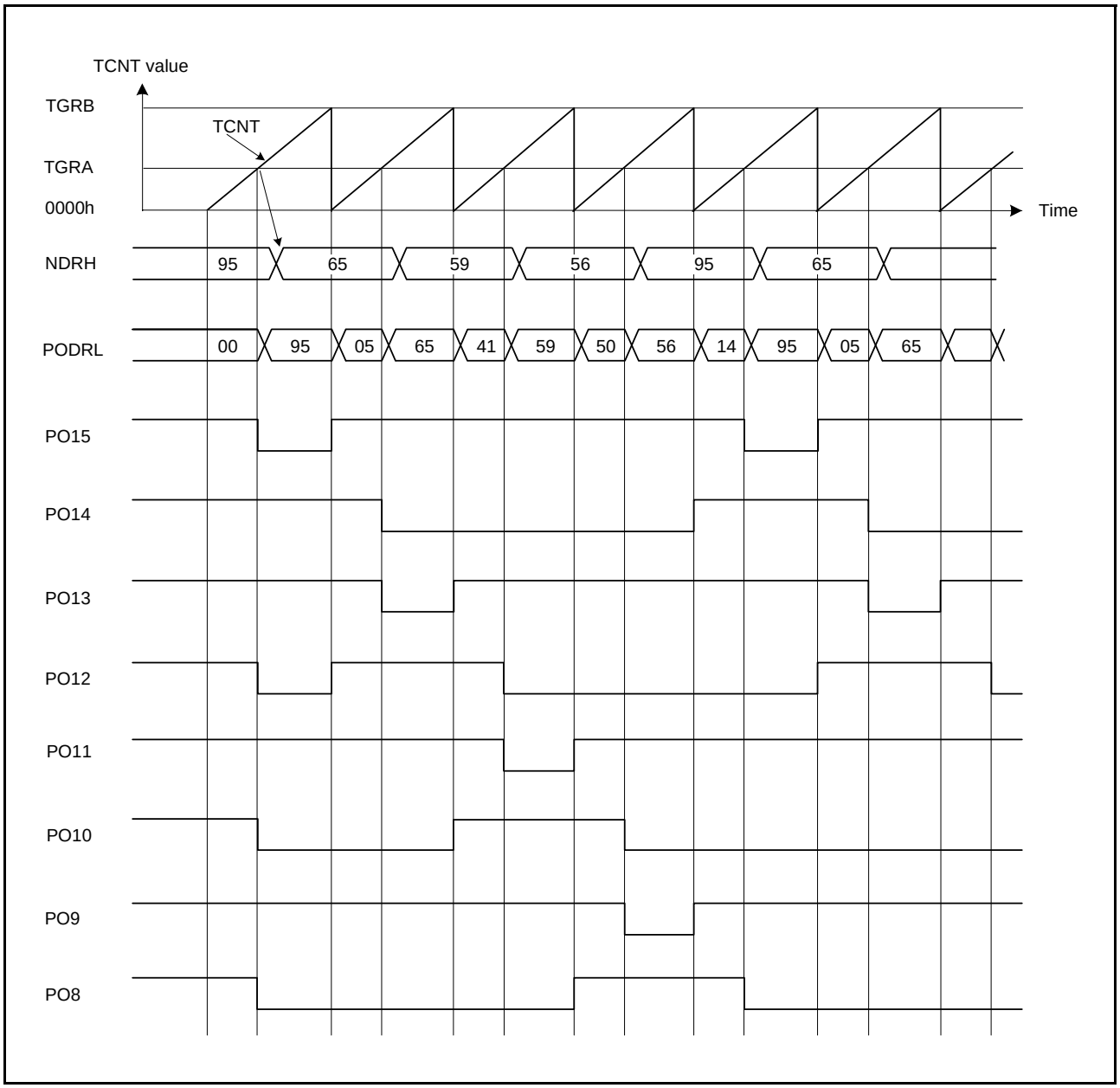


Figure 25.10 Inverted Pulse Output (Example)

25.3.8 Pulse Output Triggered by Input Capture

Pulse output from the PPG0 can be triggered by the MTU input capture as well as by compare match. When MTUn.TGRA (n = 0 to 3) functions as an input capture register in the MTU channel selected by PPG0.PCR, pulse output is triggered by the input capture signal.

Figure 25.11 shows the timing of pulse output triggered by input capture.

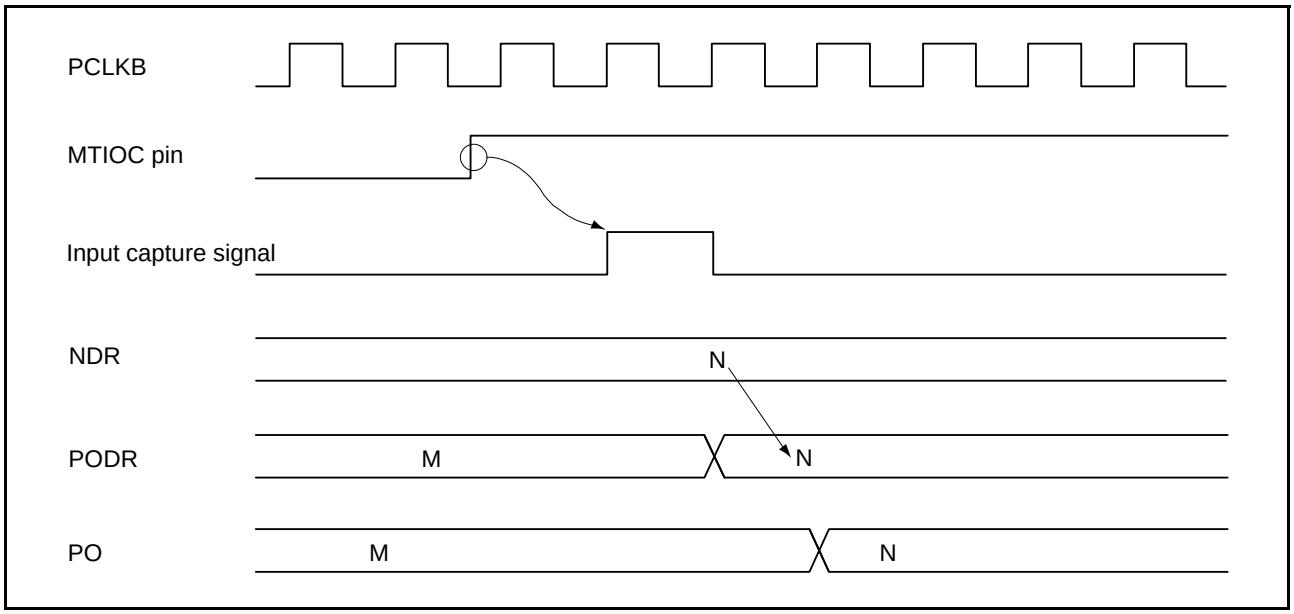


Figure 25.11 Timing of Pulse Output Triggered by Input Capture (Example)

25.4 Usage Note

25.4.1 Module-Stop Function Setting

Operation of the PPG can be disabled or enabled by the module stop control register. The initial setting is for operation of the PPG to be stopped. Register access is enabled by clearing module-stop state. For details, see section 11, Low Power Consumption.

26. 8-Bit Timer (TMR)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four 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, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a baud rate clock for the SCI and an operating clock for the remote control signal receiver (RCR).
In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

Table 26.1 lists the specifications of the TMR.
Figure 26.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 26.2 shows that of the 8-bit timer module (unit 1).

Table 26.1 Specifications of TMR

Item	Description
Count clock	• Frequency dividing clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock
Number of channels	(8 bits × 2 channels) × 2 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 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) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 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, TMR2)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0, TMR2) (2) Event counting operation (TMR0, TMR2) (3) Counting restart operation (TMR0, TMR2)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*1
Capable of generating receive clock for RCR	Generates operating clock for remote control signal receiver (RCR)*2
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, see section 30, Serial Communications Interface (SCIe, SCIf).
Note 2. For details, see section 31, Remote Control Signal Receiver (RCR).

Table 26.2 TMR Functions

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	—
SCI baud rate clock generation*1		○		—	○		—
ELC output event	Compare match A	○	—	○	○	—	○
	Compare match B	○	—	○	○	—	○
	TCNT overflow	○	—	○	○	—	○
ELC input event	Counting start	○	—	—	○	—	—
	Event counting	○	—	—	○	—	—
	Counting restart	○	—	—	○	—	—
Capable of generating receive clock for RCR*2		○	—	—	○	—	—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

○: Possible
—: Impossible
Note 1. For details, see section 30, Serial Communications Interface (SCIe, SCIf).
Note 2. For details, see section 31, Remote Control Signal Receiver (RCR).
Note 3. For details, see section 11, Low Power Consumption.

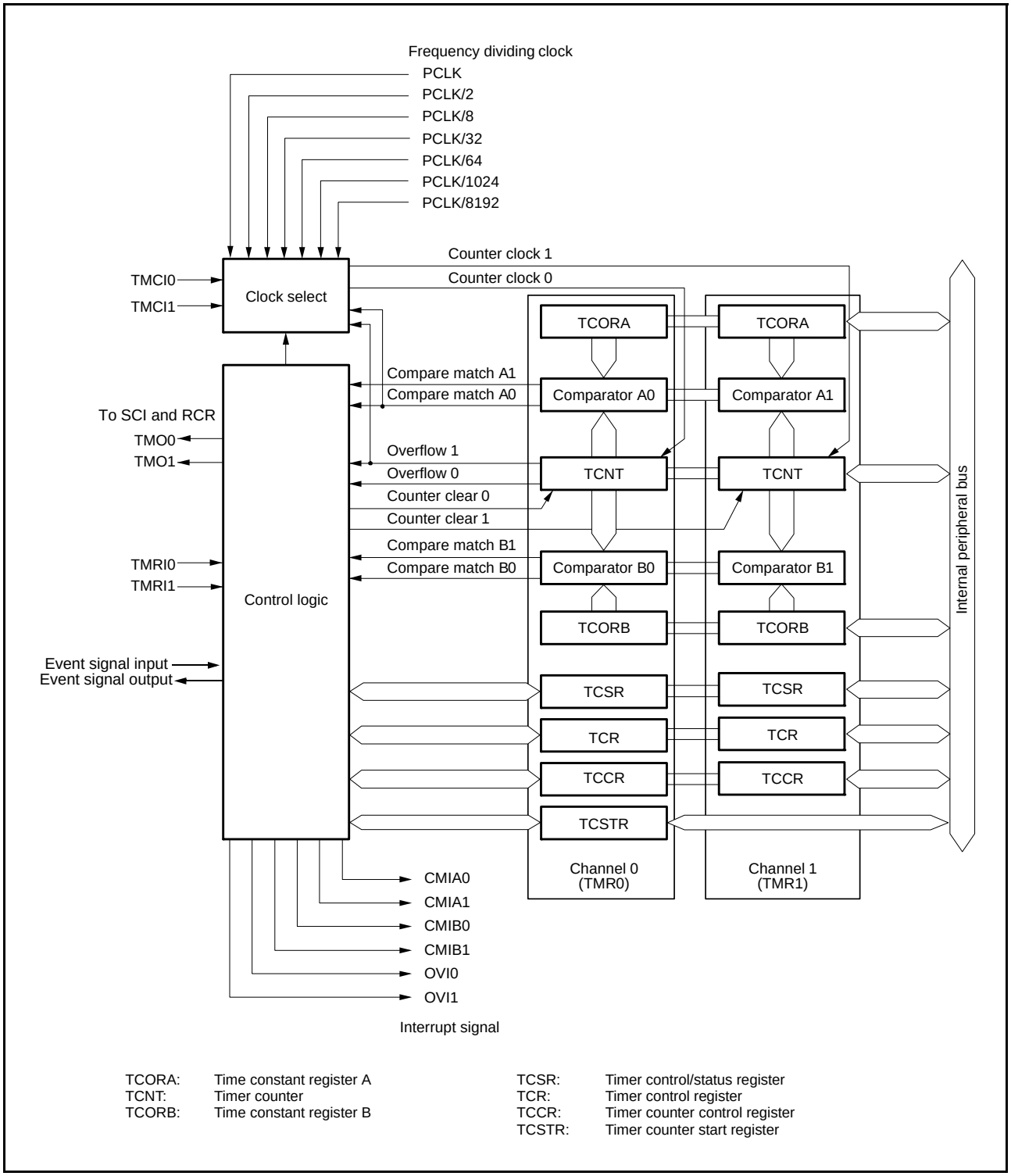


Figure 26.1 Block Diagram of TMR (Unit 0)

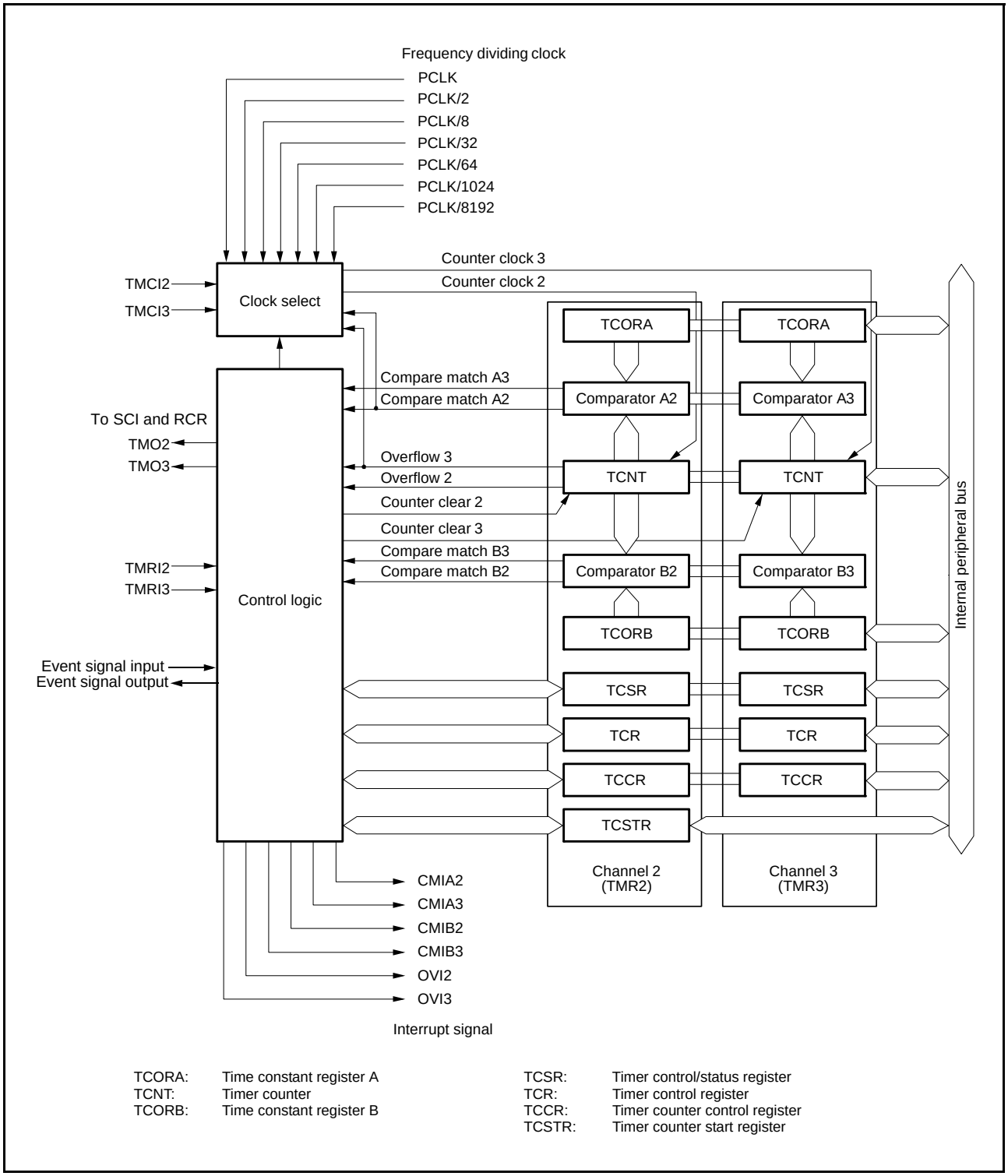


Figure 26.2 Block Diagram of TMR (Unit 1)

Table 26.3 lists the I/O pins of the TMR.

Table 26.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external clock for counter
		TMRI0	Input	Inputs external reset to counter
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external clock for counter
		TMRI1	Input	Inputs external reset to counter
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external clock for counter
		TMRI2	Input	Inputs external reset to counter
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external clock for counter
		TMRI3	Input	Inputs external reset to counter

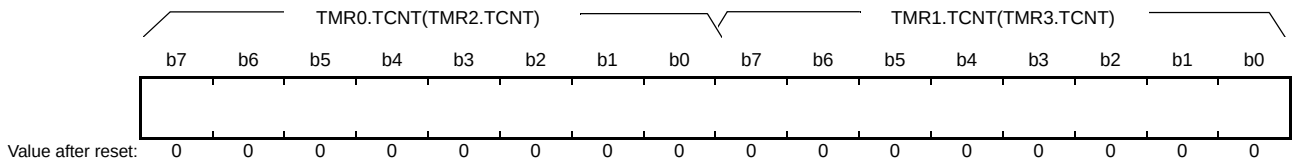
26.2 Register Descriptions

Table 26.4 Register Allocation for 16-Bit Access

Address	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR2.TCCR	TMR3.TCCR

26.2.1 Timer Counter (TCNT)

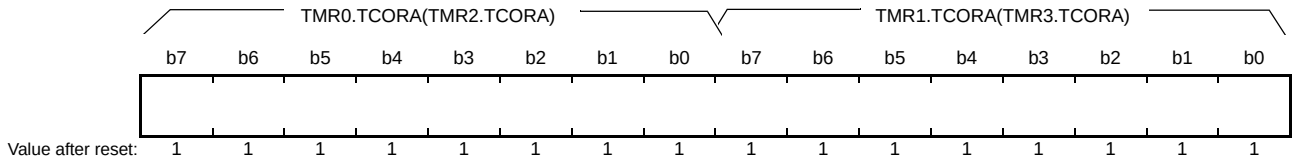
Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h



TCNT is an 8-bit readable/writable up-counter.
TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) comprise a single 16-bit counter so they can be accessed together by a word transfer instruction.
The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a counter clock.
TCNT can be cleared by an external reset input 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 (low-level pulse) is output provided the interrupt request is enabled by the TCR.OVIE bit.
For details on the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb), and Table 26.6, TMR Interrupt Sources.

26.2.2 Time Constant Register A (TCORA)

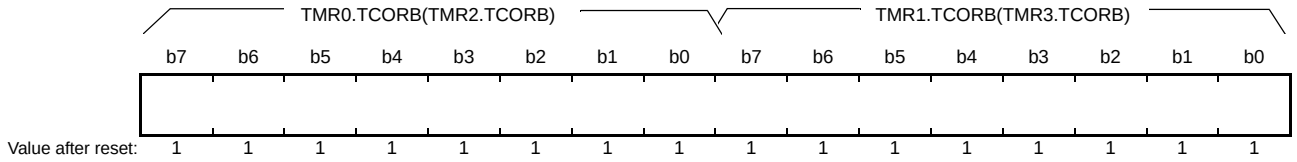
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h



TCORA is an 8-bit readable/writable register.
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.
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 (low-level pulse) 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.

26.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h



TCORB is an 8-bit readable/writable register.
TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.
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 (low-level pulse) 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.

26.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h, TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h

	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*1	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 reset input (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Timer Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match Interrupt Enable A	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match Interrupt Enable B	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 reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Timer Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match Interrupt Enable A)

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 Interrupt Enable B)

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.

26.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh

	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 26.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 26.5.	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 reset 1: Cleared when the external reset is high	R/W

Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

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 26.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external reset input) and selects the condition for detecting external reset (level or edge).

Table 26.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register					Description
	CSS[1:0]		CKS[2:0]			
	b4	b3	b2	b1	b0	
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external clock. Counts at rising edge*1.
				1	0	Uses external clock. Counts at falling edge*1.
					1	Uses external clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.
					1	Uses frequency dividing clock. Counts at PCLK/2.
				1	0	Uses frequency dividing clock. Counts at PCLK/8.
					1	Uses frequency dividing clock. Counts at PCLK/32.
			1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
					1	Uses frequency dividing clock. Counts at PCLK/1024.
				1	0	Uses frequency dividing clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external clock. Counts at rising edge*1.
				1	0	Uses external clock. Counts at falling edge*1.
					1	Uses external clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.
					1	Uses frequency dividing clock. Counts at PCLK/2.
				1	0	Uses frequency dividing clock. Counts at PCLK/8.
					1	Uses frequency dividing clock. Counts at PCLK/32.
			1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
					1	Uses frequency dividing clock. Counts at PCLK/1024.
				1	0	Uses frequency dividing clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no incrementing clock is generated. Do not use this setting.

26.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h

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

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 0. The write value should be 0.	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 the OSA[1:0] and OSB[1:0] bits are all 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 is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 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

Address(es): TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h

	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

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 the OSA[1:0] and OSB[1:0] bits are all 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 is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 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.

26.2.7 Time Count Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR2.TCSTR 0008 821Ch

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

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, see section 26.7, Link Operation by ELC, or section 19, Event Link Controller (ELC).

26.3 Operation

26.3.1 Pulse Output

Figure 26.3 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 output) and 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 cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output 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.

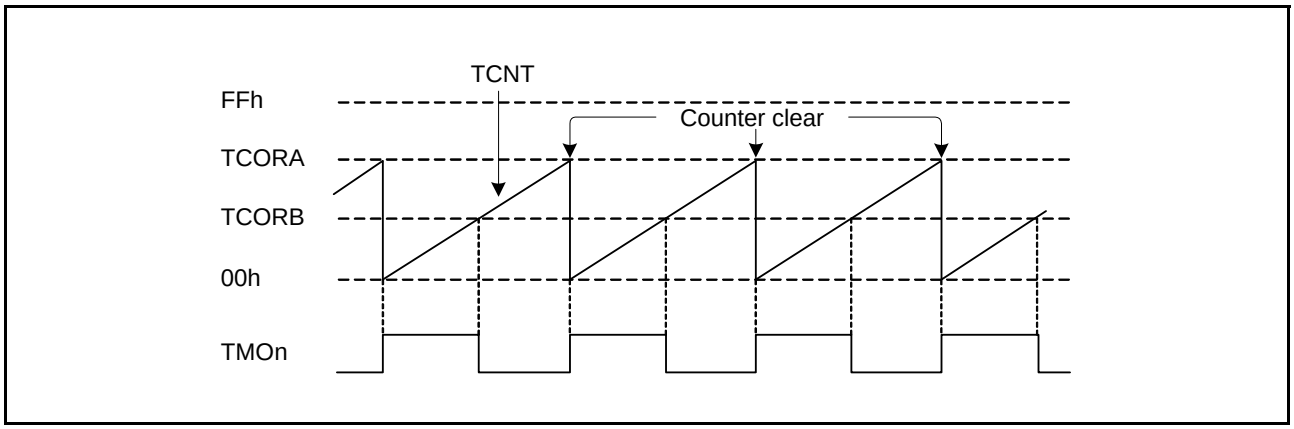


Figure 26.3 Example of Pulse Output (n = 0 to 3)

26.3.2 Reset Input

Figure 26.4 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 reset input) and set the TMRIS bit in TCCR to 1 (cleared when the external reset 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.

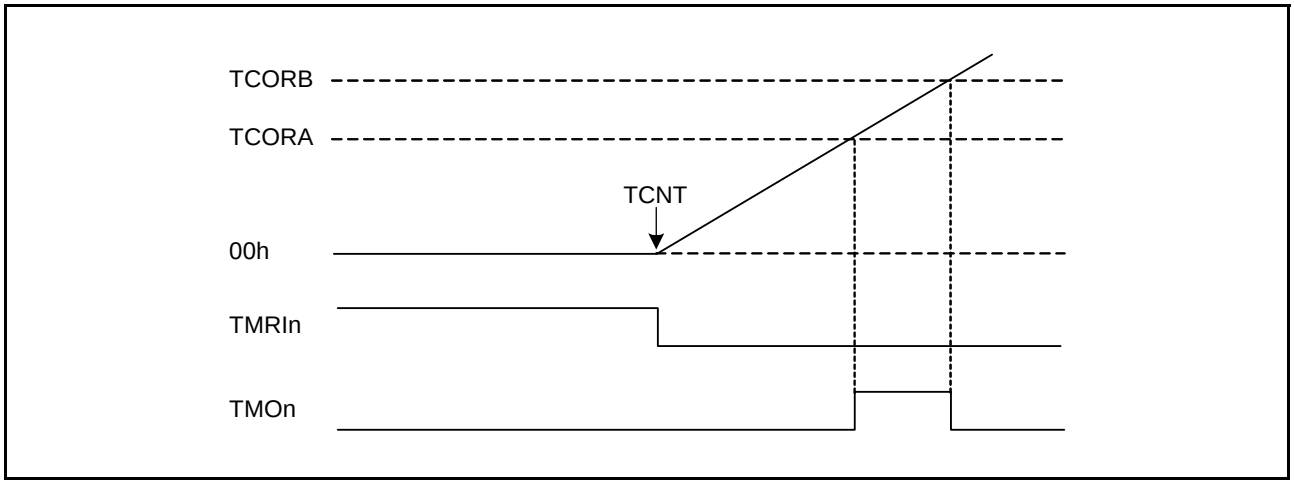


Figure 26.4 Example of Reset Input (n = 0 to 3)

26.4 Operation Timing

26.4.1 TCNT Count Timing

Figure 26.5 shows the count timing of TCNT for frequency dividing clock input. Figure 26.6 shows the count timing of TCNT for external clock input.

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.

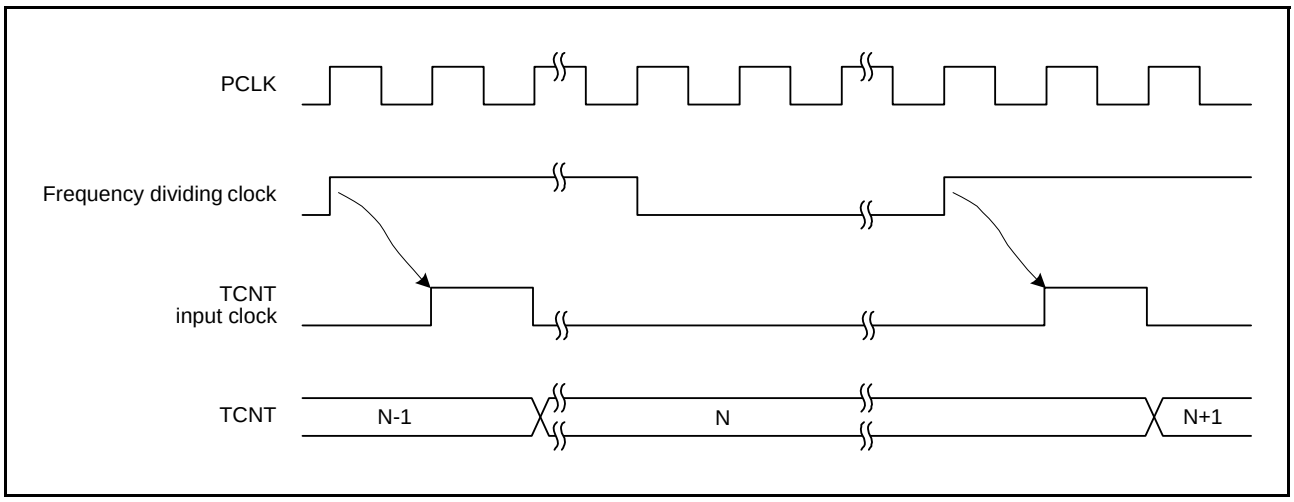


Figure 26.5 Count Timing for Frequency Dividing Clock Input

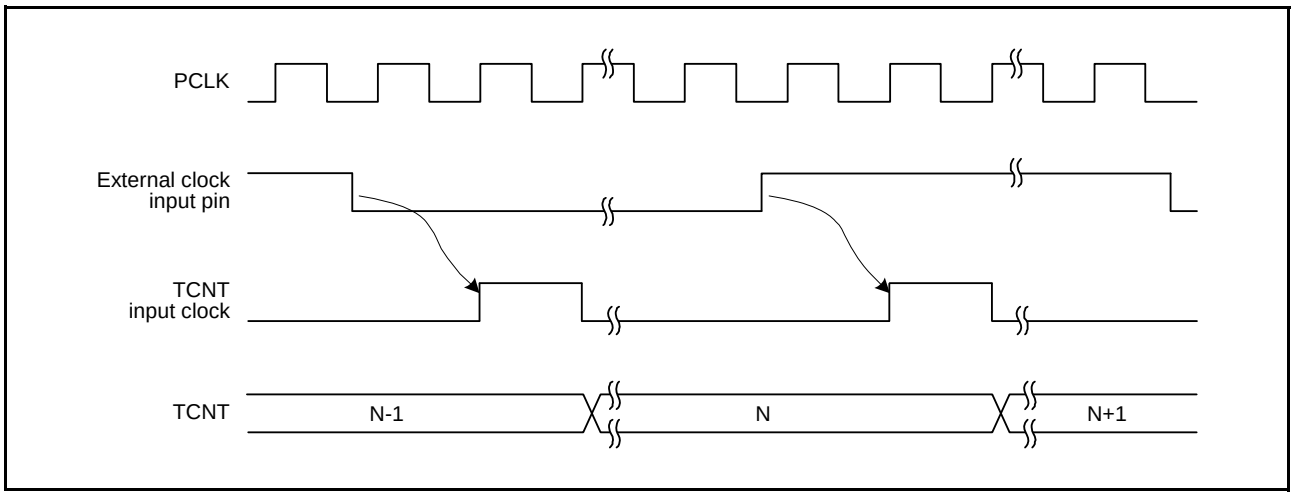


Figure 26.6 Count Timing for External Clock Input (at Both Edges)

26.4.2 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 input clock for the TCNT counter. Figure 26.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 26.6.

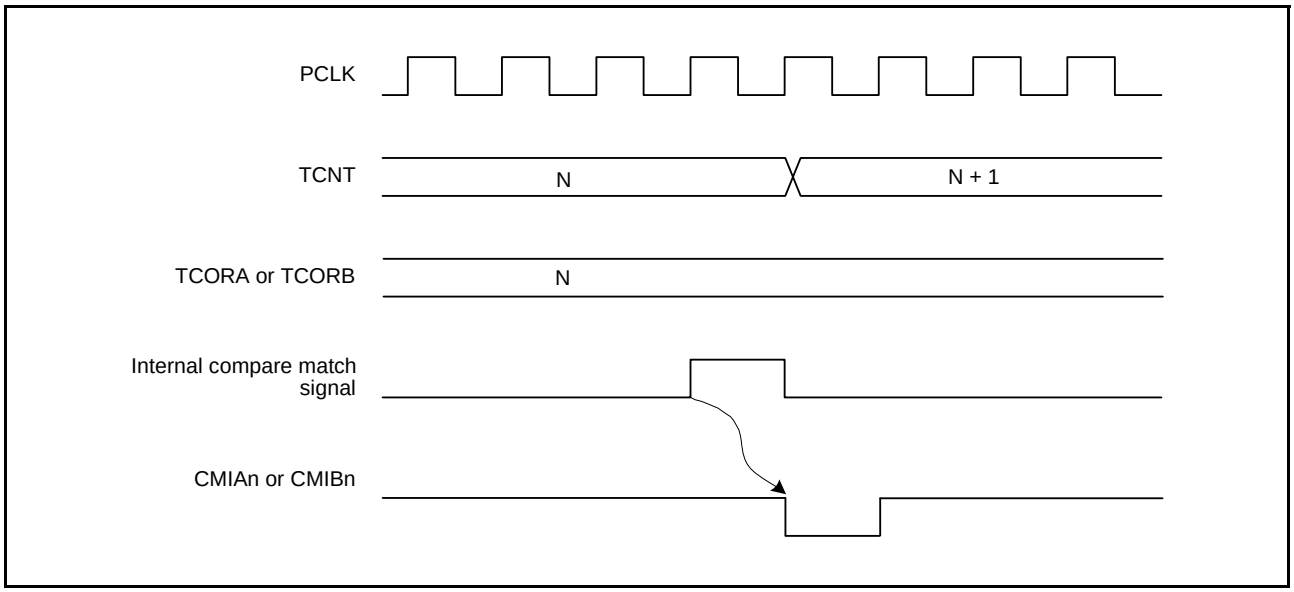


Figure 26.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

26.4.3 Timing of Timer Output 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 26.8 shows the timing when the timer output is toggled by the compare match A signal.

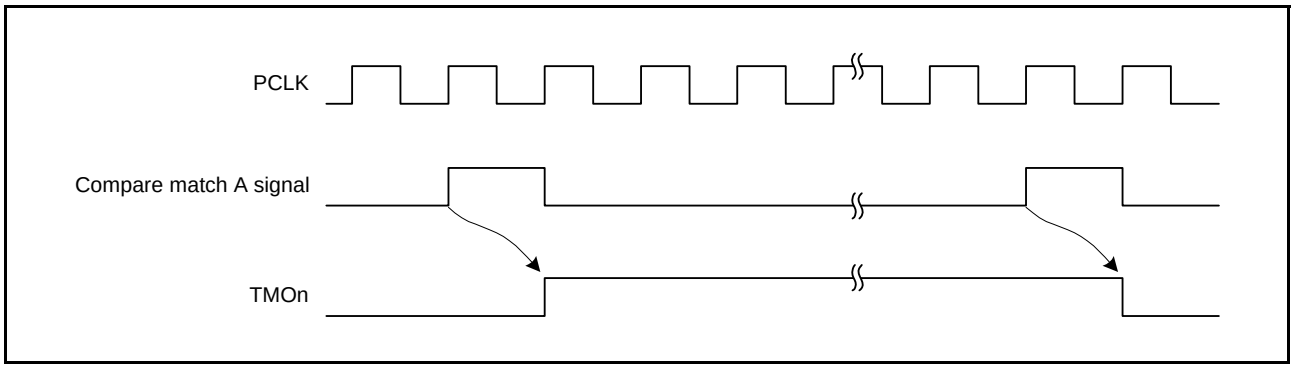


Figure 26.8 Timing of Timer Output at Compare Match A Signal (n = 0 to 3)

26.4.4 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 26.9 shows the timing of this operation.

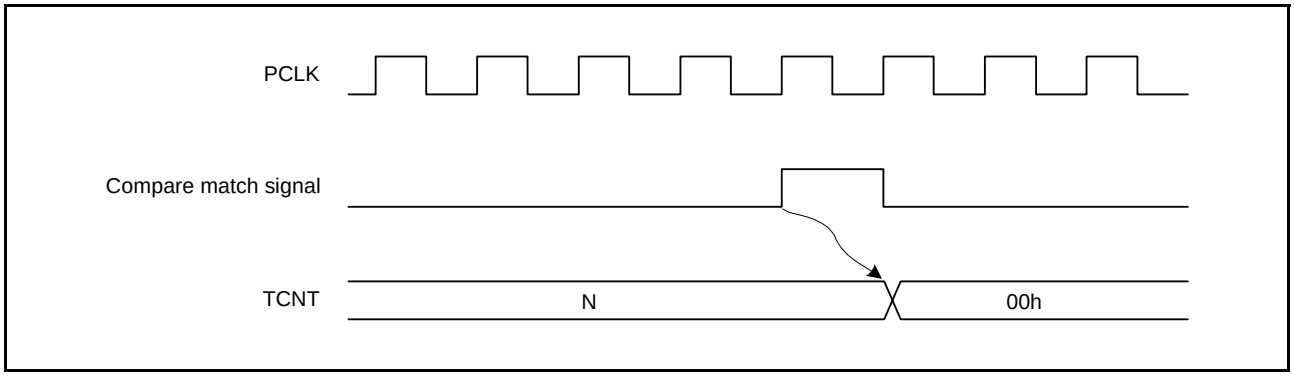


Figure 26.9 Timing of Counter Clear by Compare Match

26.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external reset input, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from an external reset input to clearing of TCNT. Figure 26.10 and Figure 26.11 show the timing of this operation.

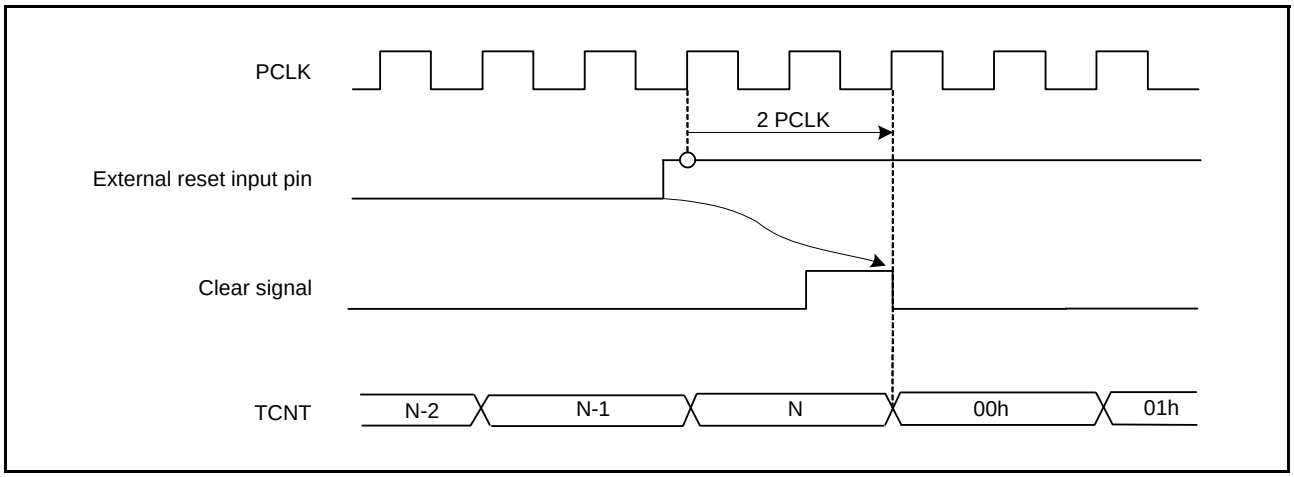


Figure 26.10 Timing of Clearance by External Reset (Rising Edge)

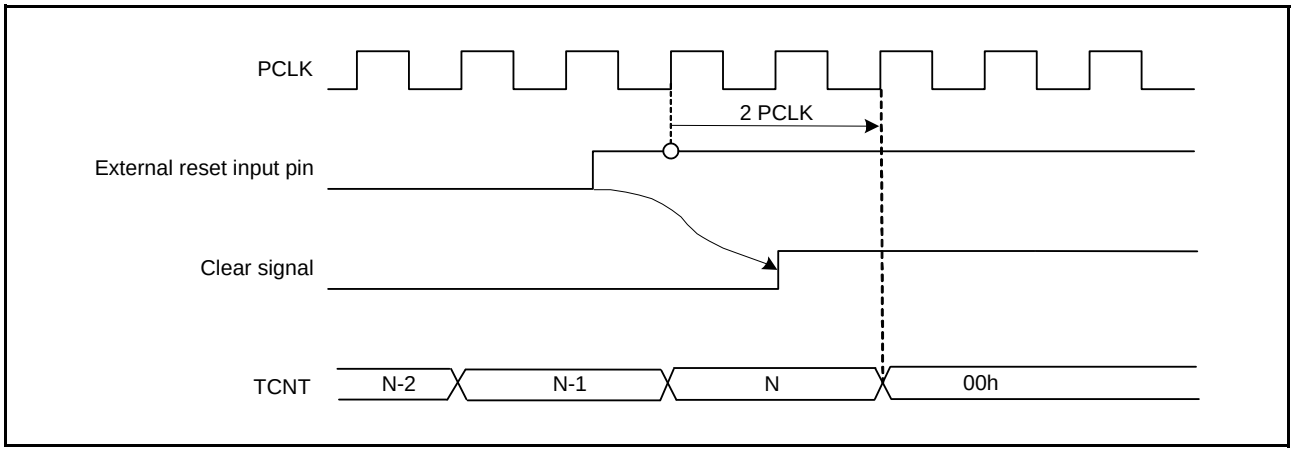


Figure 26.11 Timing of Clearance by External Reset (High Level)

26.4.6 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 26.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 26.6.

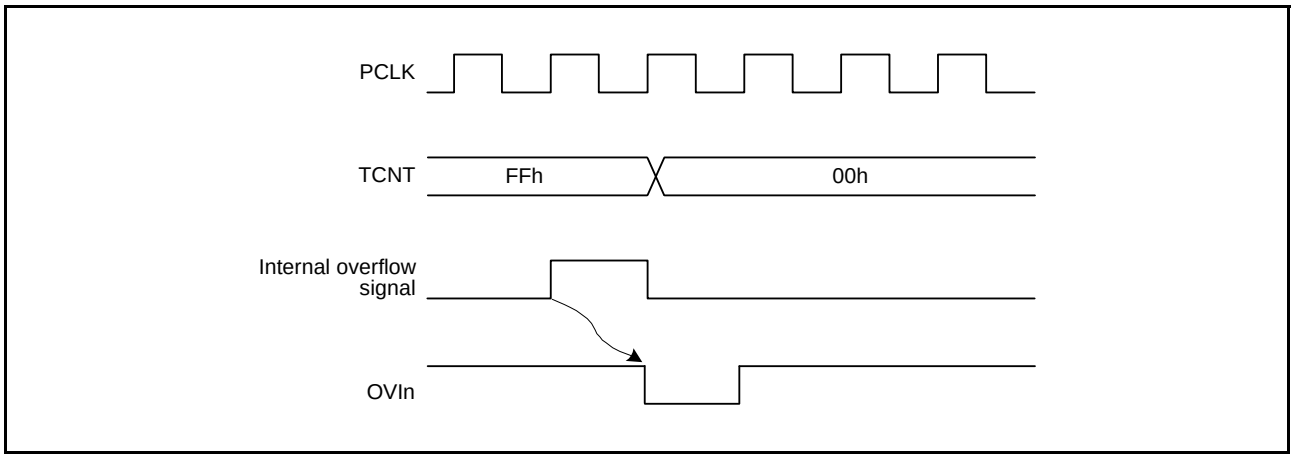


Figure 26.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

26.5 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).

Supplementary information: This section describes unit 0. The operation of unit 1 with cascaded connection is the same as unit 0.

26.5.1 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.

26.5.2 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 (n = 0, 1) pin, and counter clear are in accordance with the settings for each channel.

26.6 Interrupt Sources

26.6.1 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 26.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 26.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation	Priority
CMIA0	TMR0.TCORA compare match	Possible	High
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	Low

26.7 Link Operation by ELC

26.7.1 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 and TMR2.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMR0.TCR.OVIE or TMR2.TCR.OVIE, TMR0.TCR.CMIEA or TMR2.TCR.CMIEA, and TMR0.TCR.CMIEB or TMR2.TCR.CMIEB). For details, see section 19, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

26.7.2 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 and ELOPD.TMR2MD 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.

26.7.3 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.

26.8 Usage Notes

26.8.1 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, see section 11, Low Power Consumption.

26.8.2 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).

$$f = PCLK / (N + 1)$$

26.8.3 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 26.13.

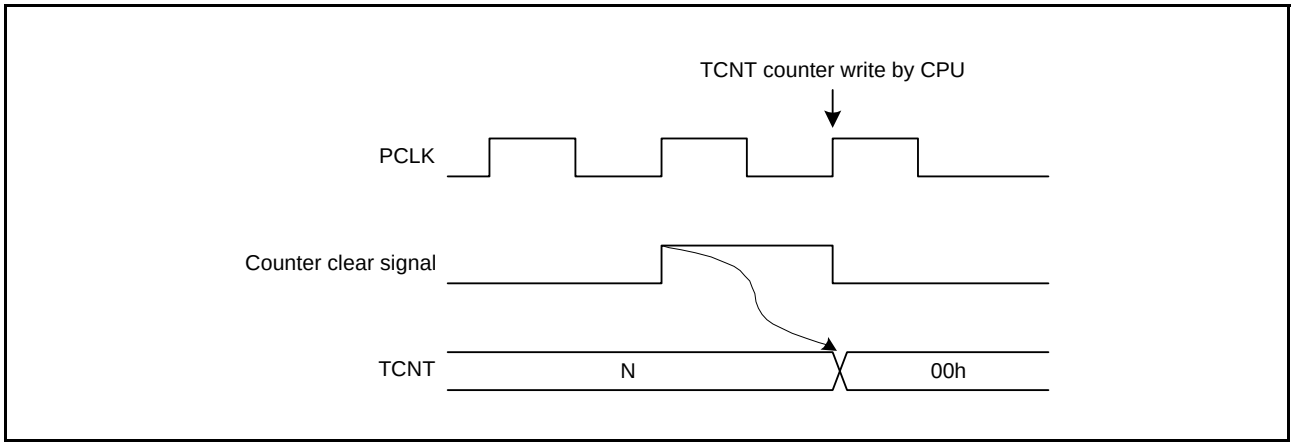


Figure 26.13 Conflict between TCNT Write and Counter Clear

26.8.4 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 26.14.

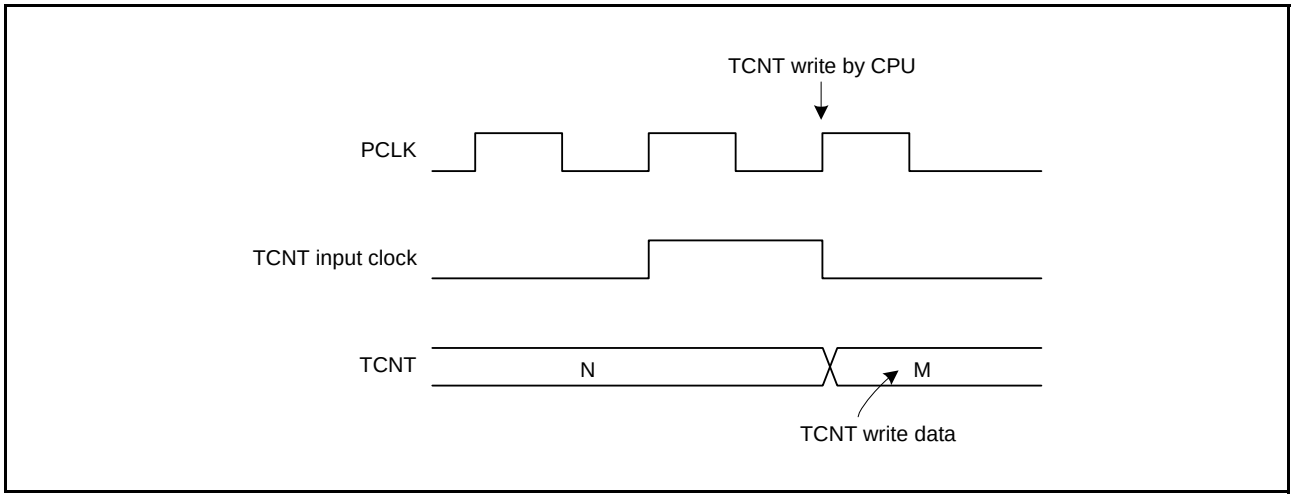


Figure 26.14 Conflict between TCNT Write and Increment

26.8.5 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 26.15, the write takes priority and the compare match signal does not reach High level.

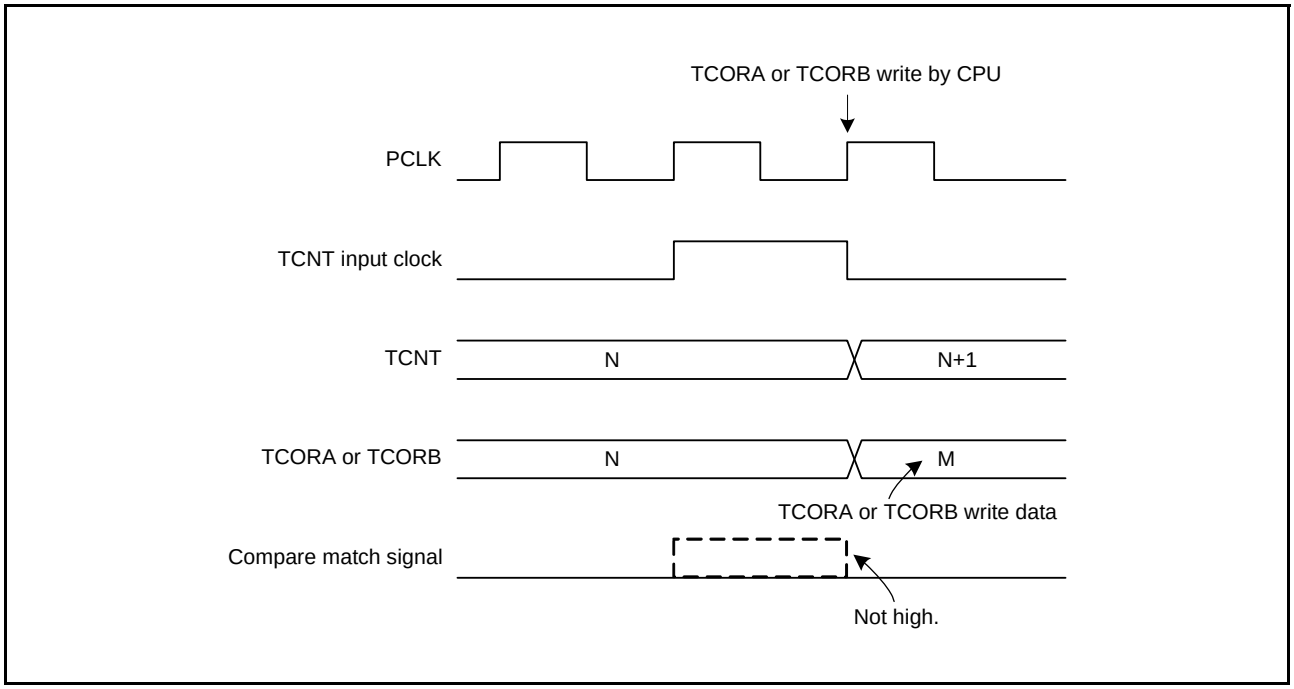


Figure 26.15 Conflict between TCORA or TCORB Write and Compare Match

26.8.6 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 statuses high for compare match A and compare match B, as listed in Table 26.7.

Table 26.7 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	↓
No change	Low

26.8.7 Switching of Frequency Dividing Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the frequency dividing clock is switched. Table 26.8 lists the relationship between the timing at which the frequency dividing clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT clock is generated from an frequency dividing clock, the rising edge of the frequency dividing 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 26.8, the change is considered as an edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented.

The erroneous increment of TCNT can also happen when switching between internal and frequency dividing clocks.

Table 26.8 Switching of Frequency Dividing Clocks and TCNT Operation (1/2)

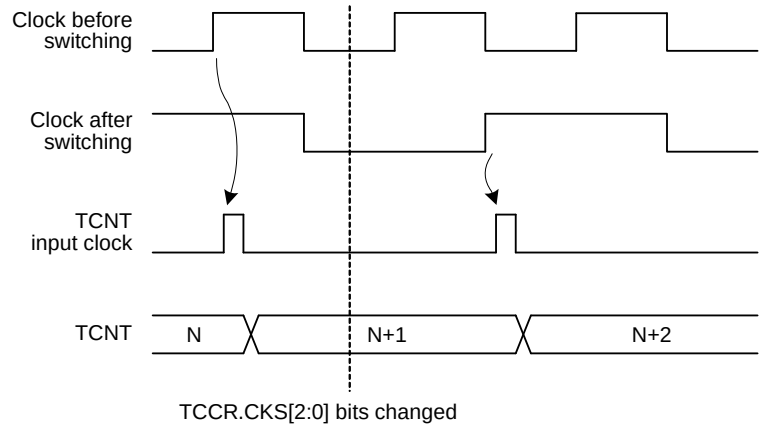
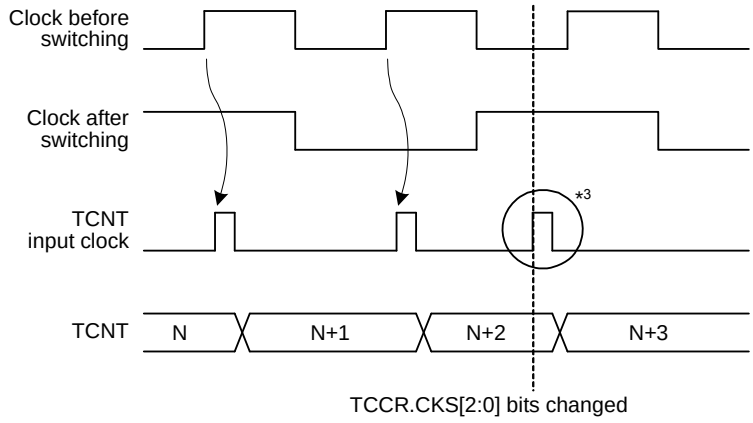
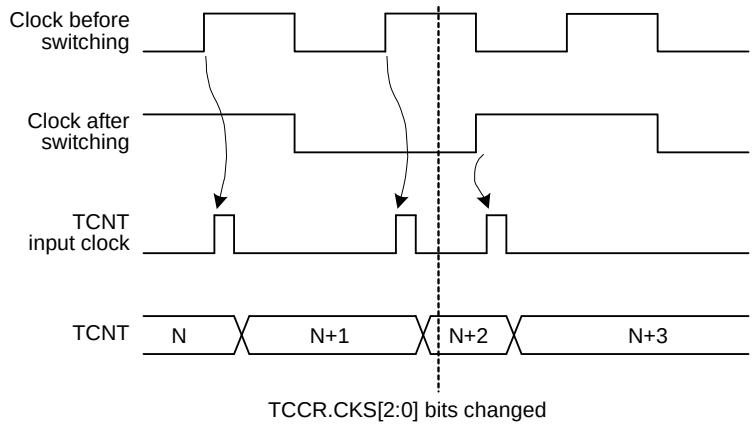
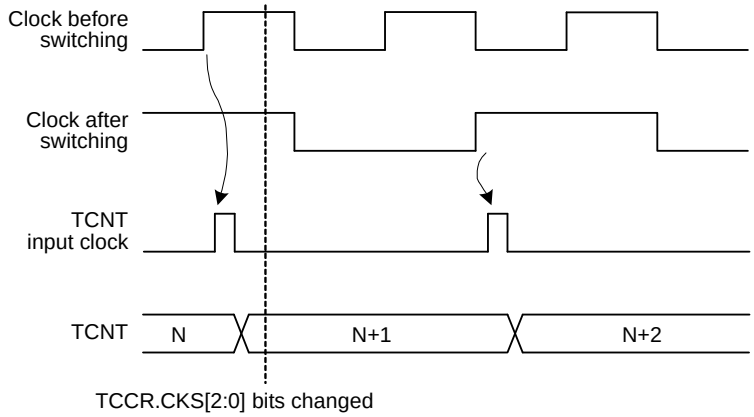
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
1	Switching from low to low*1	

Table 26.8 Switching of Frequency Dividing Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock 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 is incremented.
Note 4. Includes switching from high to stop.

26.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

26.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the frequency dividing 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 26.16 shows operation timing when the compare match interrupt signal is continuously output.

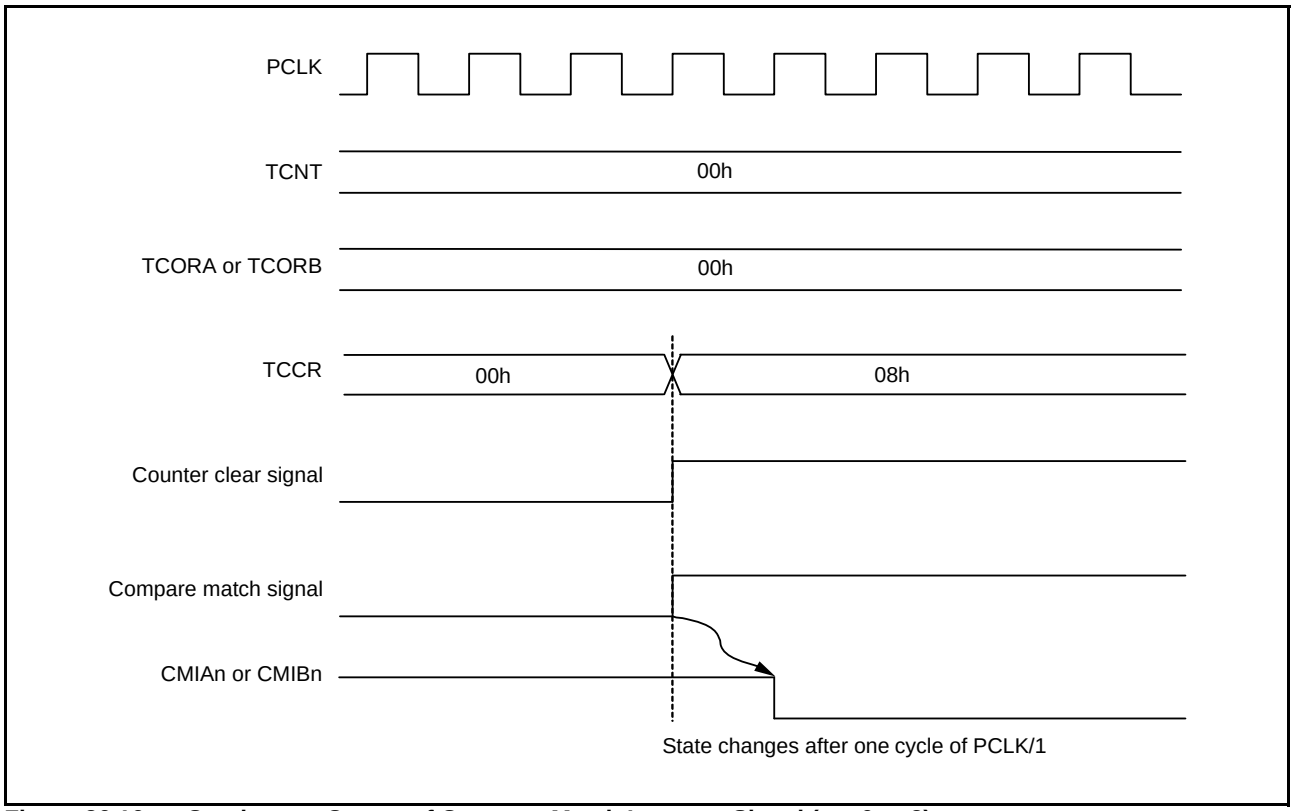


Figure 26.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

27. 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.

27.1 Overview

Table 27.1 lists the specifications for the CMT.
Figure 27.1 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 27.1 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.

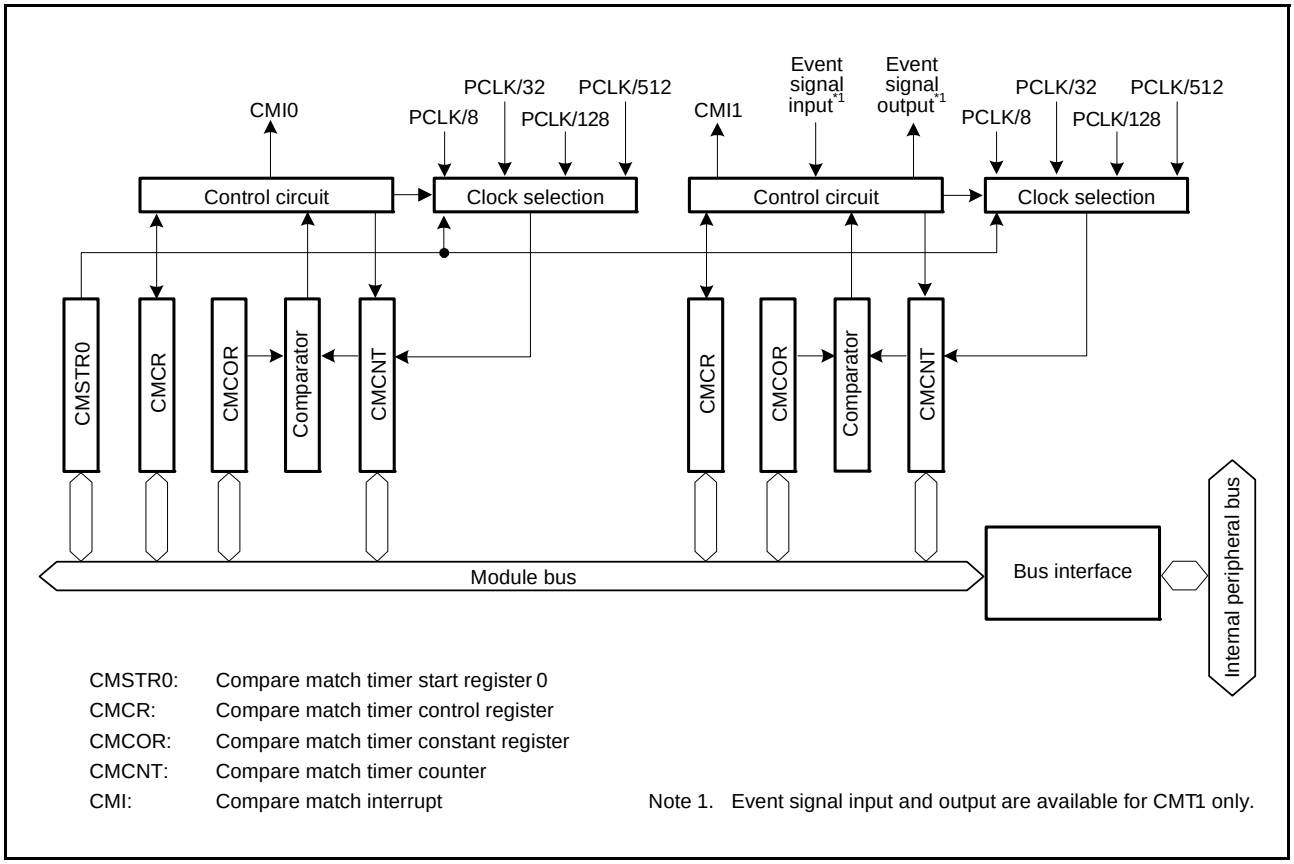


Figure 27.1 CMT (Unit 0) Block Diagram

27.2 Register Descriptions

27.2.1 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

27.2.2 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

27.2.3 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.

27.2.4 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.

27.2.5 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.

27.3 Operation

27.3.1 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 27.2 shows the operation of the CMCNT counter.

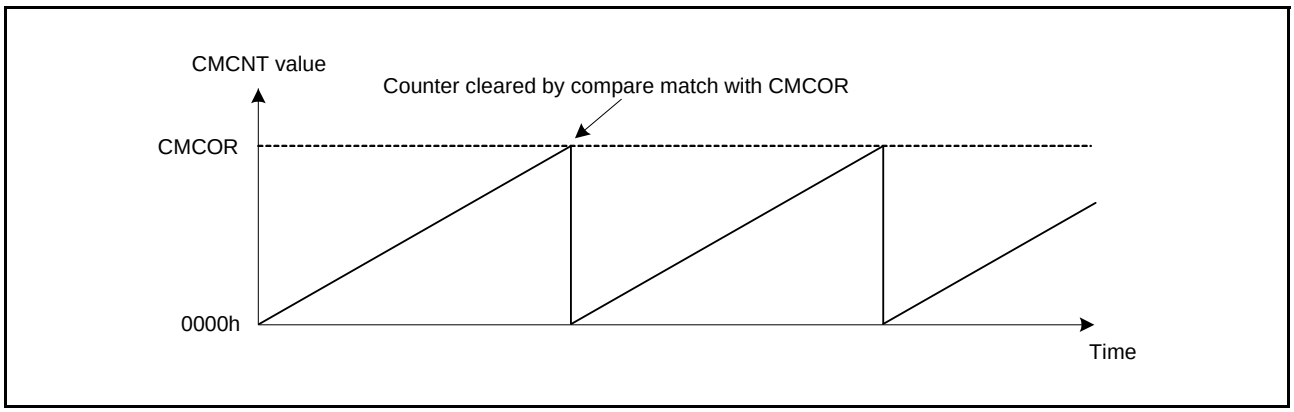


Figure 27.2 CMCNT Counter Operation

27.3.2 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 27.3 shows the timing of the CMCNT counter.

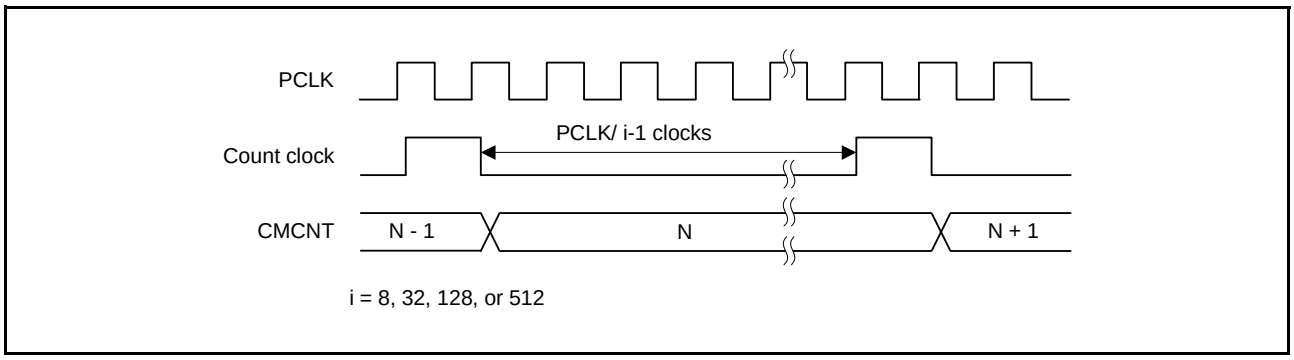


Figure 27.3 CMCNT Count Timing

27.4 Interrupts

27.4.1 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 (ICUb).

Table 27.2 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

27.4.2 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 27.4 shows the timing of a compare match interrupt.

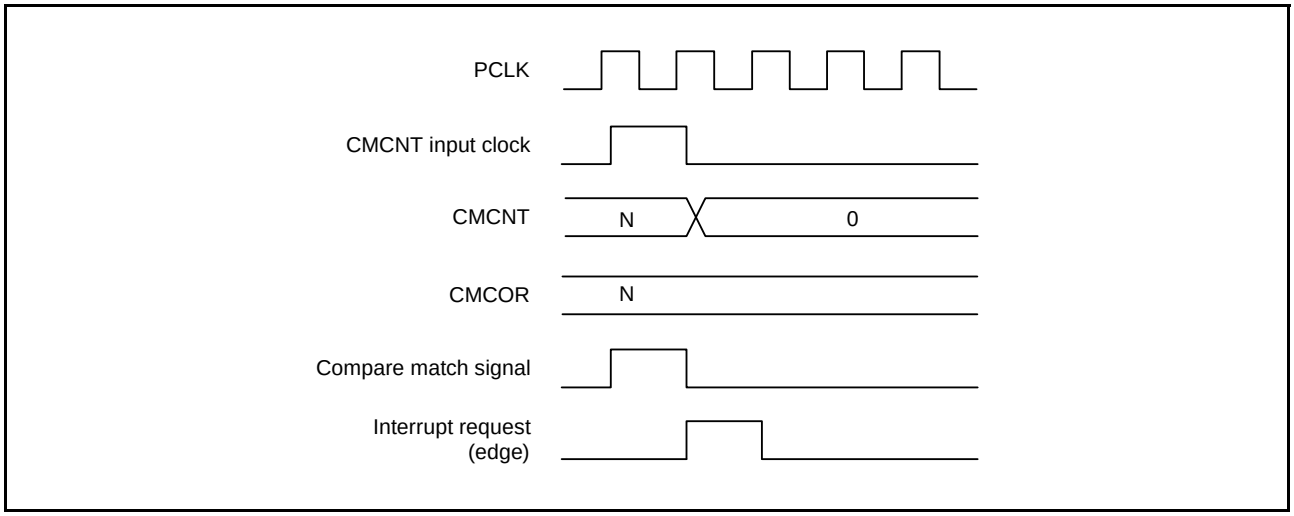


Figure 27.4 Timing of a Compare Match Interrupt

27.5 Link Operations by ELC

27.5.1 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).

27.5.2 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.

27.5.3 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.

27.6 Usage Notes

27.6.1 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.

27.6.2 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 27.5 shows the timing to clear the CMCNT counter.

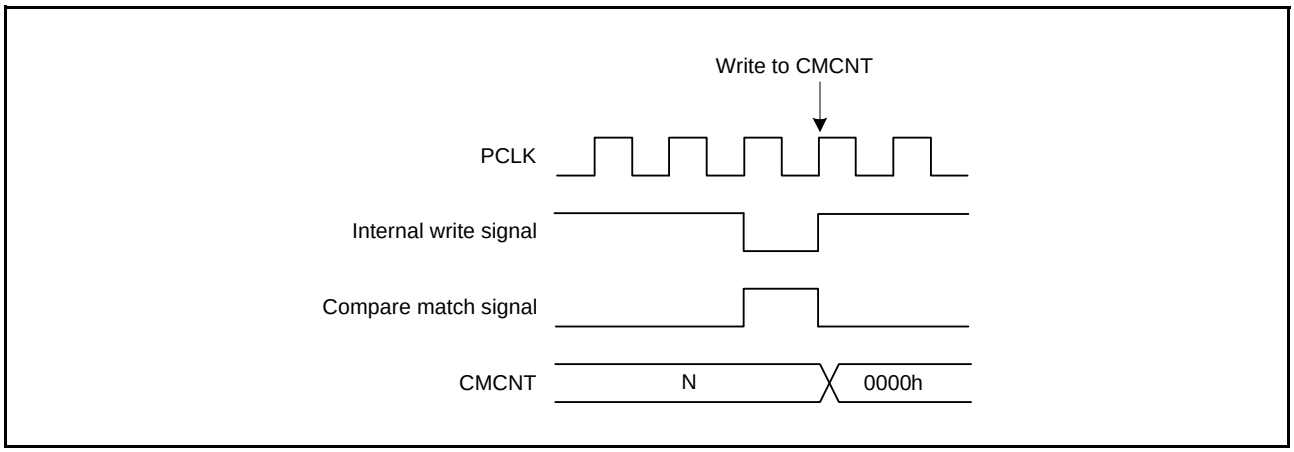


Figure 27.5 Conflict between CMCNT Counter Writing and Compare Match

27.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 27.6 shows the timing to write the CMCNT counter.

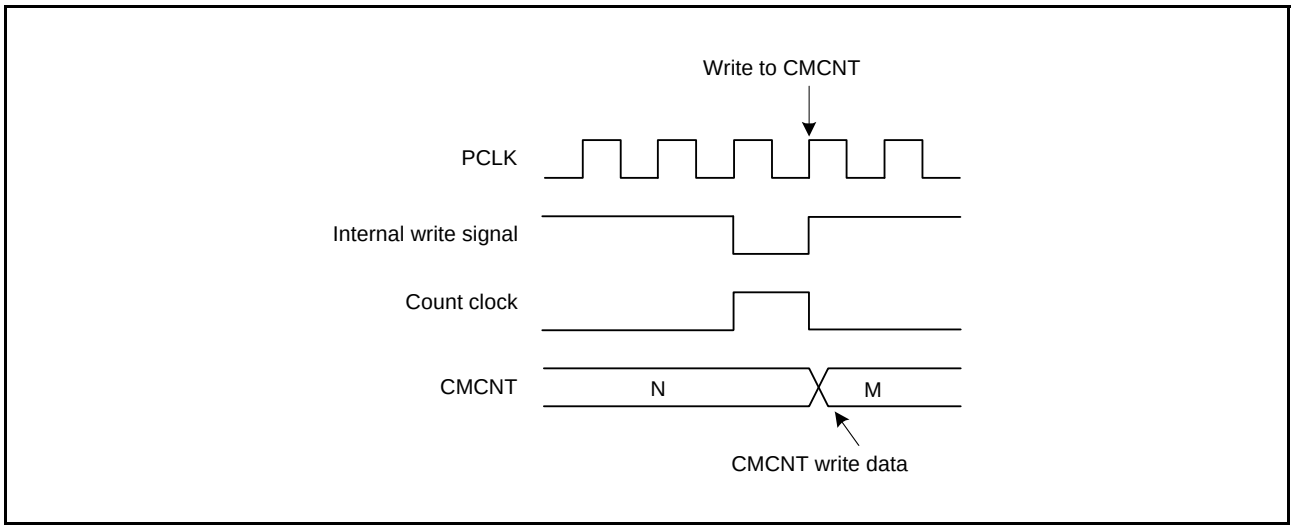


Figure 27.6 Conflict between CMCNT Counter Writing and Incrementing

28. 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.

28.1 Overview

Table 28.1 lists the specifications of the WDT and Figure 28.1 shows a block diagram of the WDT.

Table 28.1 WDT Specifications

Item	Specifications
Count source	Peripheral clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2,048, or 8,192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs • Register start mode: Counting is started by refresh operation (writing to the WDTRR register)
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated
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 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.

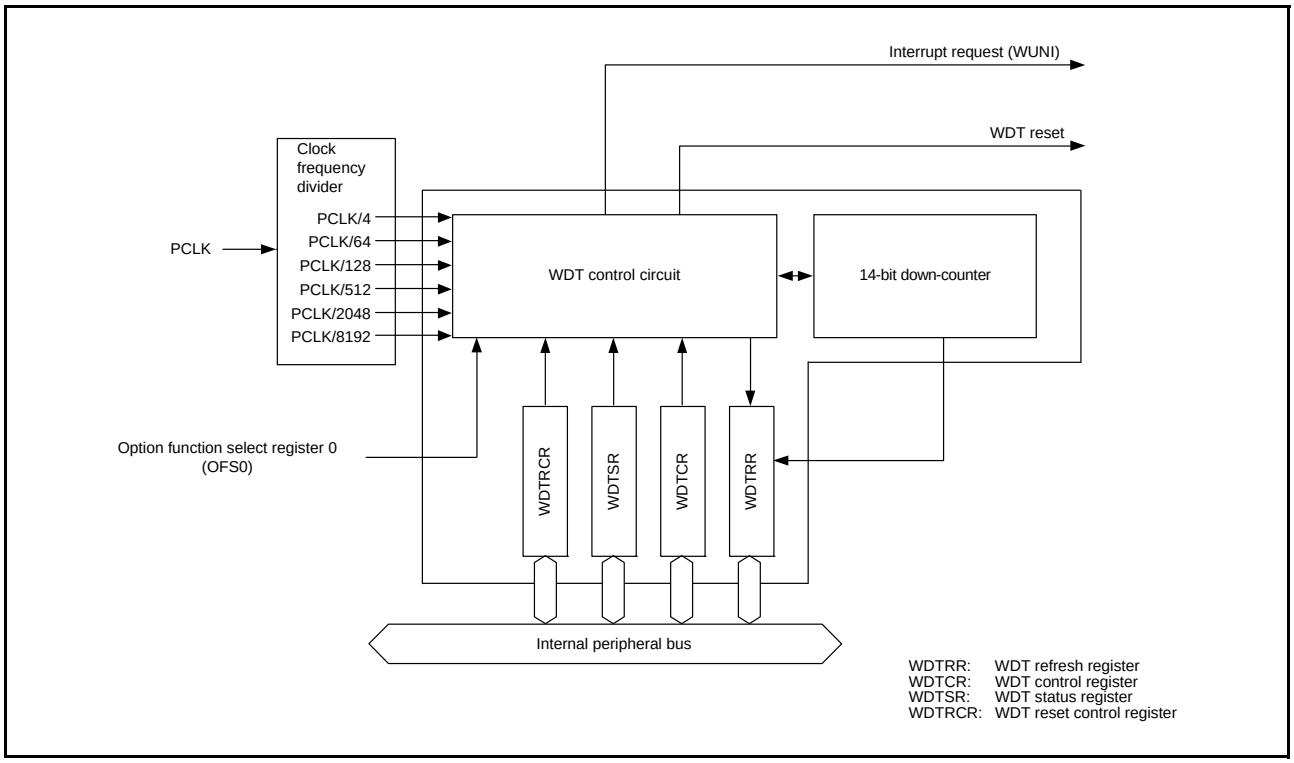
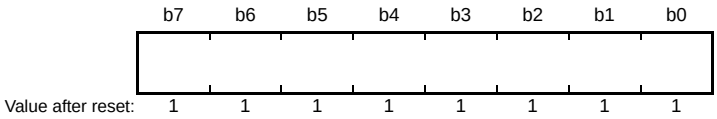


Figure 28.1 WDT Block Diagram

28.2 Register Descriptions

28.2.1 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 WDT timeout period select bits (OFS0.WDTTOPS[1:0]) in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period selection bits (WDTCR.TOPS[1:0]) in the WDT control register.
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 28.3.3, Refresh Operation.

28.2.2 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: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 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: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/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 28.3.2, 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 28.3.7, 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 1,024, 4,096, 8,192, and 16,384 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 28.2.

Table 28.2 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	PCLK/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	PCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	PCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	PCLK/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	PCLK/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	PCLK/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 clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8,192. 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. 75%, 50%, 25%, or 0% 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 28.3 lists the counter values for the window start and end positions and Figure 28.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 28.3 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%.

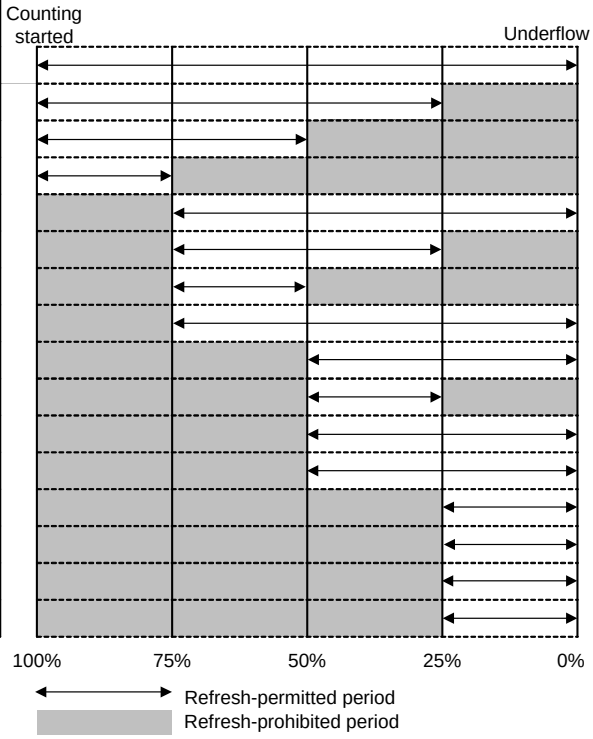


Figure 28.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

28.2.3 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.

28.2.4 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 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 28.3.2, 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 28.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

28.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 28.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

28.3 Operation

28.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing (writing to the register).
In auto-start mode, counting automatically starts after release from the reset state in accordance with the settings in option function select register 0 (OFS0) in the ROM.
In register start mode, counting is started by refreshing (writing to the register) after the respective registers are set after release from the reset state.
Select auto-start mode or register start mode by setting the WDT start mode select bit (OFS0.WDTSTRT) in the OFS0 register.
When the auto-start mode is selected, the settings in the WDT control register (WDTCR) and WDT reset control register (WDTRCR) 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 WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled.

28.3.1.1 Register Start Mode

When the WDT start mode select bit (OFS0.WDTSTRT) is 1, register start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled.
After the reset state 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 timeout period selection bits (WDTCR.TOPS[1:0]).
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 (WUNI). Reset output or interrupt request output can be selected by setting the WDT reset interrupt request selection bit (WDTRCR.RSTIRQS).

Figure 28.3 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)

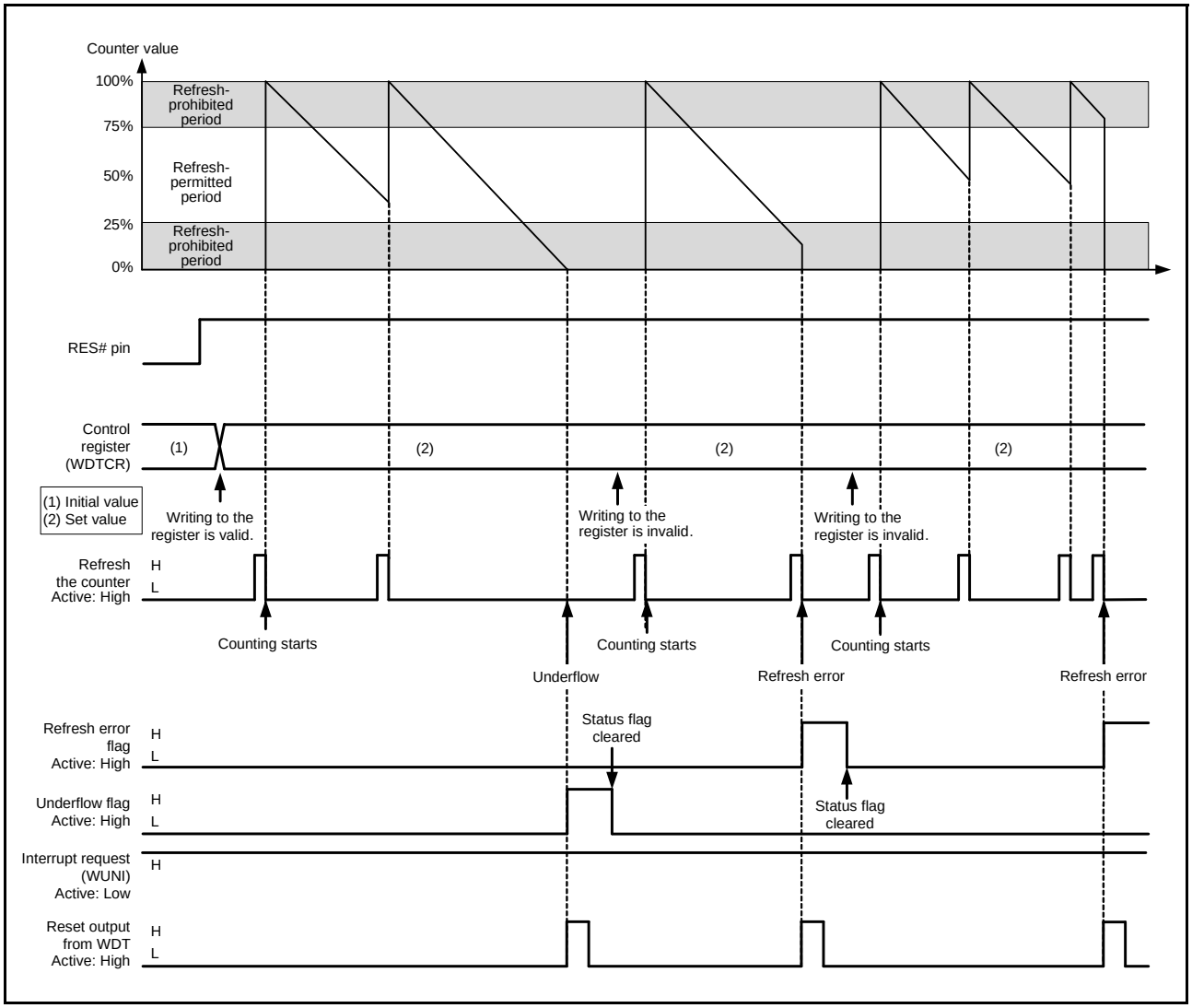


Figure 28.3 Operation Example in Register Start Mode

28.3.1.2 Auto-Start Mode

When the WDT start mode select bit (OFS0.WDTSTRT) in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDT control register (WDTCR) and WDT reset control register (WDTRCR) 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 option function select register 0 (OFS0) are set in the WDT registers.
When the reset state is released, the down-counter automatically starts counting down from the value set by the WDT timeout period select bits (OFS0.WDTTOPS[1:0]).
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 (WUNI).
After the reset signal or non-maskable 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 WDT reset interrupt request select bit (OFS0.WDTRSTIRQS).

Figure 28.4 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% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

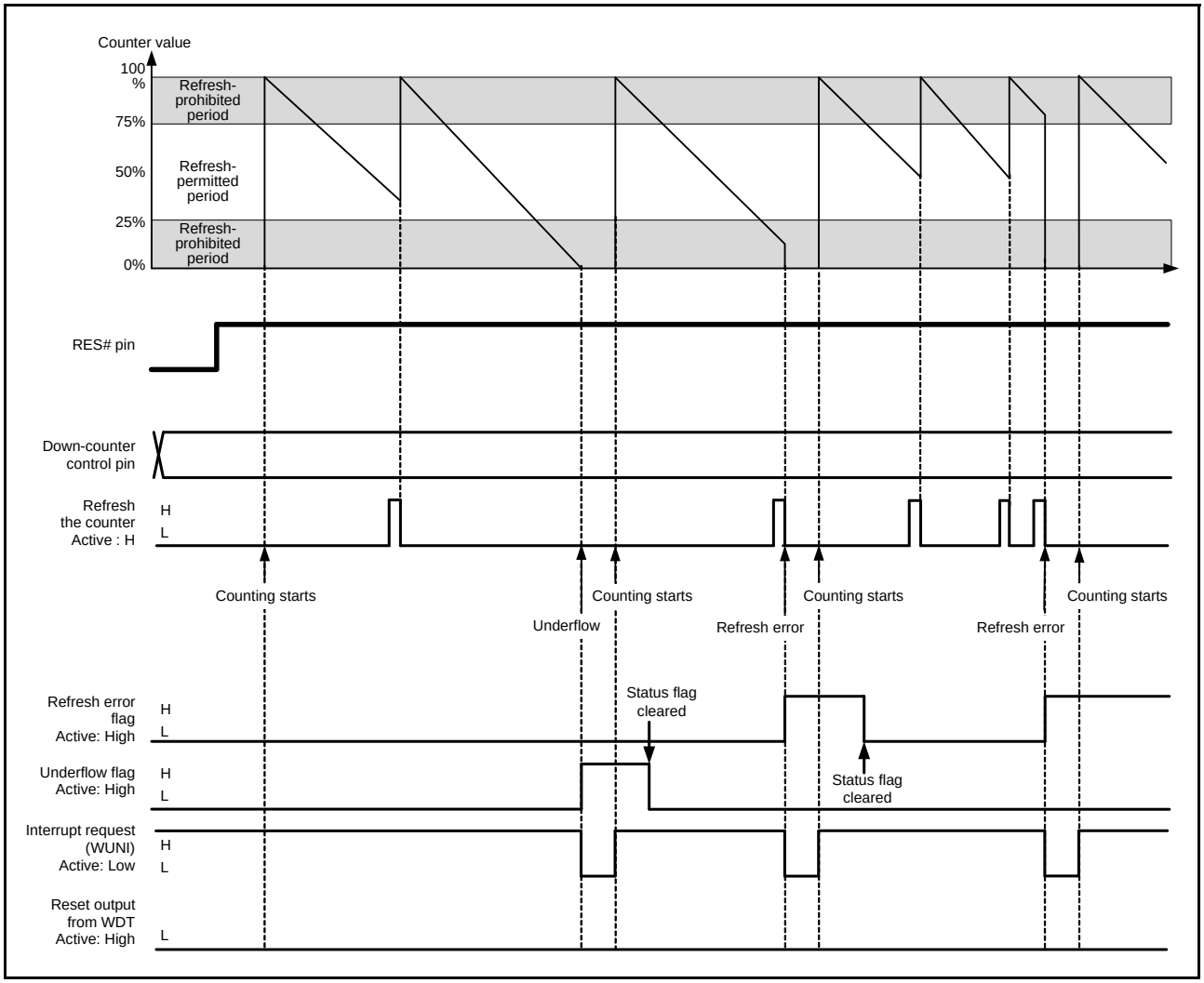


Figure 28.4 Operation Example in Auto-Start Mode

28.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDT control register (WDTCR) or WDT reset control register (WDTRCR) 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 WDTCR or WDTRCR, the protection signal in the WDT becomes 1 to protect WDTCR and WDTRCR 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 28.5 shows control waveforms produced in response to writing to the WDTCR.

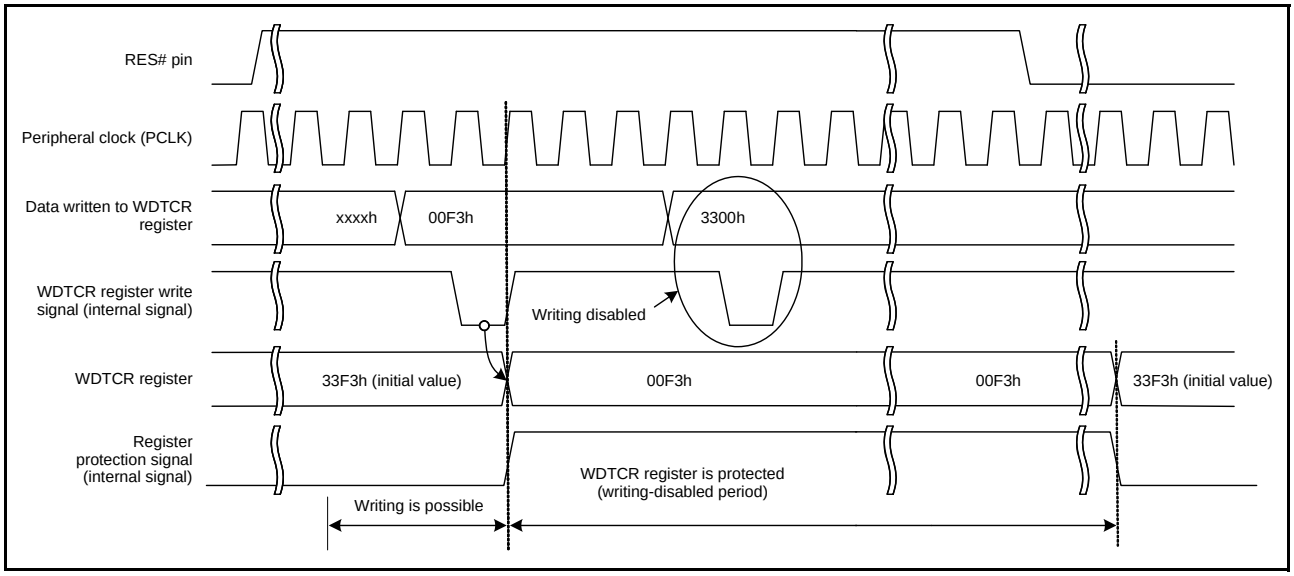


Figure 28.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

28.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDT refresh register (WDTRR). 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 WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, 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 WDTRR → 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 WDT refresh register (WDTRR), refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR should be completed four-count cycles before the down-counter underflows.

Figure 28.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

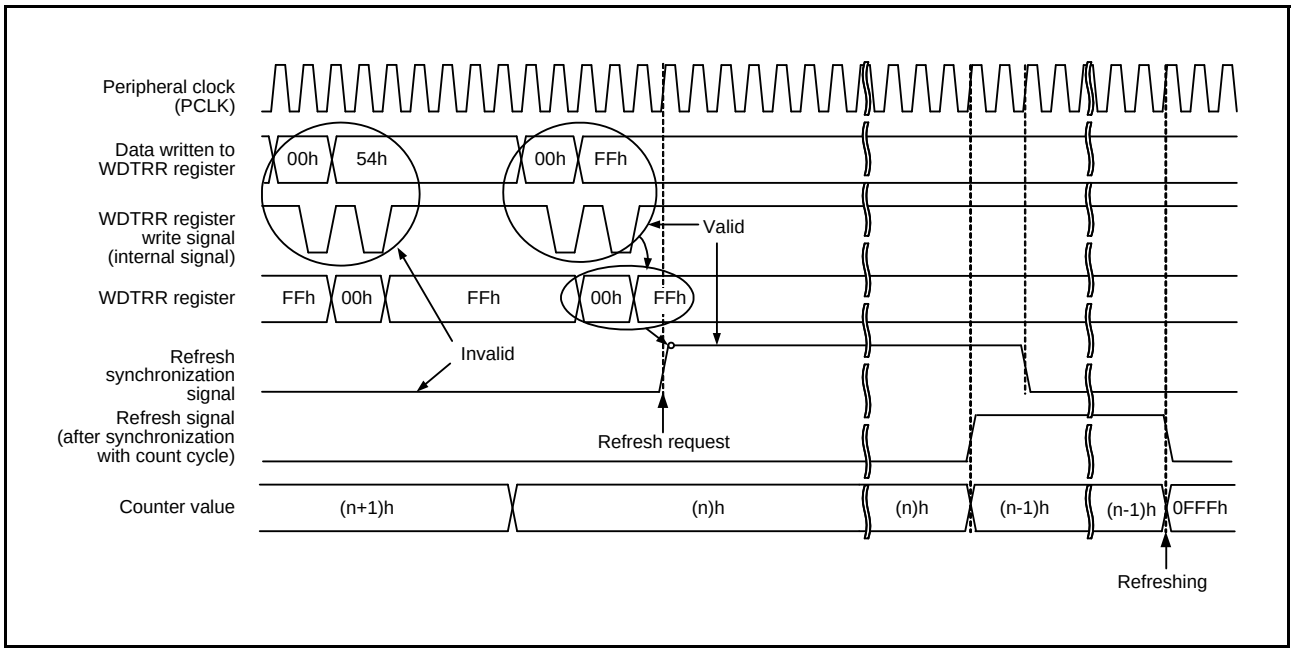


Figure 28.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.4 Reset Output

When the reset interrupt selection bit (WDTRCR.RSTIRQS) is set to 1 in register start mode or when the WDT reset interrupt request select bit (OFS0.WDTRSTIRQS) 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 state 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 state is released.

28.3.5 Interrupt Source

When the reset interrupt selection bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT reset interrupt request select bit (OFS0.WDTRSTIRQS) in option function select register 0 (OFS0) 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. For details, refer to section 14, Interrupt Controller (ICUb).

Table 28.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

28.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value (WDTSR.CNTVAL[13:0]) bits of the WDT status register. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 28.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

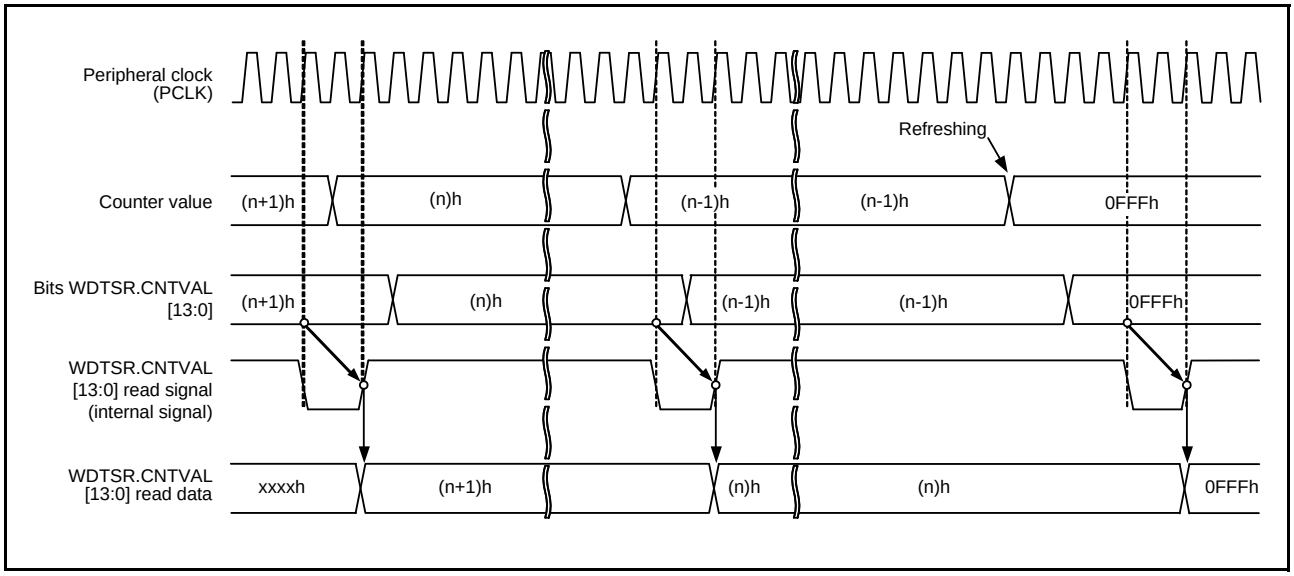


Figure 28.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

28.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 28.5 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 option function select register 0 (OFS0), refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 28.5 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

29. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

29.1 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 low-speed clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode, the IWDTCSTPR.SLCSTP bit can be used to select whether to stop the counter or not.

Table 29.1 lists the specifications of the IWDT and Figure 29.1 shows a block diagram of the IWDT.

Table 29.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Counting automatically starts after a reset (auto-start mode) • Counting is started (register start mode) by refreshing the counter (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated Counting restarts (In auto-start mode, counting automatically restarts after a reset or after a non-maskable interrupt request is output. In register start mode, counting restarts after refreshing.)
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 sources	• Down-counter underflows • When refreshing is done 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 division 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, deep software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency division 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, deep 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 clock source after division).

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 29.1 is a block diagram of the IWDT.

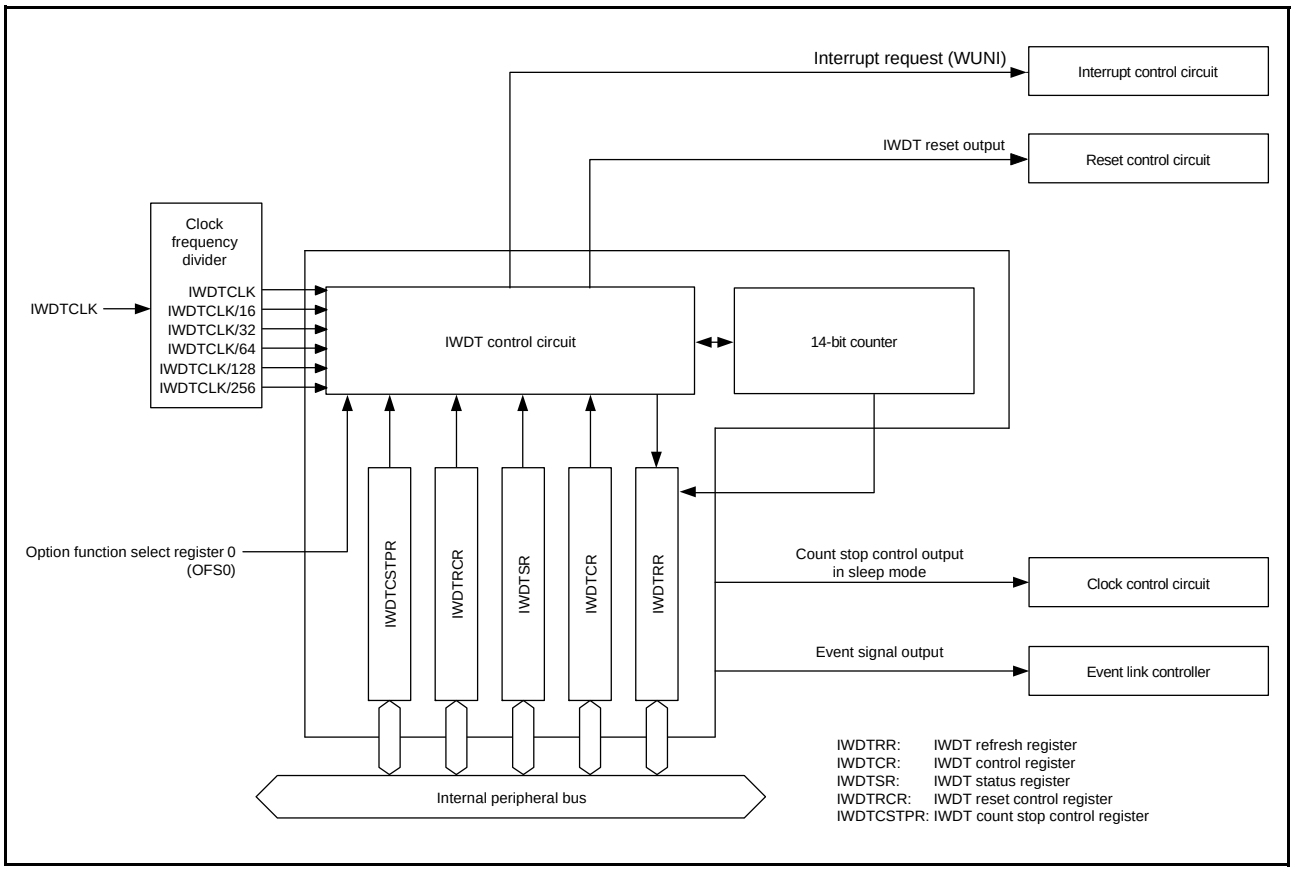
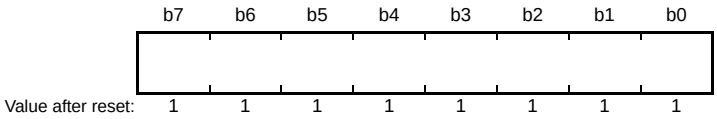


Figure 29.1 IWDT Block Diagram

29.2 Register Descriptions

29.2.1 IWDT Refresh Register (IWDTRR)

Address(es): 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

IWDTRR refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]) 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 timeout period select bits (TOPS[1:0]) in the IWDT control register (IWDTCR) in the first refresh operation after release from the reset state.
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 29.3.3, Refresh Operation.

29.2.2 IWDT Control Register (IWDTCR)

Address(es): 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 Division Ratio Select	b7 b4 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/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 29.3.2, 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 option function select register 0 (OFS0). For details, refer to section 29.3.8, 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 29.2.

Table 29.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Division Ratio Select)

These bits select the IWDTCLK clock division ratio from among division 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 29.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 29.3 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 29.2 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 ? window start setting, the window end setting is set to 0%.

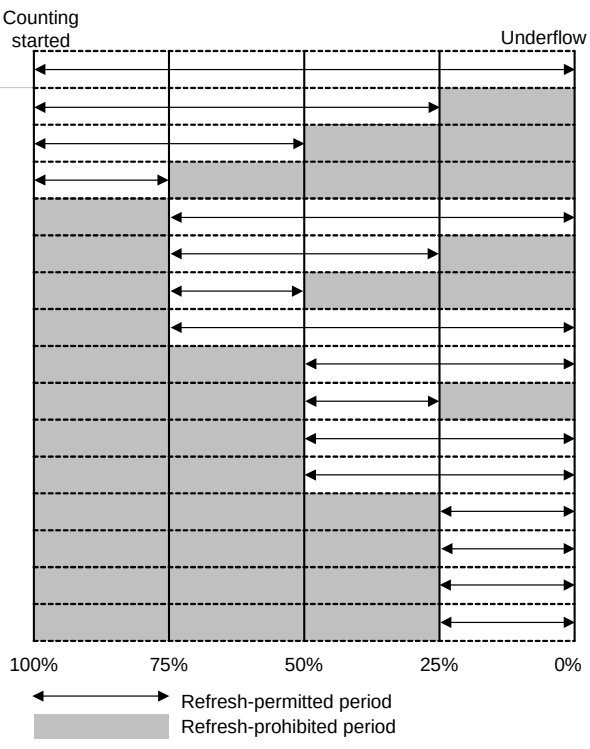


Figure 29.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

29.2.3 IWDT Status Register (IWDTSR)

Address(es): 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.

IWDTSR is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

Read these bits 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)

Read this bit 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)

Read this bit 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.

29.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): 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 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 29.3.2, 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 option function select register 0. For details, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

29.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): 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, deep software standby mode, or all-module clock stop mode.	R/W

IWDTCSTPR 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 29.3.2, 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 option function select register 0 (OFS0). For details, refer to section 29.3.8, 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, deep software standby mode, or all-module clock stop mode.

29.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

29.3 Operation

29.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0.
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled, and counting is started by refreshing (writing) the IWDT refresh register (IWDTRR). When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of option function select register 0 (OFS0) is enabled, and counting automatically starts after reset.

29.3.1.1 Register Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 1, register start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled.
After the reset state is released, set the clock division 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 timeout period select bits (IWDTCR.TOPS[1:0]).
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 (WUNI). Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 29.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The IWDT window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

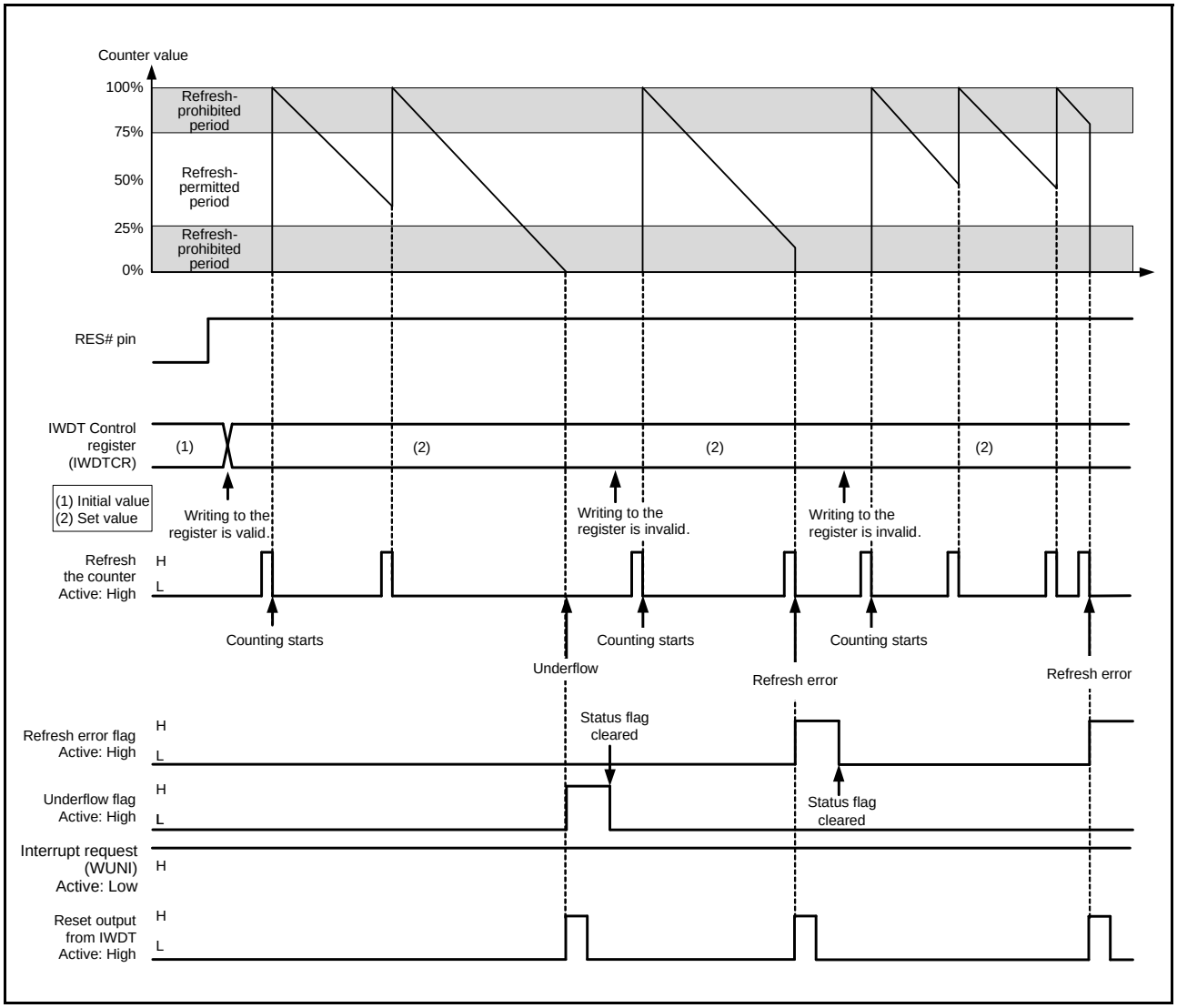


Figure 29.3 Operation Example in Register Start Mode

29.3.1.2 Auto-Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 0, auto-start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are disabled.
Within the reset state, the clock division 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 should be specified in option function select register 0 (OFS0). When the reset state is released, the counter automatically starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]).
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 (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 29.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The IWDT window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

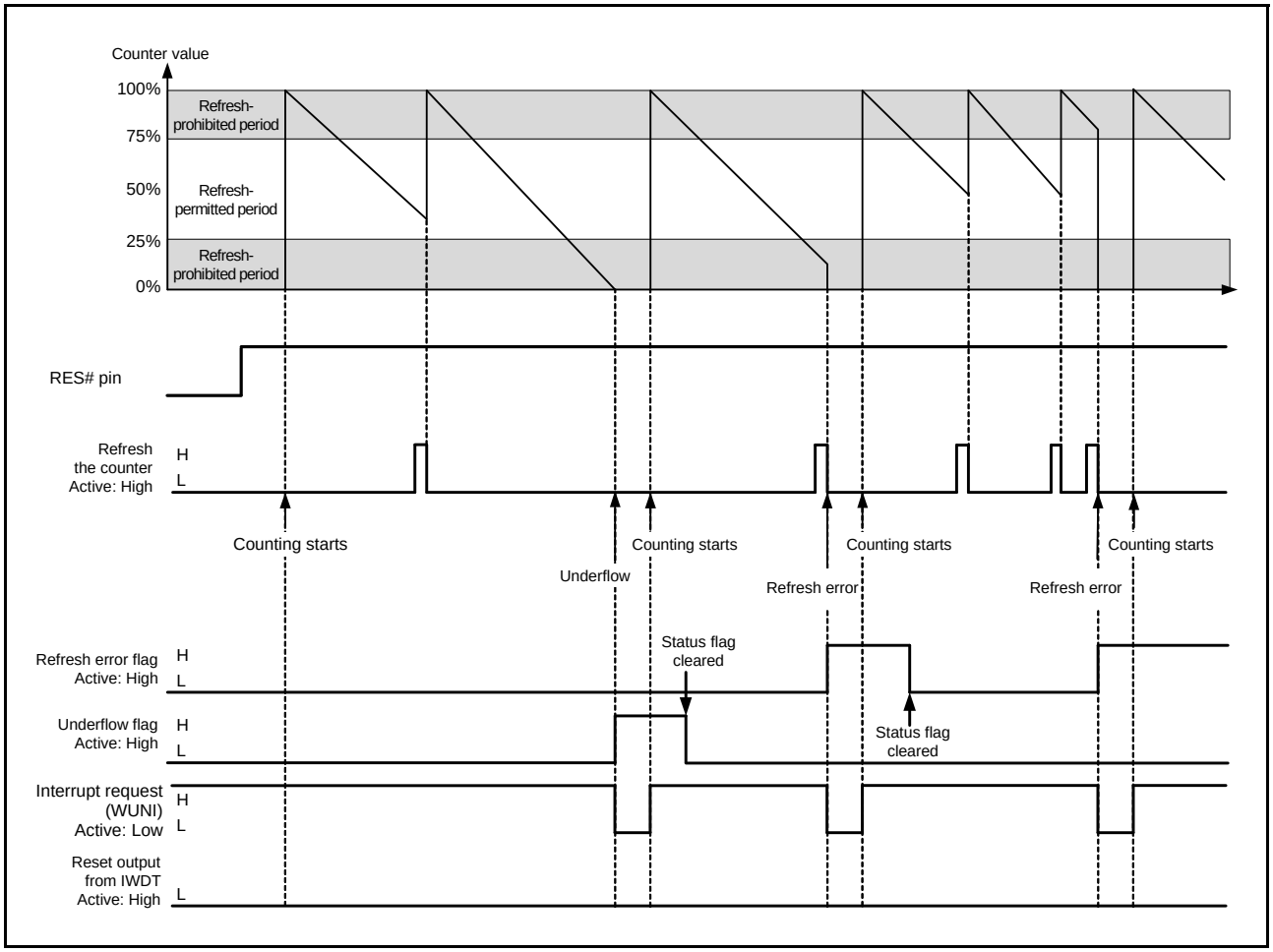


Figure 29.4 Operation Example in Auto-Start Mode

29.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), or IWDT count stop control register (IWDTCSTPR) is only possible once between the release from the reset state and the first refresh operation. After a refresh operation (counting starts) or IWDTCR, IWDTRCR, or IWDTCSTPR is written to, the protection signal in the IWDT becomes 1 to protect 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 29.5 shows control waveforms produced in response to writing to the IWDTCR register.

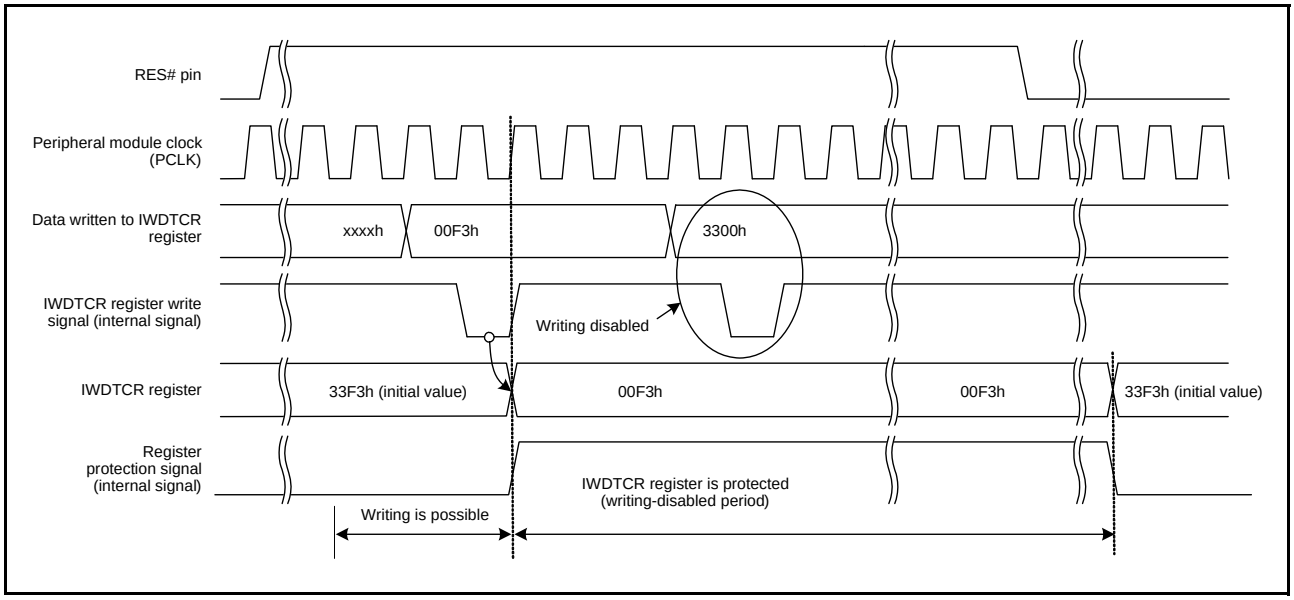


Figure 29.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

29.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). 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 IWDT refresh register (IWDTRR).
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 IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR, 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 IWDTRR → 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 IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR 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 clock division ratio selection bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR 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 counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to IWDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to IWDTRR 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 IWDTRR.
- 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 IWDTRR, no underflow occurs and refreshing is done.

Figure 29.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

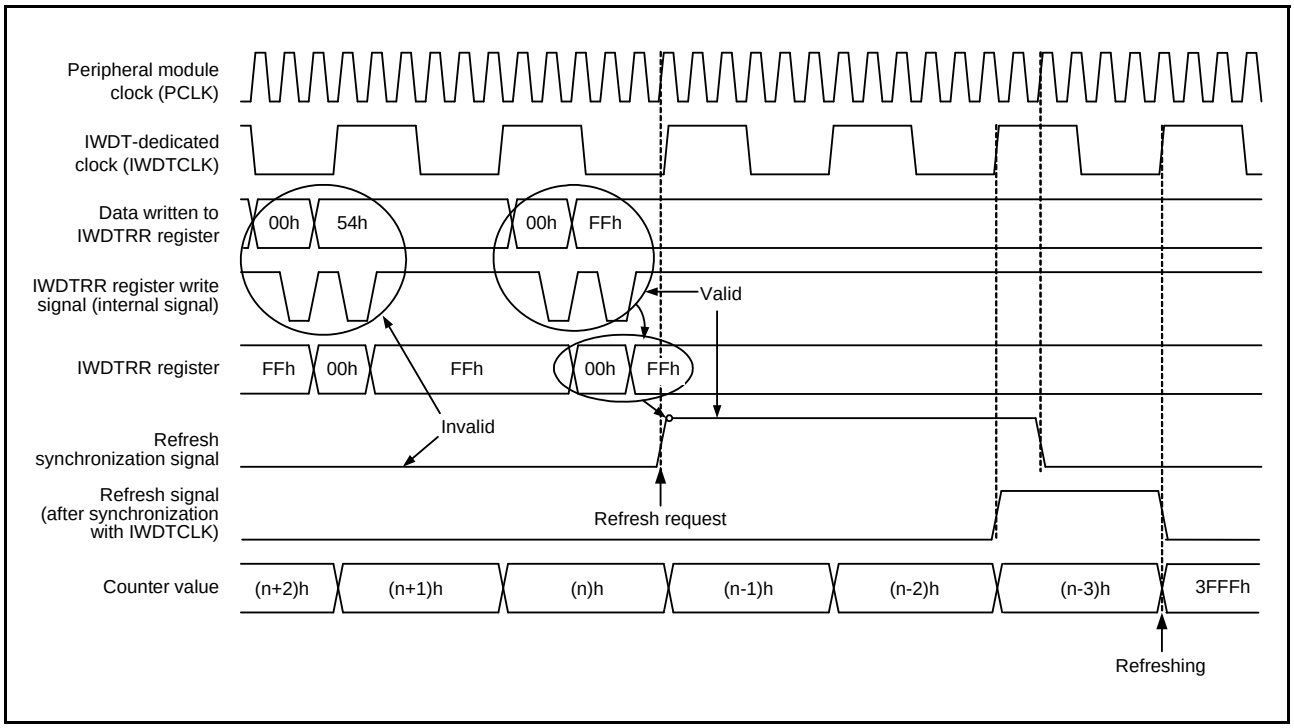


Figure 29.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

29.3.4 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (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.

29.3.5 Reset Output

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) 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 (all bits set to 0) 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.

29.3.6 Interrupt Sources

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in option function select register 0 (OFS0) is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 14, Interrupt Controller (ICUb).

Table 29.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Counter underflow Refresh error	Not possible	Not possible

29.3.7 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 counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT status register. 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 29.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

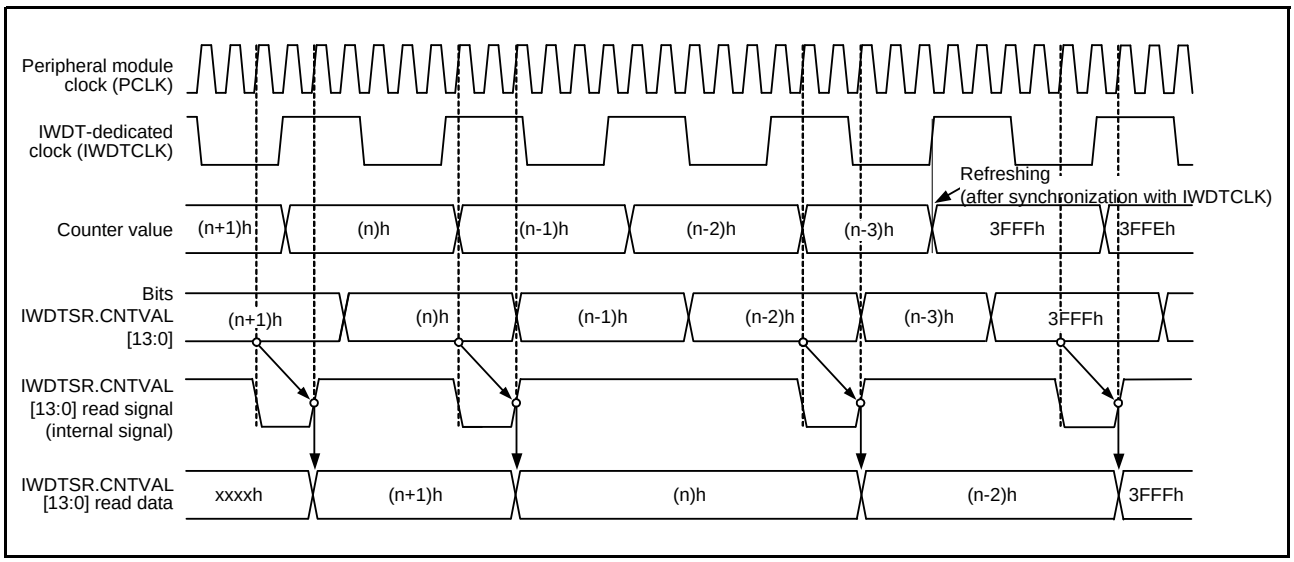


Figure 29.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

29.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 29.5 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 option function select register 0 (OFS0), refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 29.5 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 division 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

29.4 Link Operation by ELC

The IWDT is capable of link operation for the previously specified module when interrupt request signal is used as an event signal by the event link controller (ELC). The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the reset interrupt request selection bit (IWDTRCR.RSTIRQS) in register start mode or auto-start mode. An event signal can also be output upon generation of the next interrupt source while the refresh error flag (IWDTSR.REFEF) or underflow flag (IWDTSR.UNDFF) is 1.
For details, see section 19, Event Link Controller (ELC).

29.5 Usage Notes

29.5.1 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.

29.5.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count clock source after division).

30. Serial Communications Interface (SCIe, SCIf)

This MCU has 13 independent serial communications interface (SCI) channels. The SCI consists of the SCIe module (SCI0 to SCI11) and the SCIf module (SCI12).
The SCIe module (SCI0 to SCI11) 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 SCIf module includes the functions of the SCIe 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.

30.1 Overview

Table 30.1 lists the specifications of the SCIe module, Table 30.2 lists the specifications of the SCIf module, and Table 30.3 lists the specifications of the individual SCI channels.
Figure 30.1 shows block diagrams of the SCIe module, and Figure 30.3 shows the block diagram of the SCIf module.

Table 30.1 SCIe 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		See Table 30.4 to Table 30.6.
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 I^2C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7 or 8 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. (SCI5, SCI6)
	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.

Table 30.1 SCIe Specifications (2/2)

Item		Description
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 (see section 30.2.9, 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.
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 30.2 SCIf 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	See Table 30.4 to Table 30.7.
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 I^2C mode)
Low power consumption function	Module stop state can be set.

Table 30.2 SCIf Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7 or 8 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. (SCI12)
	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 (see section 30.2.9, 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.
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 • 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 • Signals received on RXDX12 can be passed though to SCIe when the extended serial mode control section is off.
	Timer function	• Usable as a reloading timer

Note 1. In simple I^2C mode, only MSB first is available.

Table 30.3 Functions of SCI Channels

Item	SCI0 to SCI4, SCI7 to SCI11	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
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

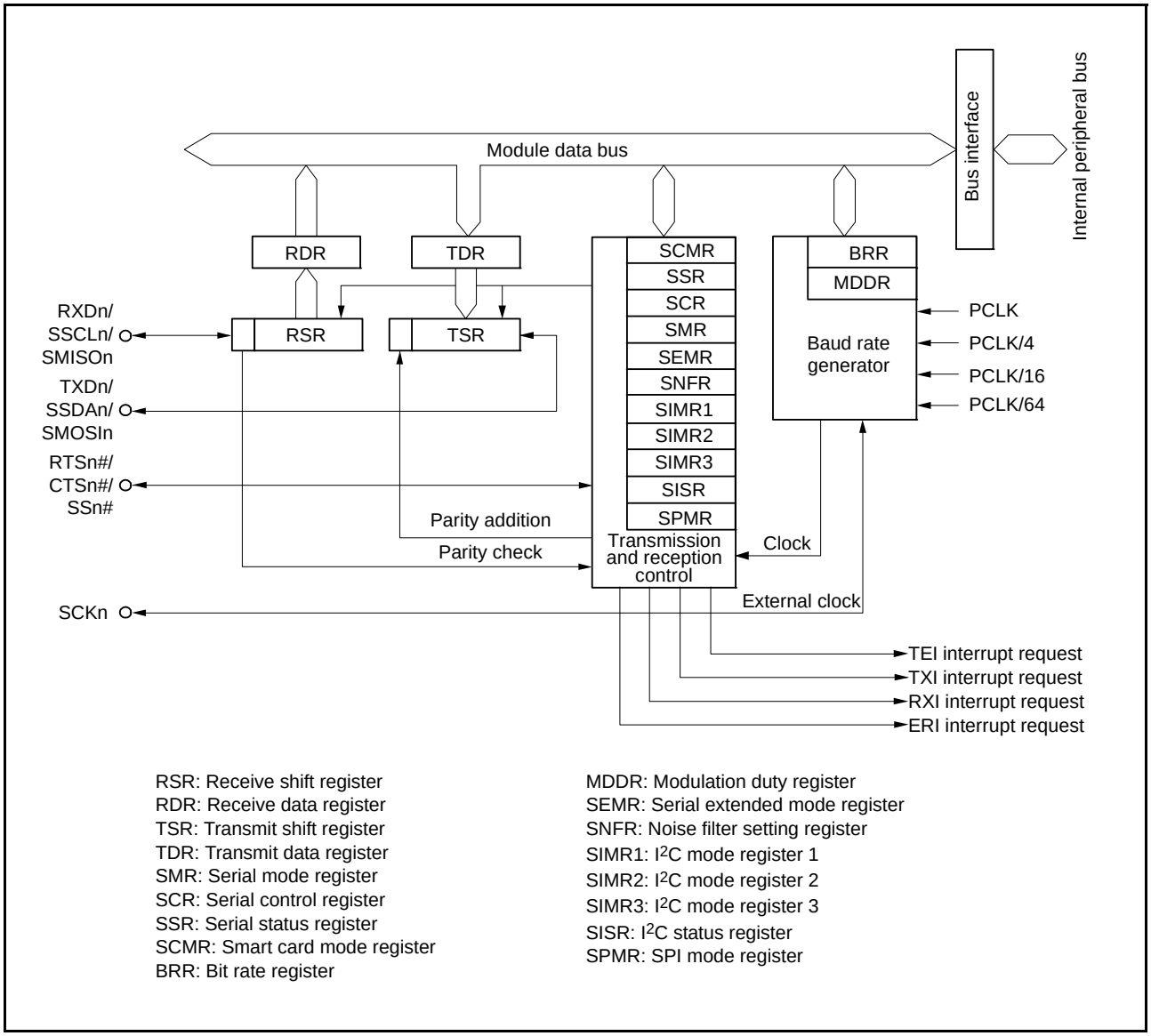


Figure 30.1 Block Diagram of SCIe (SCI0 to SCI4 and SCI7 to SCI11)

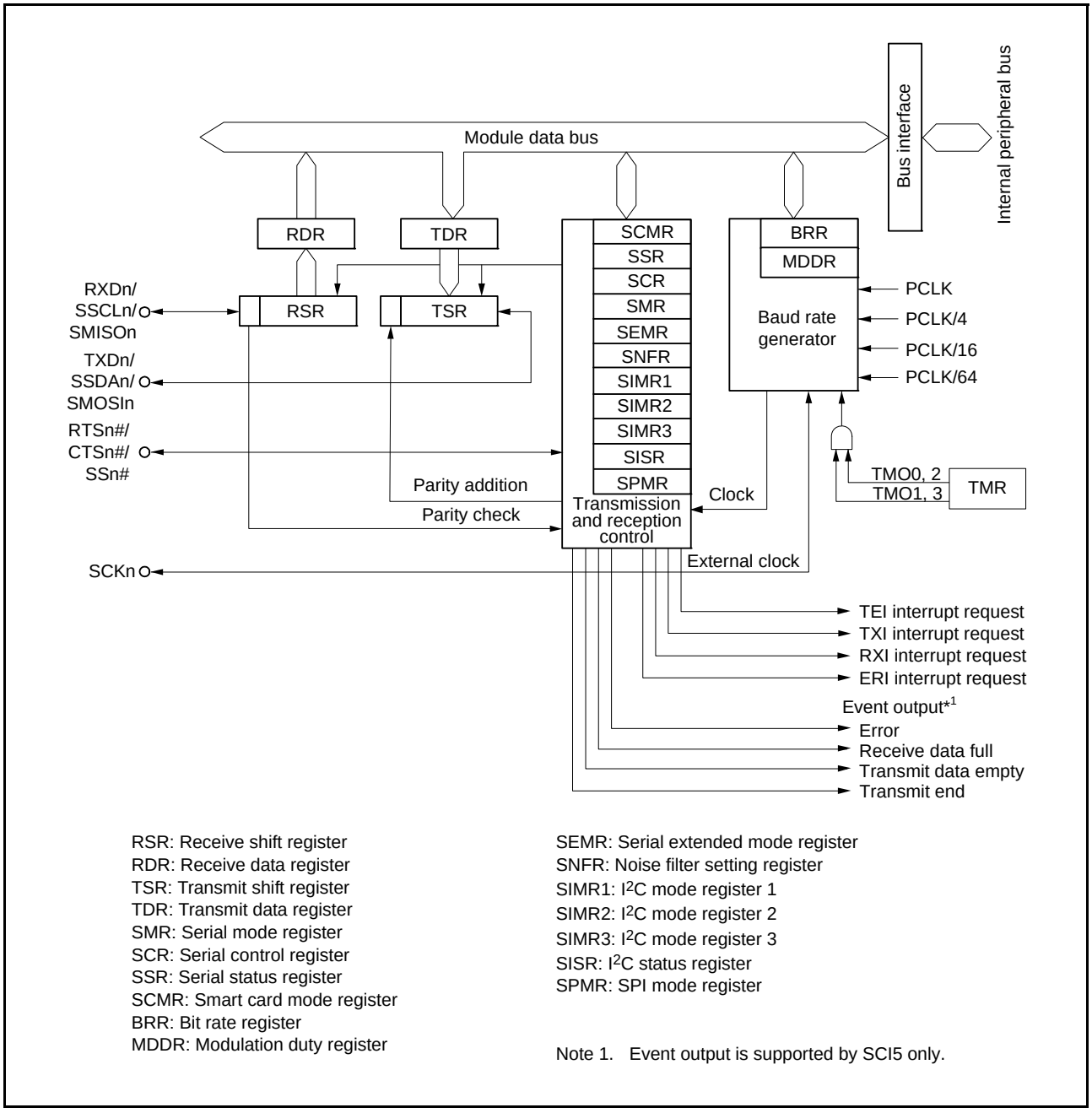


Figure 30.2 Block Diagram of SCIe (SCI5 and SCI6)

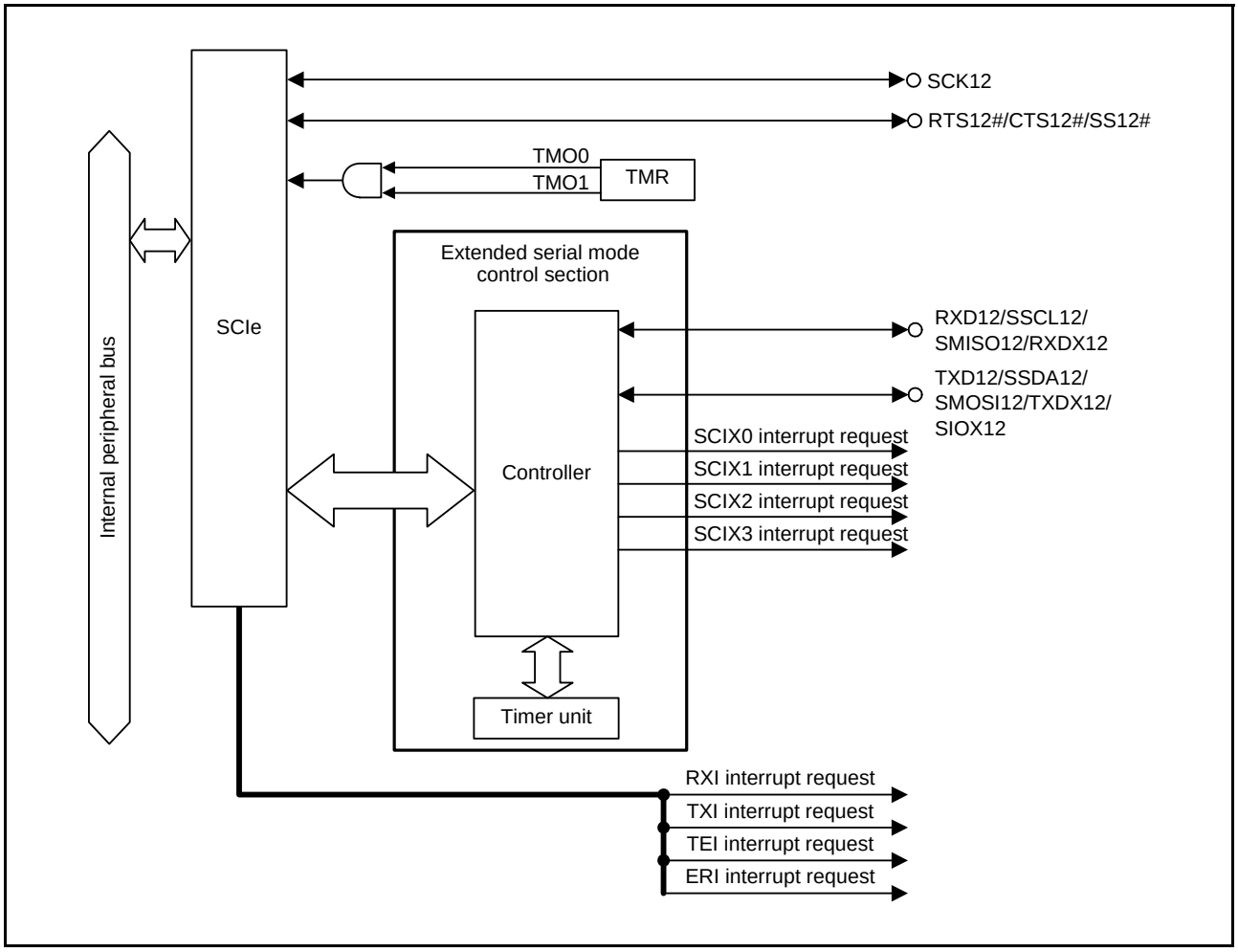


Figure 30.3 Block Diagram of SCIf (SCI12)

Table 30.4 to Table 30.7 list the pin configuration of the SCIs for the individual modes.

Table 30.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
	CTS0#/RTS0#	I/O	SCI0 transfer start control input/output
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
SCI2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
	CTS2#/RTS2#	I/O	SCI2 transfer start control input/output
SCI3	SCK3	I/O	SCI3 clock input/output
	RXD3	Input	SCI3 receive data input
	TXD3	Output	SCI3 transmit data output
	CTS3#/RTS3#	I/O	SCI3 transfer start control input/output
SCI4	SCK4	I/O	SCI4 clock input/output
	RXD4	Input	SCI4 receive data input
	TXD4	Output	SCI4 transmit data output
	CTS4#/RTS4#	I/O	SCI4 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
SCI7	SCK7	I/O	SCI7 clock input/output
	RXD7	Input	SCI7 receive data input
	TXD7	Output	SCI7 transmit data output
	CTS7#/RTS7#	I/O	SCI7 transfer start control input/output
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8	Input	SCI8 receive data input
	TXD8	Output	SCI8 transmit data output
	CTS8#/RTS8#	I/O	SCI8 transfer start control input/output
SCI9	SCK9	I/O	SCI9 clock input/output
	RXD9	Input	SCI9 receive data input
	TXD9	Output	SCI9 transmit data output
	CTS9#/RTS9#	I/O	SCI9 transfer start control input/output

Table 30.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI10	SCK10	I/O	SCI10 clock input/output
	RXD10	Input	SCI10 receive data input
	TXD10	Output	SCI10 transmit data output
	CTS10#/RTS10#	I/O	SCI10 transfer start control input/output
SCI11	SCK11	I/O	SCI11 clock input/output
	RXD11	Input	SCI11 receive data input
	TXD11	Output	SCI11 transmit data output
	CTS11#/RTS11#	I/O	SCI11 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 30.5 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI0	SSCL0	I/O	SCI0 I^2C clock input/output
	SSDA0	I/O	SCI0 I^2C data input/output
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI2	SSCL2	I/O	SCI2 I^2C clock input/output
	SSDA2	I/O	SCI2 I^2C data input/output
SCI3	SSCL3	I/O	SCI3 I^2C clock input/output
	SSDA3	I/O	SCI3 I^2C data input/output
SCI4	SSCL4	I/O	SCI4 I^2C clock input/output
	SSDA4	I/O	SCI4 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
SCI7	SSCL7	I/O	SCI7 I^2C clock input/output
	SSDA7	I/O	SCI7 I^2C data input/output
SCI8	SSCL8	I/O	SCI8 I^2C clock input/output
	SSDA8	I/O	SCI8 I^2C data input/output
SCI9	SSCL9	I/O	SCI9 I^2C clock input/output
	SSDA9	I/O	SCI9 I^2C data input/output
SCI10	SSCL10	I/O	SCI10 I^2C clock input/output
	SSDA10	I/O	SCI10 I^2C data input/output
SCI11	SSCL11	I/O	SCI11 I^2C clock input/output
	SSDA11	I/O	SCI11 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 30.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	SMISO0	I/O	SCI0 slave transmit data input/output
	SMOSI0	I/O	SCI0 master transmit data input/output
	SS0#	Input	SCI0 chip select input
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
SCI2	SCK2	I/O	SCI2 clock input/output
	SMISO2	I/O	SCI2 slave transmit data input/output
	SMOSI2	I/O	SCI2 master transmit data input/output
	SS2#	Input	SCI2 chip select input
SCI3	SCK3	I/O	SCI3 clock input/output
	SMISO3	I/O	SCI3 slave transmit data input/output
	SMOSI3	I/O	SCI3 master transmit data input/output
	SS3#	Input	SCI3 chip select input
SCI4	SCK4	I/O	SCI4 clock input/output
	SMISO4	I/O	SCI4 slave transmit data input/output
	SMOSI4	I/O	SCI4 master transmit data input/output
	SS4#	Input	SCI4 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
SCI7	SCK7	I/O	SCI7 clock input/output
	SMISO7	I/O	SCI7 slave transmit data input/output
	SMOSI7	I/O	SCI7 master transmit data
	SS7#	Input	SCI7 chip select input
SCI8	SCK8	I/O	SCI8 clock input/output
	SMISO8	I/O	SCI8 slave transmit data input/output
	SMOSI8	I/O	SCI8 master transmit data input/output
	SS8#	Input	SCI8 chip select input
SCI9	SCK9	I/O	SCI9 clock input/output
	SMISO9	I/O	SCI9 slave transmit data input/output
	SMOSI9	I/O	SCI9 master transmit data input/output
	SS9#	Input	SCI9 chip select input
SCI10	SCK10	I/O	SCI10 clock input/output
	SMISO10	I/O	SCI10 slave transmit data input/output
	SMOSI10	I/O	SCI10 master transmit data input/output
	SS10#	Input	SCI10 chip select input

Table 30.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
SCI11	SCK11	I/O	SCI11 clock input/output
	SMISO11	I/O	SCI11 slave transmit data input/output
	SMOSI11	I/O	SCI11 master transmit data input/output
	SS11#	Input	SCI11 chip select input
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 30.7 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

30.2 Register Descriptions

30.2.1 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.

30.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 0008 A005h, SCI1.RDR 0008 A025h, SCI2.RDR 0008 A045h, SCI3.RDR 0008 A065h, SCI4.RDR 0008 A085h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI7.RDR 0008 A0E5h, SCI8.RDR 0008 A105h, SCI9.RDR 0008 A125h, SCI10.RDR 0008 A145h, SCI11.RDR 0008 A165h, SCI12.RDR 0008 B305h

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

RDR 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 RSR to RDR. Then the RSR register can receive the next data.
Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.
Read RDR 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 RDR, an overrun error occurs.
RDR cannot be written to by the CPU.

30.2.3 Transmit Data Register (TDR)

Address(es): SCI0.TDR 0008 A003h, SCI1.TDR 0008 A023h, SCI2.TDR 0008 A043h, SCI3.TDR 0008 A063h, SCI4.TDR 0008 A083h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, SCI7.TDR 0008 A0E3h, SCI8.TDR 0008 A103h, SCI9.TDR 0008 A123h, SCI10.TDR 0008 A143h, SCI11.TDR 0008 A163h, SCI12.TDR 0008 B303h

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

TDR is an 8-bit register that stores transmit data.
When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.
The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.
The CPU is able to read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (TXI).

30.2.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.
TSR cannot be directly accessed by the CPU.

30.2.5 Serial Mode Register (SMR)

Note: Some bits in SMR 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): SCI0.SMR 0008 A000h, SCI1.SMR 0008 A020h, SCI2.SMR 0008 A040h, SCI3.SMR 0008 A060h, SCI4.SMR 0008 A080h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI7.SMR 0008 A0E0h, SCI8.SMR 0008 A100h, SCI9.SMR 0008 A120h, SCI10.SMR 0008 A140h, SCI11.SMR 0008 A160h, 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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (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) 0: Selects 8 bits as the data length*2 1: Selects 7 bits as the data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous 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 BRR (refer to section 30.2.9, 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 TDR is not transmitted in transmission.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 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 30.2.9, 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.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI0.SMR 0008 A000h, SCI1.SMR 0008 A020h, SCI2.SMR 0008 A040h, SCI3.SMR 0008 A060h, SCI4.SMR 0008 A080h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI7.SMR 0008 A0E0h, SCI8.SMR 0008 A100h, SCI9.SMR 0008 A120h, SCI10.SMR 0008 A140h, SCI11.SMR 0008 A160h, SCI12.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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 30.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*3
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*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (refer to section 30.2.9, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (refer to section 30.2.9, Bit Rate Register (BRR)).
Note 3. Writable only when TE in SCR = 0 and RE in SCR = 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 30.2.9, 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 30.6.4, Receive Data Sampling Timing and Reception Margin.

Table 30.8 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 BRR (refer to section 30.2.9, 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 30.6.2, 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 30.6.3, 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 30.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 30.6.8, Clock Output Control.

30.2.6 Serial Control Register (SCR)

Note: Some bits in SCR 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): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, 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	• For SCI0 to SCI4 and SCI7 to SCI11 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. 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 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. (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
b1, b0	CKE[1:0]	Clock Enable	• For SCI5, SCI6 and SCI12 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port according to 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 or TMR clock • 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 TMR clock can be used. The SCKn pin is available for use as an I/O port according to the I/O port settings when the TMR clock 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

Bit	Symbol	Bit Name	Description	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when SMR.MP = 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 ORER and FER in SSR 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*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, 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 ORER and FER in SSR to 1 is disabled. 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 30.4, Multi-Processor Communications Function.

When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, 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 RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set 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 SMR 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 SSR 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 TDR. Note that SMR 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 SSR and then setting the flag to 0, or setting the RIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, 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	• When SMR.GM = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port according to the I/O port settings.) 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 30.11, 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 30.6.8, 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 SMR 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 SSR 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 TDR. Note that SMR 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 SSR and then setting the flag to 0, or setting the RIE bit to 0.

30.2.7 Serial Status Register (SSR)

Note: Some bits in SSR 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): SCI0.SSR 0008 A004h, SCI1.SSR 0008 A024h, SCI2.SSR 0008 A044h, SCI3.SSR 0008 A064h, SCI4.SSR 0008 A084h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI7.SSR 0008 A0E4h, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI10.SSR 0008 A144h, SCI11.SSR 0008 A164h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

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
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

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 RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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
  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 RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 RDR
  In RDR, 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 ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI0.SSR 0008 A004h, SCI1.SSR 0008 A024h, SCI2.SSR 0008 A044h, SCI3.SSR 0008 A064h, SCI4.SSR 0008 A084h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI7.SSR 0008 A0E4h, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI10.SSR 0008 A144h, SCI11.SSR 0008 A164h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

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
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

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 = 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 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

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 RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 ERS after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  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 RDR
  In RDR, 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 ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

30.2.8 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 0008 A006h, SCI1.SCMR 0008 A026h, SCI2.SCMR 0008 A046h, SCI3.SCMR 0008 A066h, SCI4.SCMR 0008 A086h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI7.SCMR 0008 A0E6h, SCI8.SCMR 0008 A106h, SCI9.SCMR 0008 A126h, SCI10.SCMR 0008 A146h, SCI11.SCMR 0008 A166h, SCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	—	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	0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction	This bit can be used in the following modes. • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode Set this bit to 1 if operation is to be in simple I2C mode. 0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b6 to b4	—	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 30.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

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)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

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 30.9 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 BRR (refer to section 30.2.9, Bit Rate Register (BRR)).

30.2.9 Bit Rate Register (BRR)

Address(es): SCI0.BRR 0008 A001h, SCI1.BRR 0008 A021h, SCI2.BRR 0008 A041h, SCI3.BRR 0008 A061h, SCI4.BRR 0008 A081h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI7.BRR 0008 A0E1h, SCI8.BRR 0008 A101h, SCI9.BRR 0008 A121h, SCI10.BRR 0008 A141h, SCI11.BRR 0008 A161h, SCI12.BRR 0008 B301h

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

BRR 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 30.10 shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I2C mode.

The initial value of BRR is FFh.

BRR can be read from by the CPU, but it can be written to only when the TE and RE bits in SCR are 0.

Table 30.10 Relationship between N Setting in BRR and Bit Rate B

Mode	SEMR.ABCS Bit	BRR Setting	Error
Asynchronous, multi-processor transfer	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clock synchronous, simple SPI		$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface		$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$
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 the table below.
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 standard.

Table 30.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I2C	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 30.12 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

Table 30.13 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 30.14 lists examples of N settings in BRR in normal asynchronous mode. Table 30.15 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 30.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 30.20. Examples of BRR (N) settings in simple I^2C mode are listed in Table 30.22. 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 30.6.4, Receive Data Sampling Timing and Reception Margin. Table 30.16 and Table 30.19 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 30.14.

Table 30.14 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		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	–0.35
300	2	129	0.16	2	162	–0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35
1200	1	129	0.16	1	162	–0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35
4800	0	129	0.16	0	162	–0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35
31250	0	19	0.00	0	24	0.00	0	29	0
38400	0	15	1.73	0	19	1.73	0	23	1.73

Note: This is an example when the ABCS bit is 0.
When the ABCS bit is set to 1, the bit rate doubles.

Table 30.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0

Note: When the ABCS bit in SEMR is set to 1, the bit rate is two times.

Table 30.16 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

Table 30.17 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

Table 30.18 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

	Operating Frequency PCLK (MHz)											
	8		10		16		20		25		30	
Bit Rate (bps)	n	N	n	N	n	N	n	N	n	N	n	N
110												
250	3	124	—	—	3	249						
500	2	249	—	—	3	124	—	—			3	233
1 k	2	124	—	—	2	249	—	—	3	97	3	116
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187
25 k	0	79	0	99	0	159	0	199	0	249	1	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14
1 M	0	1			0	3	0	4	—	—	—	—
2 M	0	0*1	—	—	0	1	—	—	—	—	—	—
2.5 M			0	0*1			0	1	—	—	0	2
4 M					0	0*1	—	—	—	—	—	—
5 M							0	0*1	—	—	—	—
6.25 M									0	0*1	—	—
7.5 M											0	0*1

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is impossible. 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 $\frac{8}{9}$ times the bit rate.

Table 30.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	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

Table 30.20 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
	10.7136	0	1	25
	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

Table 30.21 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

Table 30.22 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		
	n	N	Error (%)
10 k	1	23	–2.3
25 k	1	9	–6.3
50 k	1	4	–6.3
100 k	1	2	–21.9
250 k	0	3	–6.3
350 k	0	2	–10.7

Table 30.23 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	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	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		
	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
25 k	1	7	17.92/20.48	1	9	18.66/21.33
50 k	1	3	8.96/10.24	1	4	9.33/10.66
100 k	1	1	4.48/5.12	1	2	5.60/6.40
250 k	0	3	2.24/2.56	0	3	1.86/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60

30.2.10 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 0008 A007h, SCI1.SEMR 0008 A027h, SCI2.SEMR 0008 A047h, SCI3.SEMR 0008 A067h, SCI4.SEMR 0008 A087h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI7.SEMR 0008 A0E7h, SCI8.SEMR 0008 A107h, SCI9.SEMR 0008 A127h, SCI10.SEMR 0008 A147h, SCI11.SEMR 0008 A167h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	—	NFEN	ABCS	—	—	—	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 clock cycles output from the TMR (valid only for SCI5, SCI6, and SCI12) The following table lists the correspondence between SCI channels and compare match outputs.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
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

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI6	Unit 1	TMO2, TMO3
SCI12	Unit 0	TMO0, TMO1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in 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 serial transfer base clock.

Figure 30.4 shows a setting example when the TMOn output (n = 0 to 3) of TMRn is selected.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b). An external clock input or internal TMR clock input can be selected.

Set the ACS0 bit to 0 in other than asynchronous mode.

These bits for the other SCI channels than SCI5, SCI6, and SCI12 are reserved. The write values to these bits for other than SCI5, SCI6, and SCI12 should be 0.

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.

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.

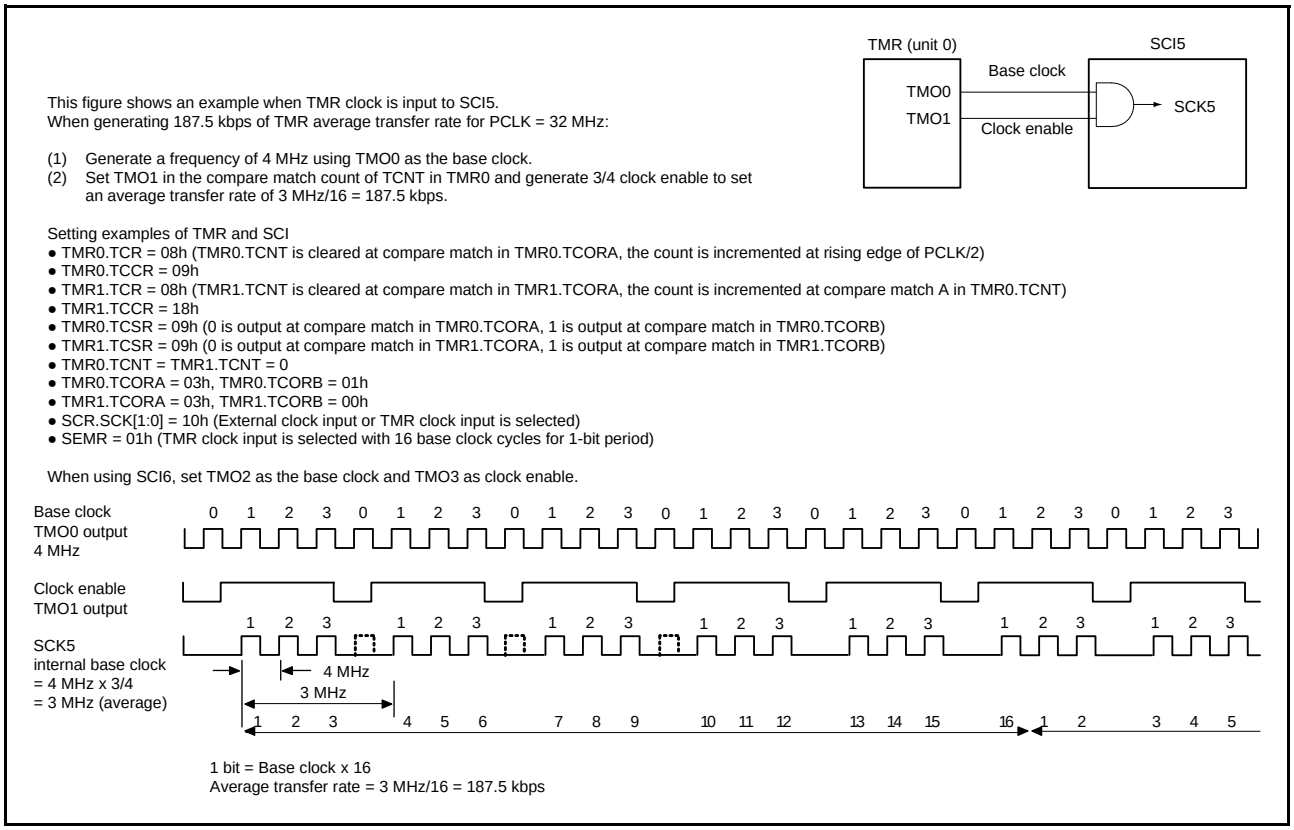


Figure 30.4 Example of Average Transfer Rate Setting When TMR Clock is Input

30.2.11 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 0008 A008h, SCI1.SNFR 0008 A028h, SCI2.SNFR 0008 A048h, SCI3.SNFR 0008 A068h, SCI4.SNFR 0008 A088h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI7.SNFR 0008 A0E8h, SCI8.SNFR 0008 A108h, SCI9.SNFR 0008 A128h, SCI10.SNFR 0008 A148h, SCI11.SNFR 0008 A168h, 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 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.

30.2.12 I²C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 0008 A009h, SCI1.SIMR1 0008 A029h, SCI2.SIMR1 0008 A049h, SCI3.SIMR1 0008 A069h, SCI4.SIMR1 0008 A089h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI7.SIMR1 0008 A0E9h, SCI8.SIMR1 0008 A109h, SCI9.SIMR1 0008 A129h, SCI10.SIMR1 0008 A149h, SCI11.SIMR1 0008 A169h, 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 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 SMIF bit in SCMR, 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 SMR.CKS[1:0] 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.

30.2.13 I²C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 0008 A00Ah, SCI1.SIMR2 0008 A02Ah, SCI2.SIMR2 0008 A04Ah, SCI3.SIMR2 0008 A06Ah, SCI4.SIMR2 0008 A08Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, SCI7.SIMR2 0008 A0EAh, SCI8.SIMR2 0008 A10Ah, SCI9.SIMR2 0008 A12Ah, SCI10.SIMR2 0008 A14Ah, SCI11.SIMR2 0008 A16Ah, 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 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.

30.2.14 I²C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 0008 A00Bh, SCI1.SIMR3 0008 A02Bh, SCI2.SIMR3 0008 A04Bh, SCI3.SIMR3 0008 A06Bh, SCI4.SIMR3 0008 A08Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI7.SIMR3 0008 A0EBh, SCI8.SIMR3 0008 A10Bh, SCI9.SIMR3 0008 A12Bh, SCI10.SIMR3 0008 A14Bh, SCI11.SIMR3 0008 A16Bh, 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. Only generate a start condition after checking the bus state and confirming that it is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that it is busy.
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.

30.2.15 I²C Status Register (SISR)

Address(es): SCI0.SISR 0008 A00Ch, SCI1.SISR 0008 A02Ch, SCI2.SISR 0008 A04Ch, SCI3.SISR 0008 A06Ch, SCI4.SISR 0008 A08Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI7.SISR 0008 A0ECh, SCI8.SISR 0008 A10Ch, SCI9.SISR 0008 A12Ch, SCI10.SISR 0008 A14Ch, SCI11.SISR 0008 A16Ch, 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.

30.2.16 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 0008 A00Dh, SCI1.SPMR 0008 A02Dh, SCI2.SPMR 0008 A04Dh, SCI3.SPMR 0008 A06Dh, SCI4.SPMR 0008 A08Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, SCI7.SPMR 0008 A0EDh, SCI8.SPMR 0008 A10Dh, SCI9.SPMR 0008 A12Dh, SCI10.SPMR 0008 A14Dh, SCI11.SPMR 0008 A16Dh, 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 TXDn pin and reception is through the RXDn pin (master mode). 1: Reception is through the TXDn pin and transmission is through the RXDn 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 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. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn 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. See Figure 30.56 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. See Figure 30.56 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

30.2.17 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 enters the following states:

- The extended serial mode control section is initialized

Table 30.24 Settings of the ESME Bit and Guaranteed Operation by Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	○*1	×	×
1	○	○	○

○: Guarantee of operation is necessary. ×: Guarantee of operation is not necessary.
Note 1. Operation is only possible with PCLK selected.

30.2.18 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

30.2.19 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 PCF1DR. 0 1: Selects comparison with the value in SCF1DR. 1 0: Selects comparison with the values in PCF1DR and SCF1DR. 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

30.2.20 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 SCI base clock*1 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	b5 b4 0 0: SCI base clock 0 1: SCI base clock frequency divided by 2 1 0: SCI base clock frequency divided by 4 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 SCI base clock 0 1: Rising edge of the 10th cycle of SCI base clock 1 0: Rising edge of the 12th cycle of SCI base clock 1 1: Rising edge of the 14th cycle of SCI base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of SCI base clock 0 1: Rising edge of the 5th cycle of SCI base clock 1 0: Rising edge of the 6th cycle of SCI base clock 1 1: Rising edge of the 7th cycle of SCI base clock	R/W

Note: The period of the SCI 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 SCI base clock, set the SCI12.SCR.TE bit to 1.

30.2.21 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.

30.2.22 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.

30.2.23 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

30.2.24 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 BFDCL bit in STCR	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 CF0MCL bit in STCR	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 CF1MCL bit in STCR	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the PIBDCL bit in STCR	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the BCDCL bit in STCR	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the AEDCL bit in STCR	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

30.2.25 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

30.2.26 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

CF0DR is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

30.2.27 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

30.2.28 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. Writing to this register from the CPU or DTC is not possible.

30.2.29 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.

30.2.30 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.

30.2.31 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
b6	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

30.2.32 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. Writing to this register from the CPU or DTC is not possible.

30.2.33 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

30.2.34 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 TPRE or TCNT is written to the reload register only or is written to both the reload register and the counter.

30.2.35 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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes one PCLK cycle to load a value from the reload register to the counter.

30.2.36 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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes one PCLK cycle to load a value from the reload register to the counter.

30.3 Operation in Asynchronous Mode

Figure 30.5 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.

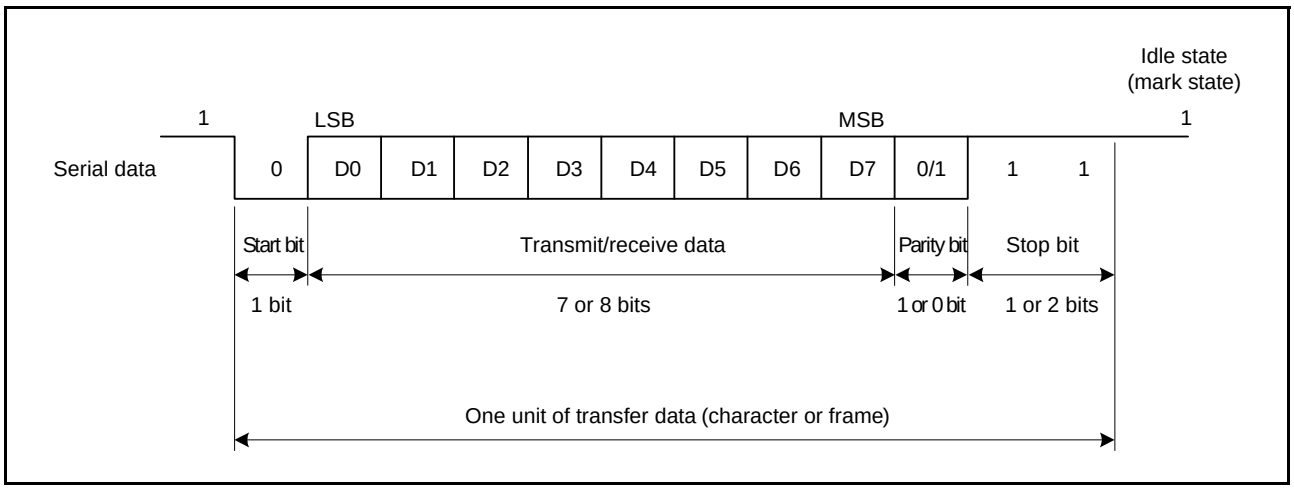


Figure 30.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

30.3.1 Serial Data Transfer Format

Table 30.25 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, refer to section 30.4, Multi-Processor Communications Function.

Table 30.25 Serial Transfer Formats (Asynchronous Mode)

SMR Setting				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

30.3.2 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 of each bit, as shown in Figure 30.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 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. This is an example when the ABCS bit in SEMR is 0. When 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.

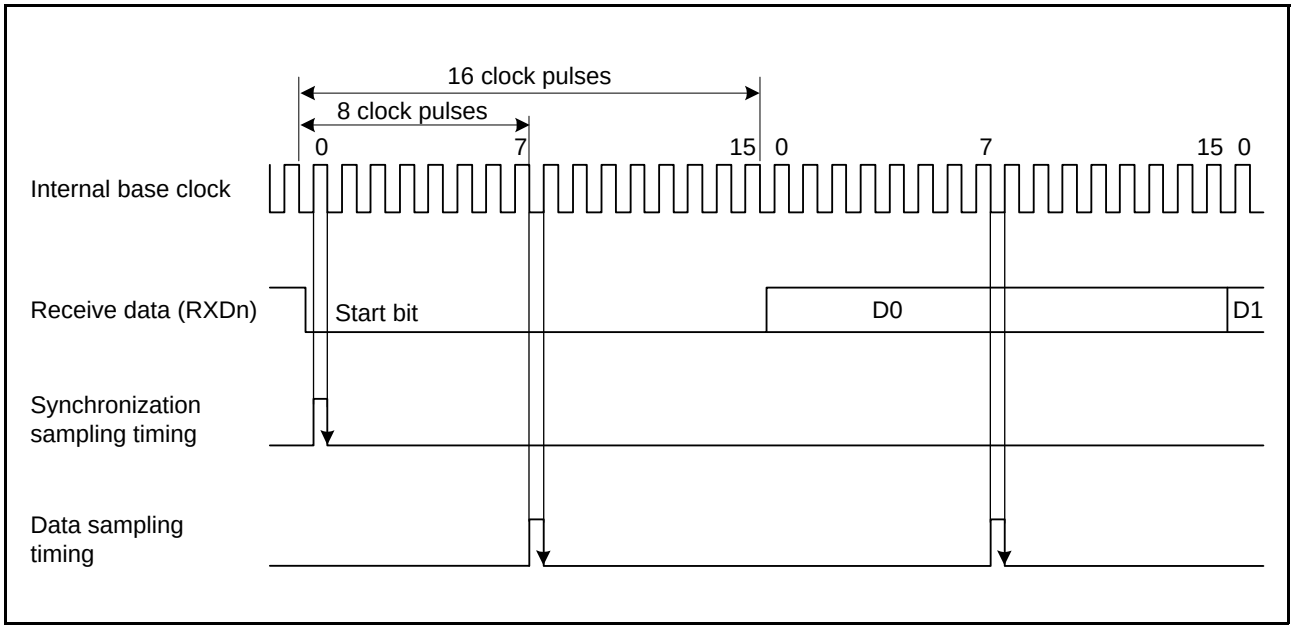


Figure 30.6 Receive Data Sampling Timing in Asynchronous Mode

30.3.3 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 CM bit in SMR and the CKE[1:0] bits in SCR.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 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 30.7.

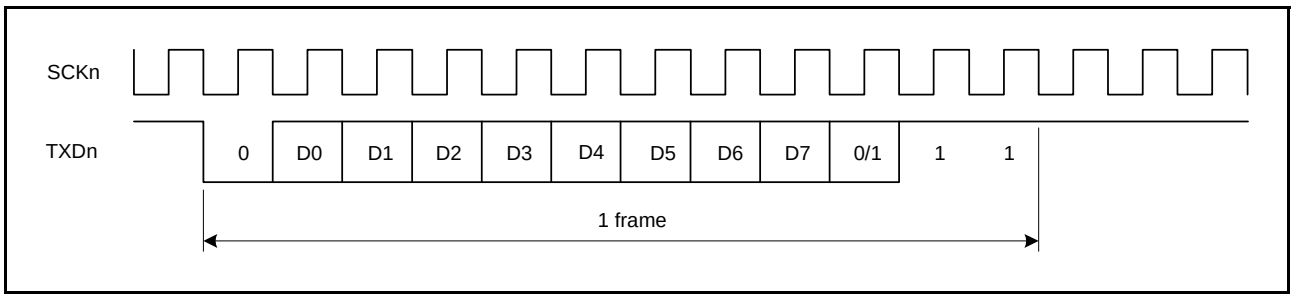


Figure 30.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

30.3.4 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]
Satisfaction of all conditions listed below

- The value of the RE bit in the SCR 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 are all 0

[Condition for high-level output]

- The conditions for low-level output have not been satisfied.
- Note that either one of CTS and RTS can be selected.

30.3.5 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to SCR and then continue through the procedure for SCI given in Figure 30.8. Whenever the operating mode or transfer format is changed, SCR 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, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

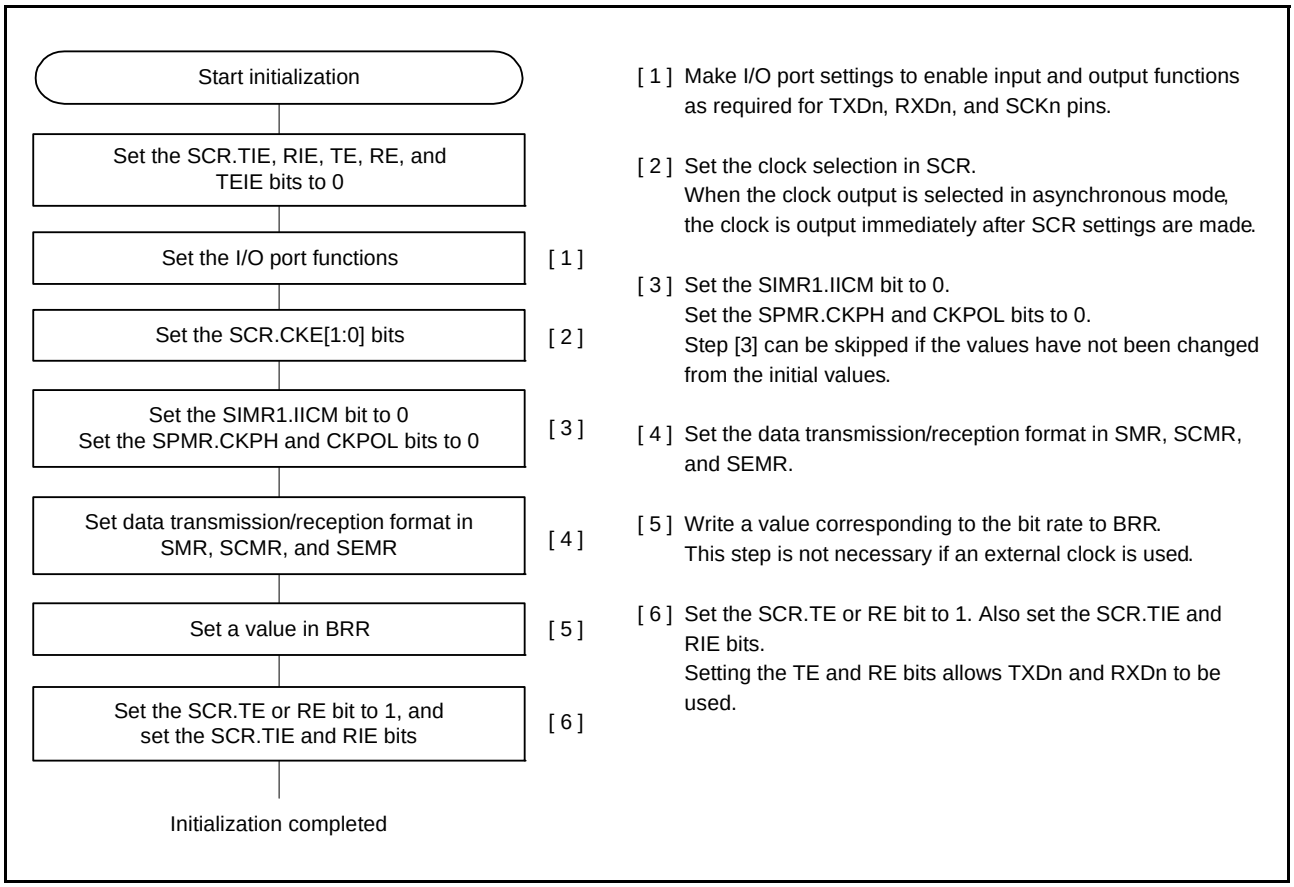


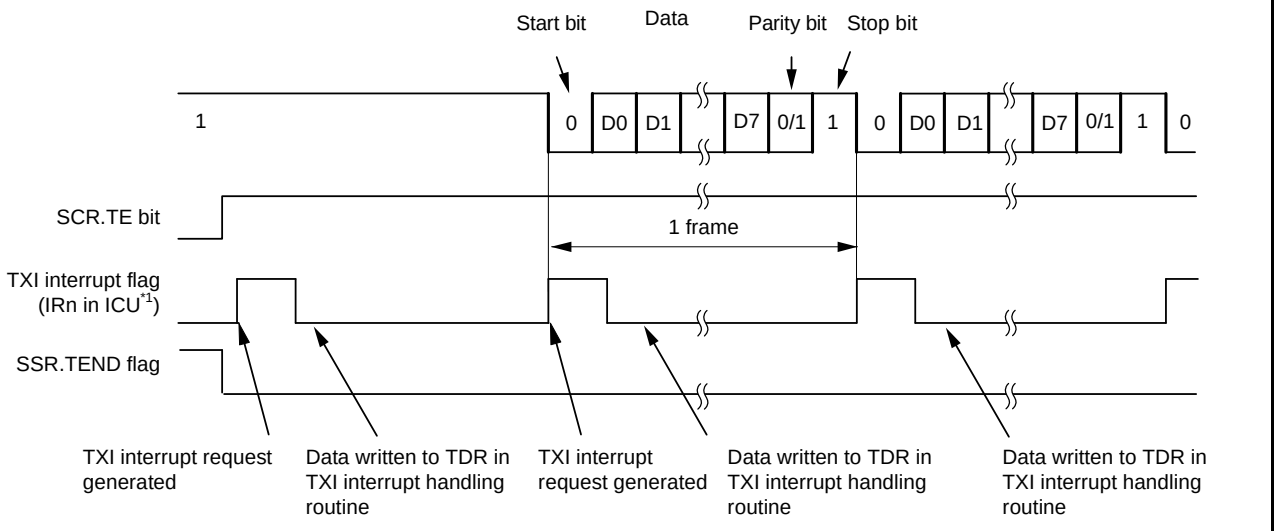
Figure 30.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

30.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 30.9 to Figure 30.11 show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

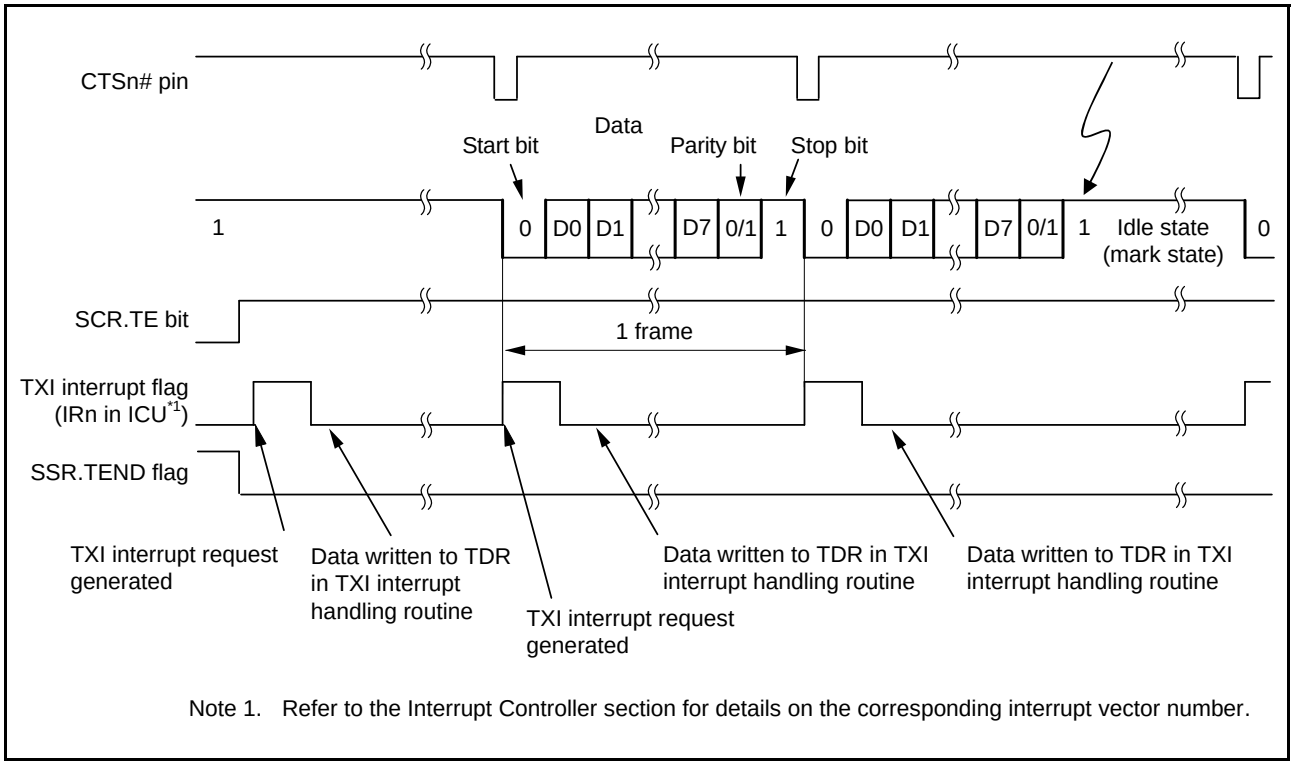
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from TDR to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR 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 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) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SPMR 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 TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE bit in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 30.12 shows a sample flowchart for serial transmission in asynchronous mode.



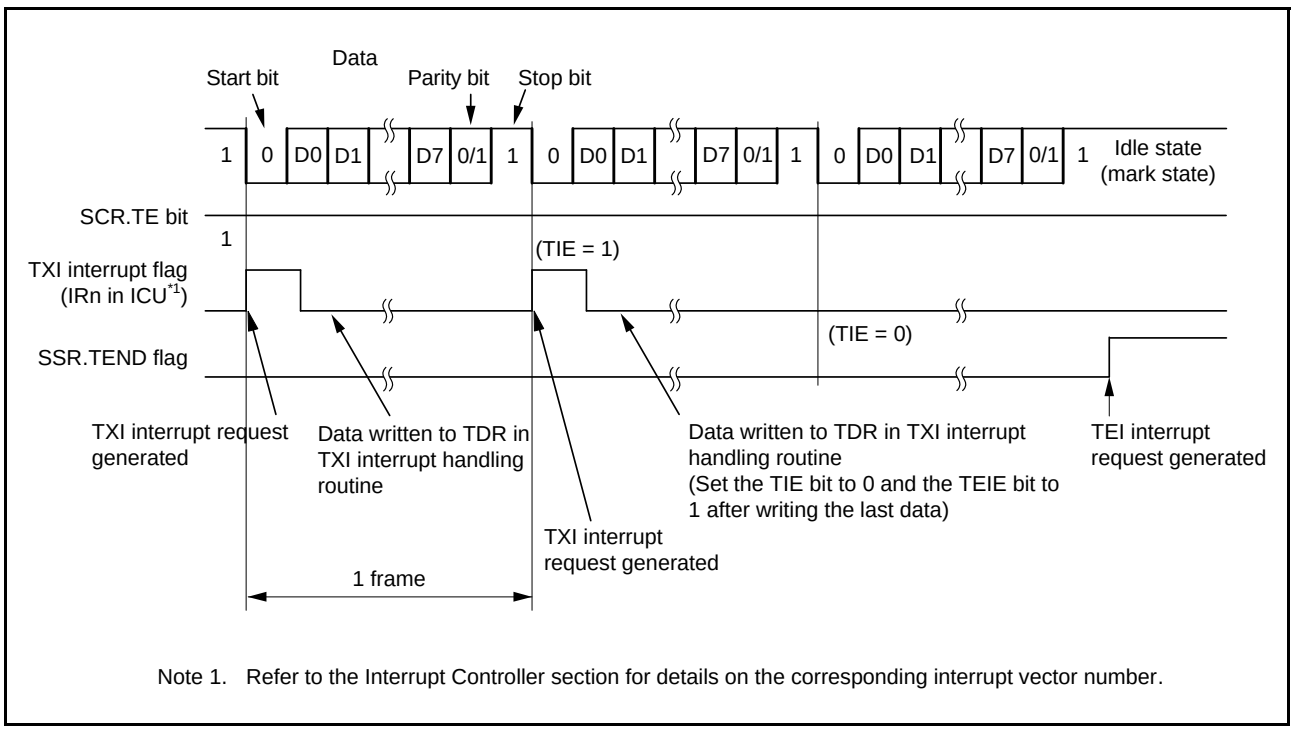
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.9 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)



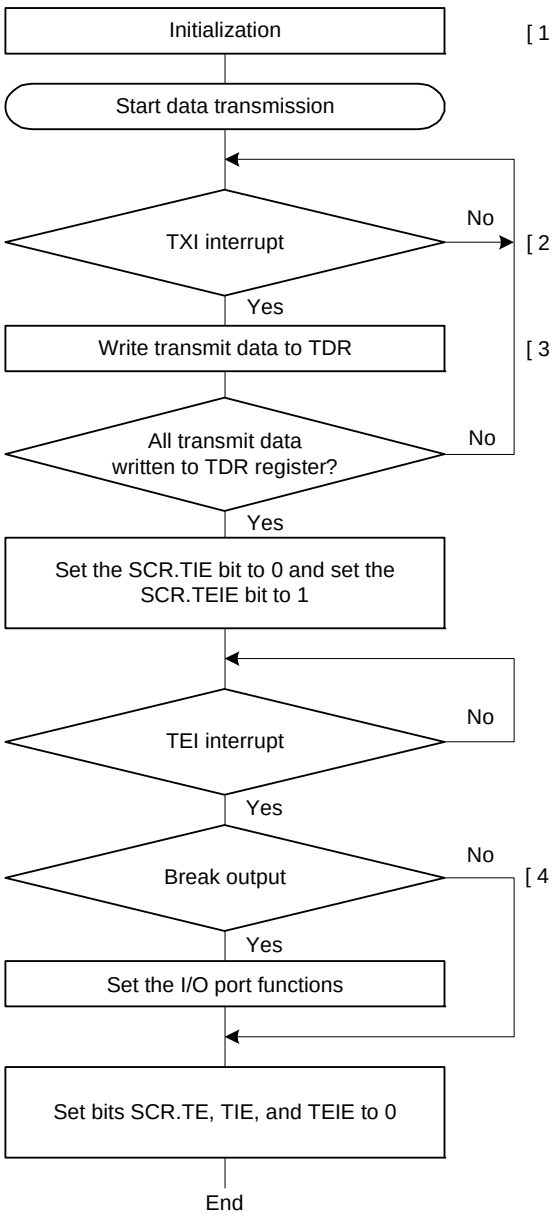
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.10 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 the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.11 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 TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the 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.

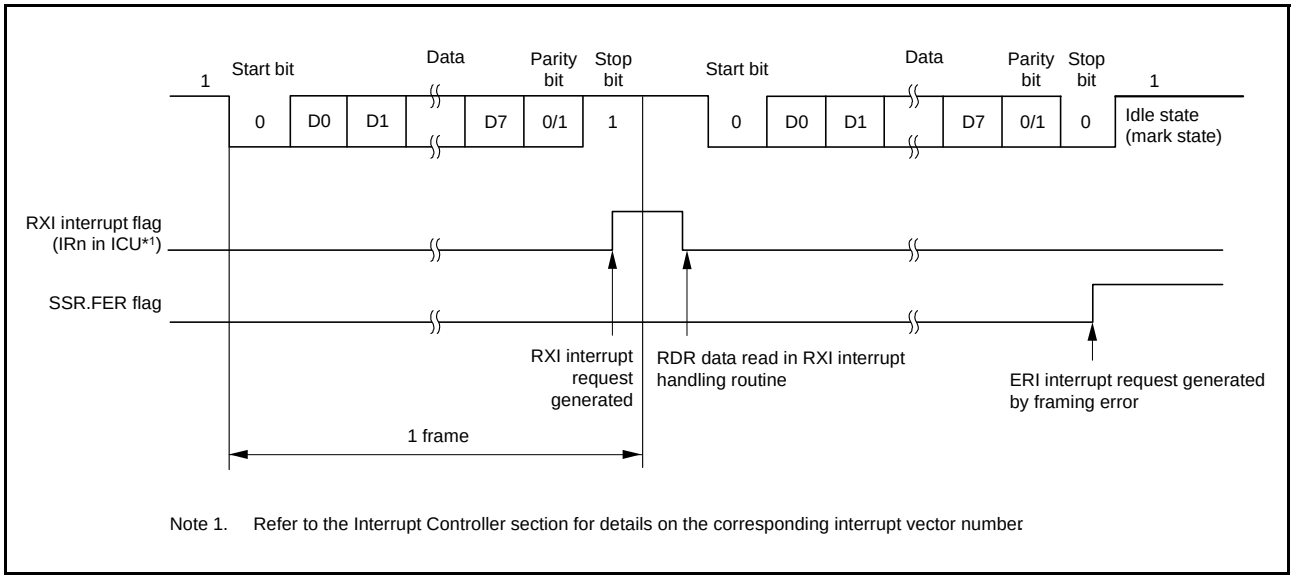
[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

Figure 30.12 Example of Serial Transmission Flowchart in Asynchronous Mode

30.3.7 Serial Data Reception (Asynchronous Mode)

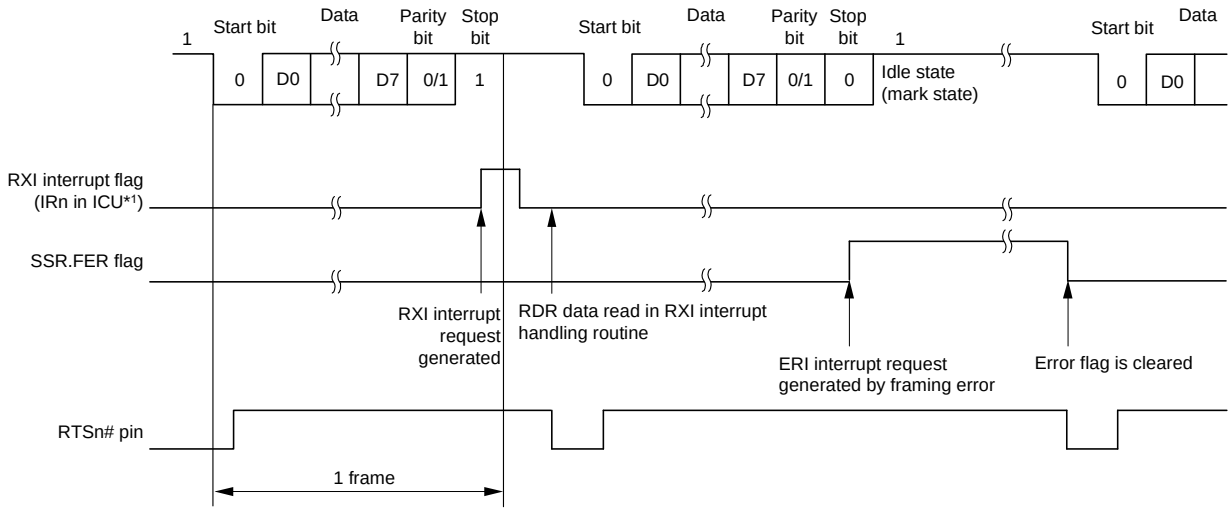
Figure 30.13 and Figure 30.14 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the RE bit in SCR 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 RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR 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 FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR 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 causes the RTSn# pin to output the low level.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number

Figure 30.13 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 the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.14 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 30.26 lists the states of the flags in the 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, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR 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 RDR.

Figure 30.15 and Figure 30.16 show samples of flowcharts for serial data reception.

Table 30.26 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**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

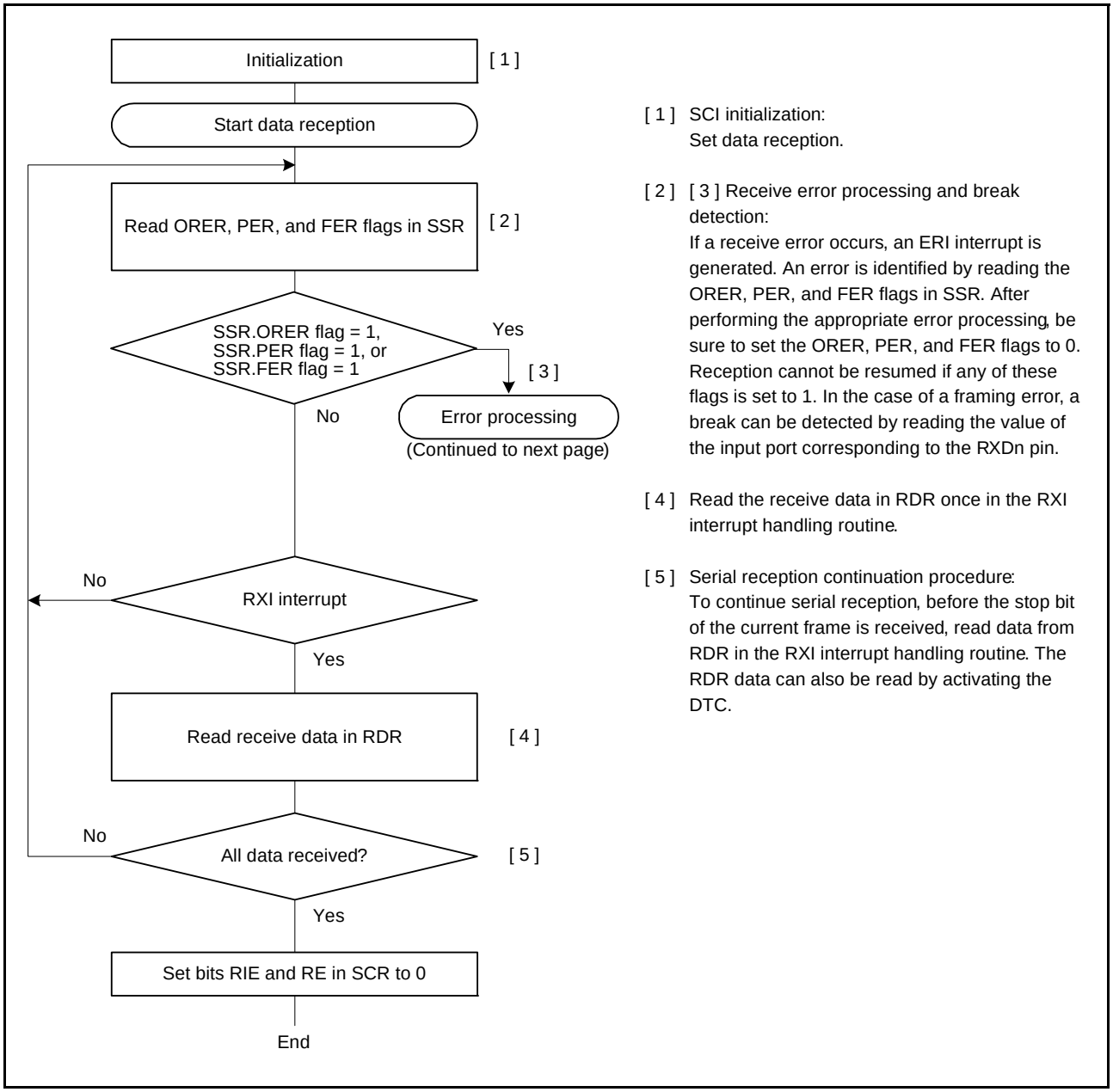


Figure 30.15 Example Flowchart of Serial Reception in Asynchronous Mode (1)

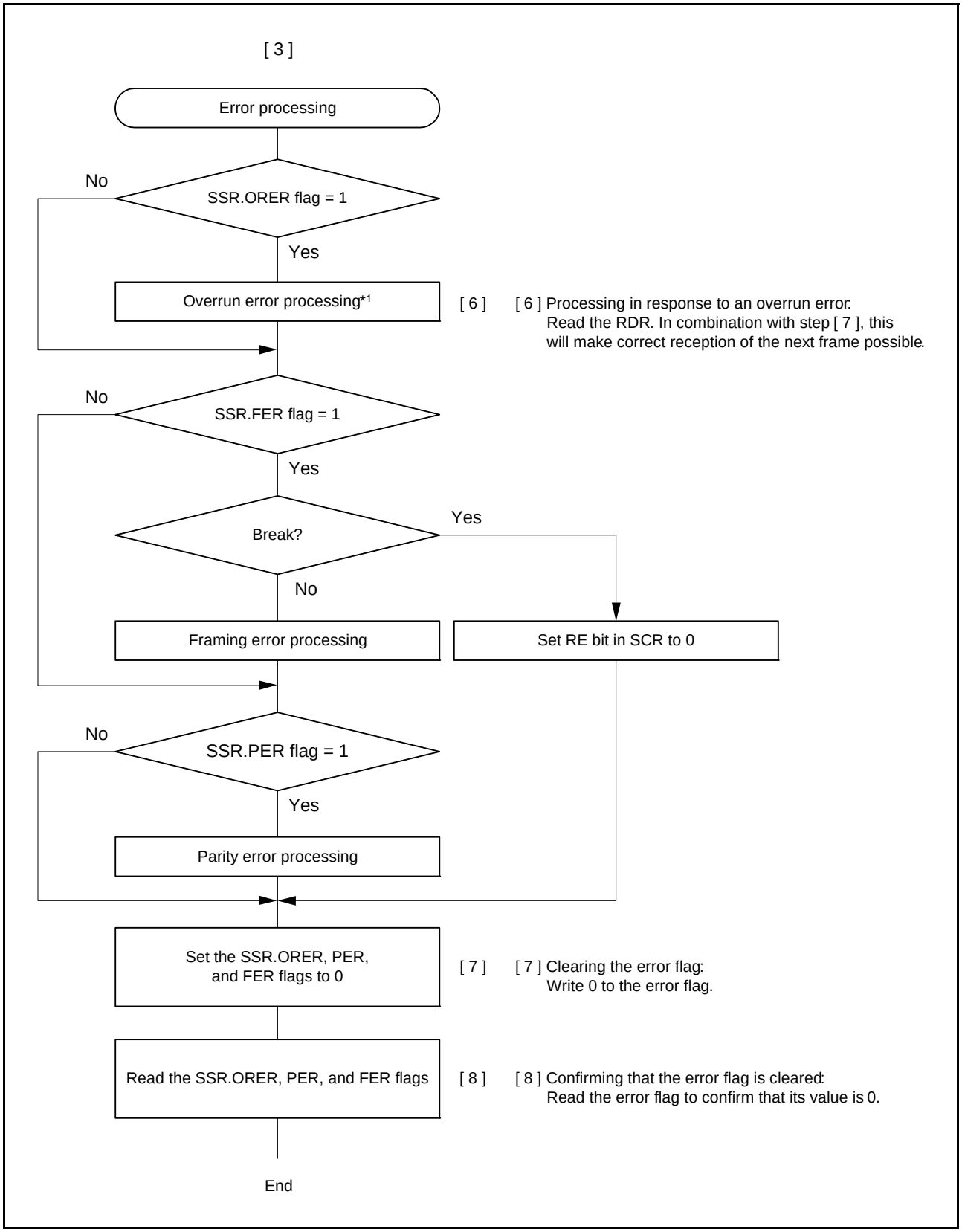


Figure 30.16 Example Flowchart of Serial Reception in Asynchronous Mode (2)

30.4 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 30.17 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 MPIE bit in SCR. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a receive error, and setting the respective status flags ORER and FER 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 MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR 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 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.

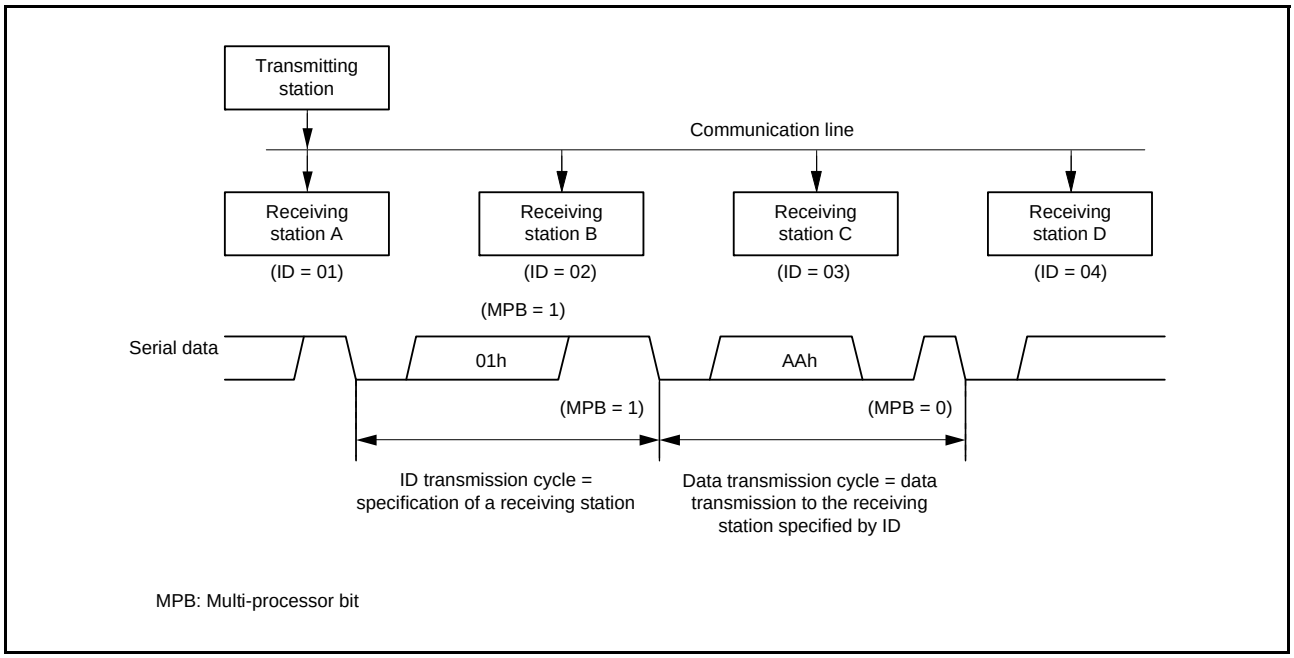


Figure 30.17 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

30.4.1 Multi-Processor Serial Data Transmission

Figure 30.18 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR 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.

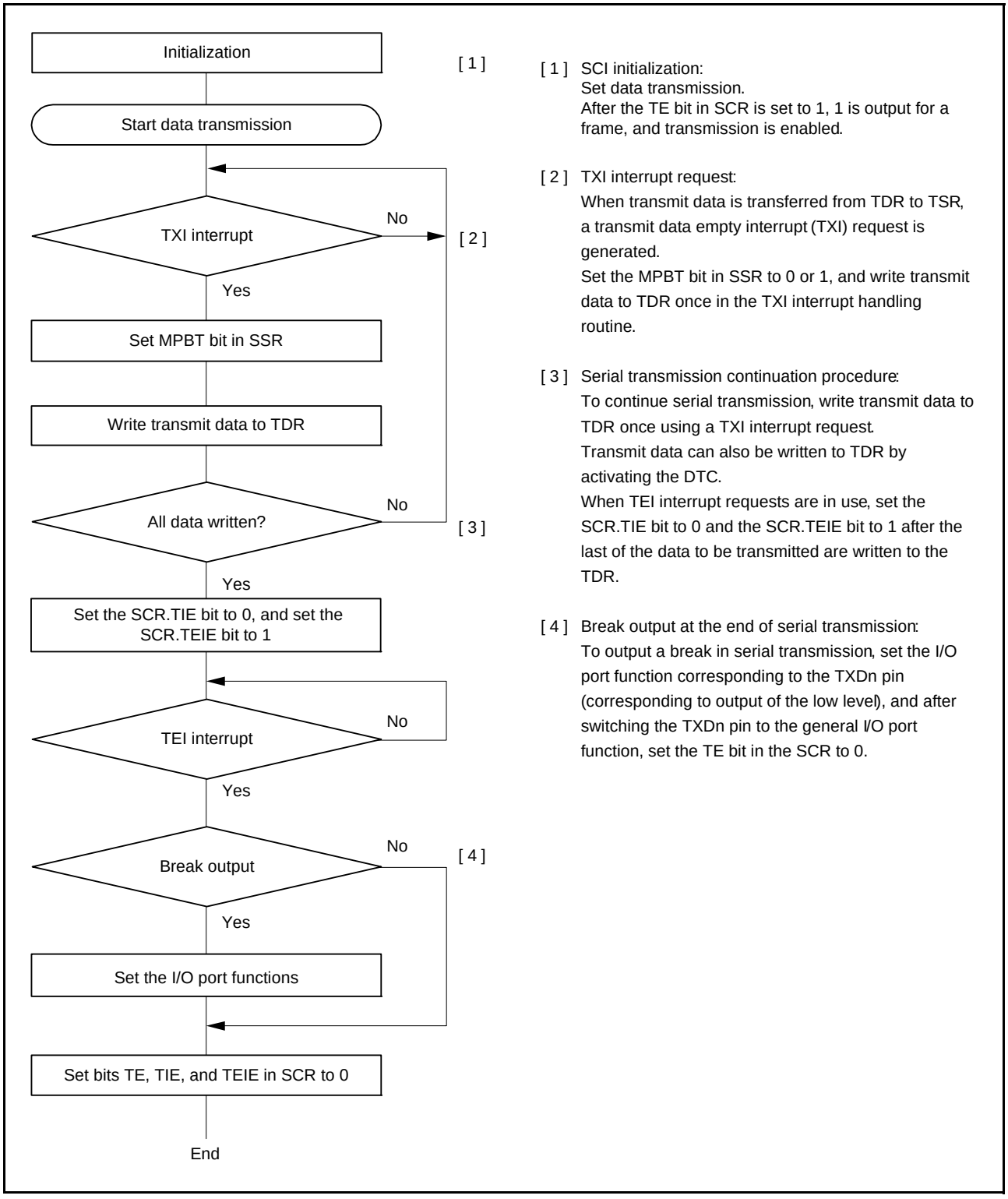


Figure 30.18 Example of Multi-Processor Serial Transmission Flowchart

30.4.2 Multi-Processor Serial Data Reception

Figure 30.20 and Figure 30.21 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR 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. During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 30.19 is the example of operation for reception.

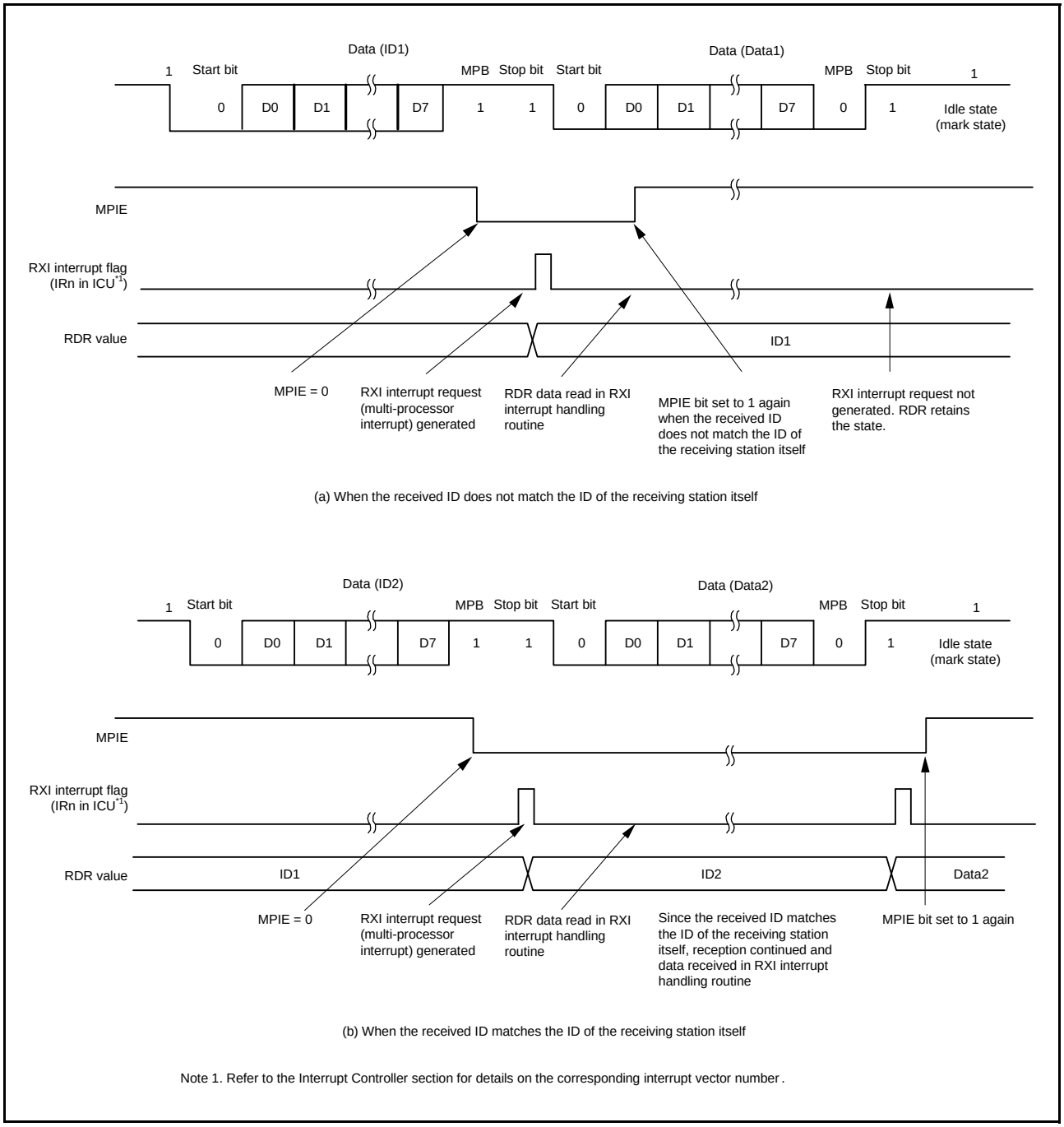


Figure 30.19 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

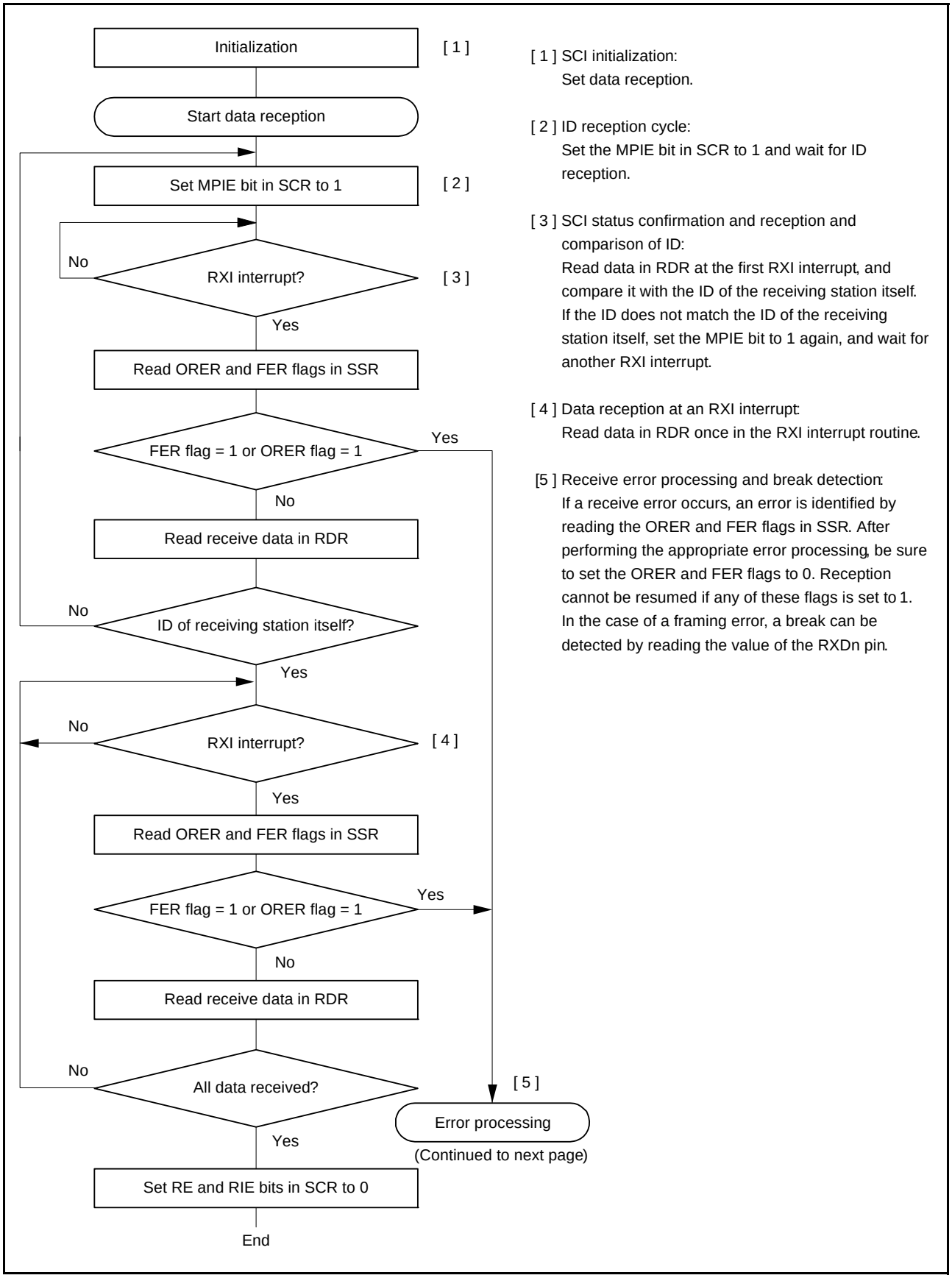


Figure 30.20 Example of Multi-Processor Serial Reception Flowchart (1)

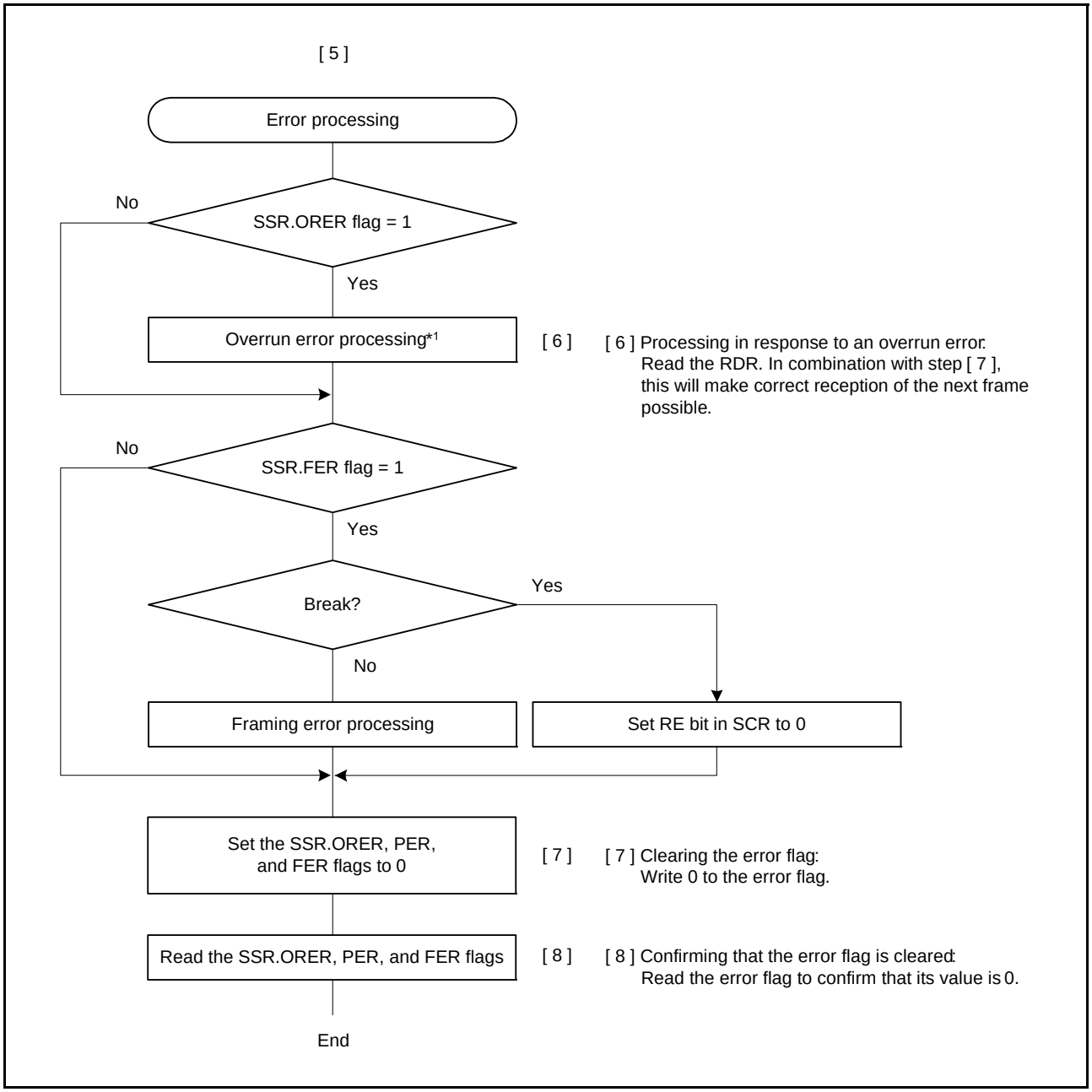


Figure 30.21 Example of Multi-Processor Serial Reception Flowchart (2)

30.5 Operation in Clock Synchronous Mode

Figure 30.22 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.

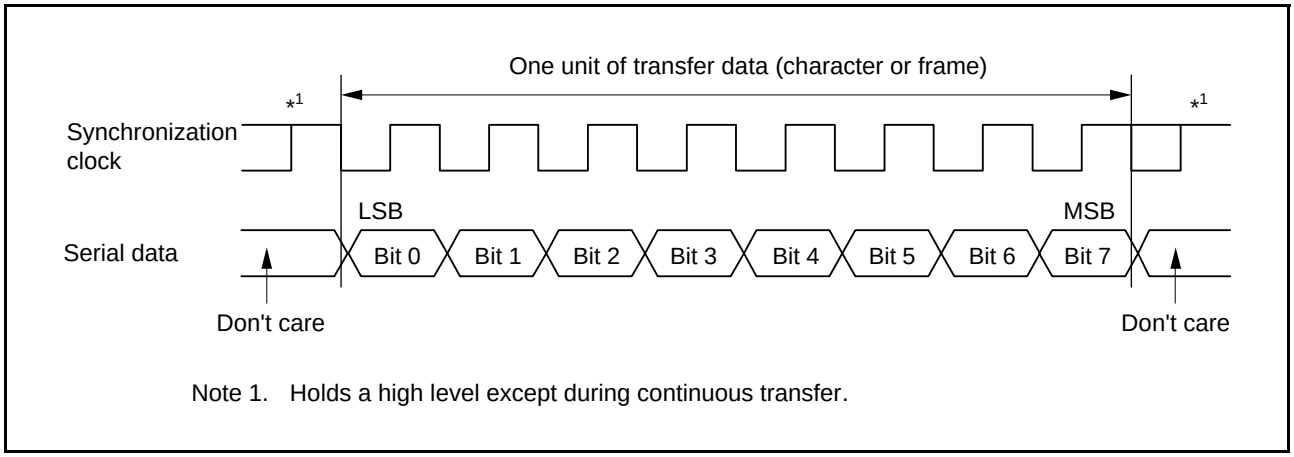


Figure 30.22 Data Format in Clock Synchronous Serial Communications (LSB First)

30.5.1 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.

30.5.2 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.
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]
Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- Neither transmission nor reception is in progress
- There are no received data yet to be read (when the SCR.RE bit is 1)
- Transmit data has been written (when the SCR.TE bit is 1)
- ORER flag in SSR is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

30.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR and then continue through the procedure for SCI given in Figure 30.23. Whenever the operating mode or transfer format is changed, the SCR 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 SSR nor RDR.
Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

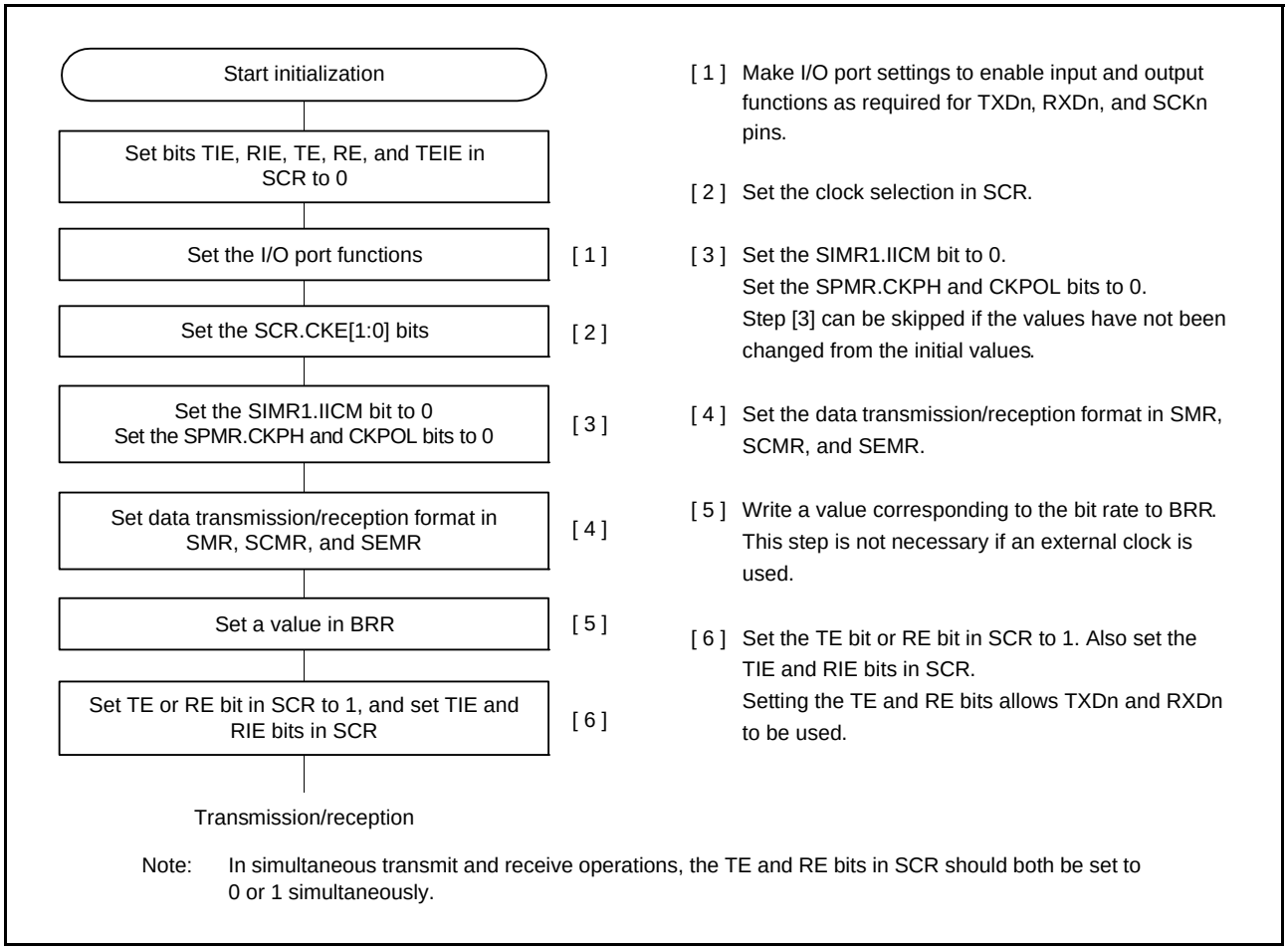


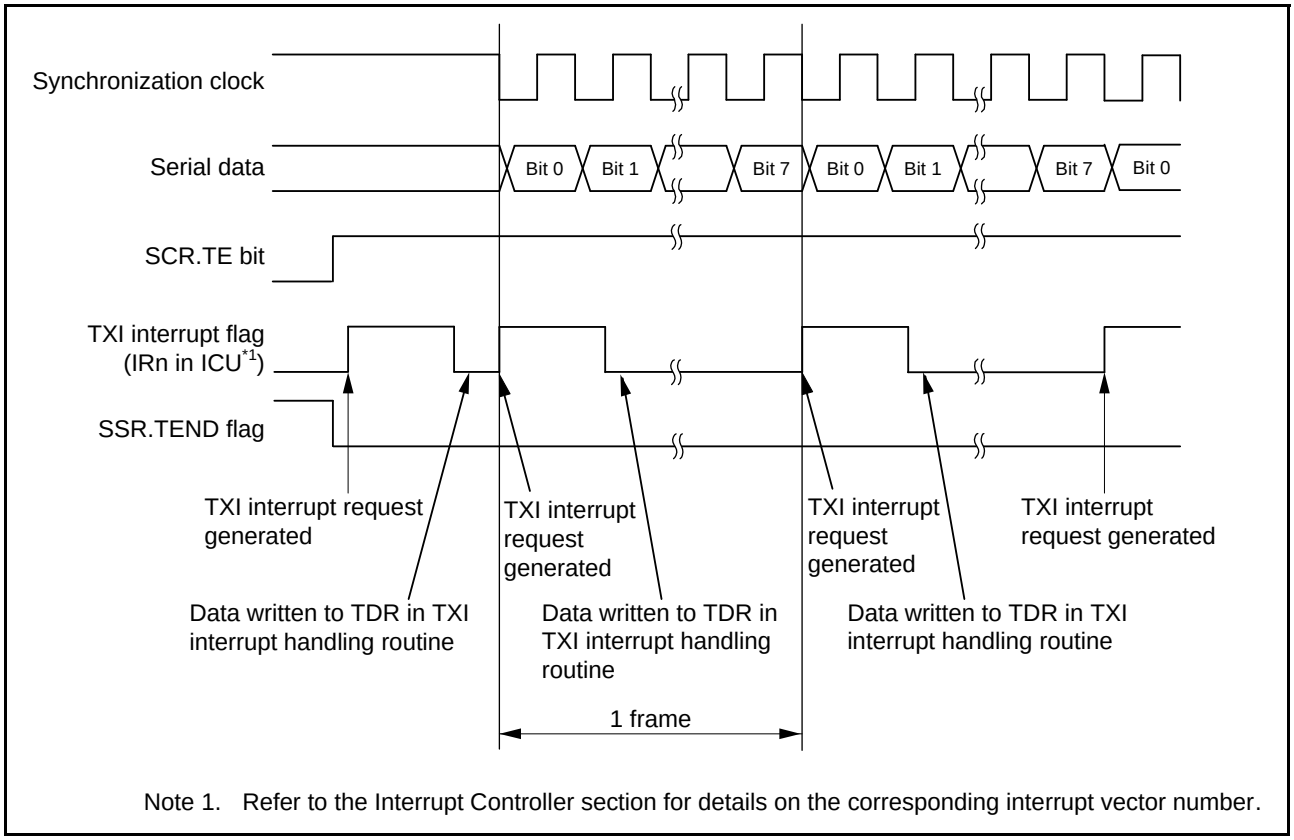
Figure 30.23 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

30.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 30.23, Figure 30.24, and Figure 30.25 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

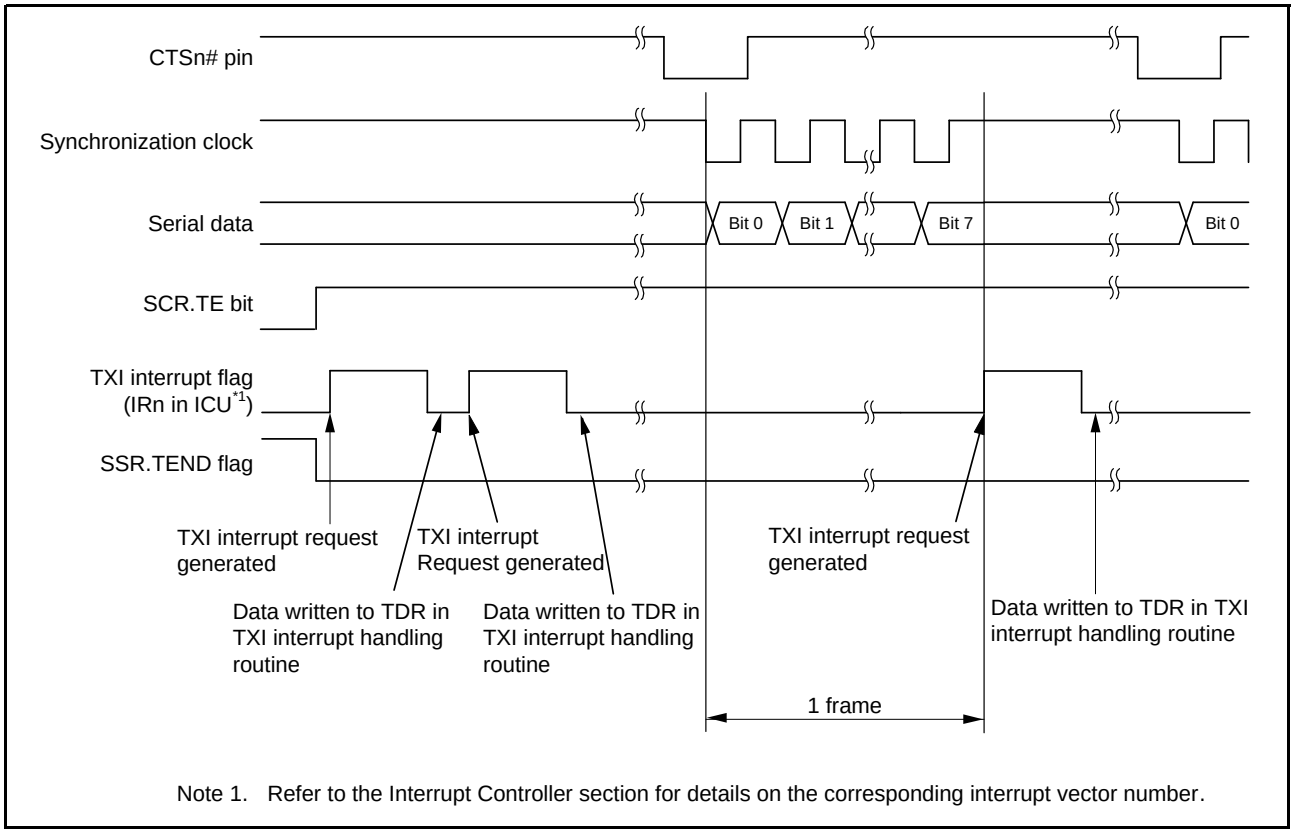
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, 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 TDR 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 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 CTSE bit in SPMR is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 30.27 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the RE bit in SCR to 0 does not clear the receive error flags.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.24 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 the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 30.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission

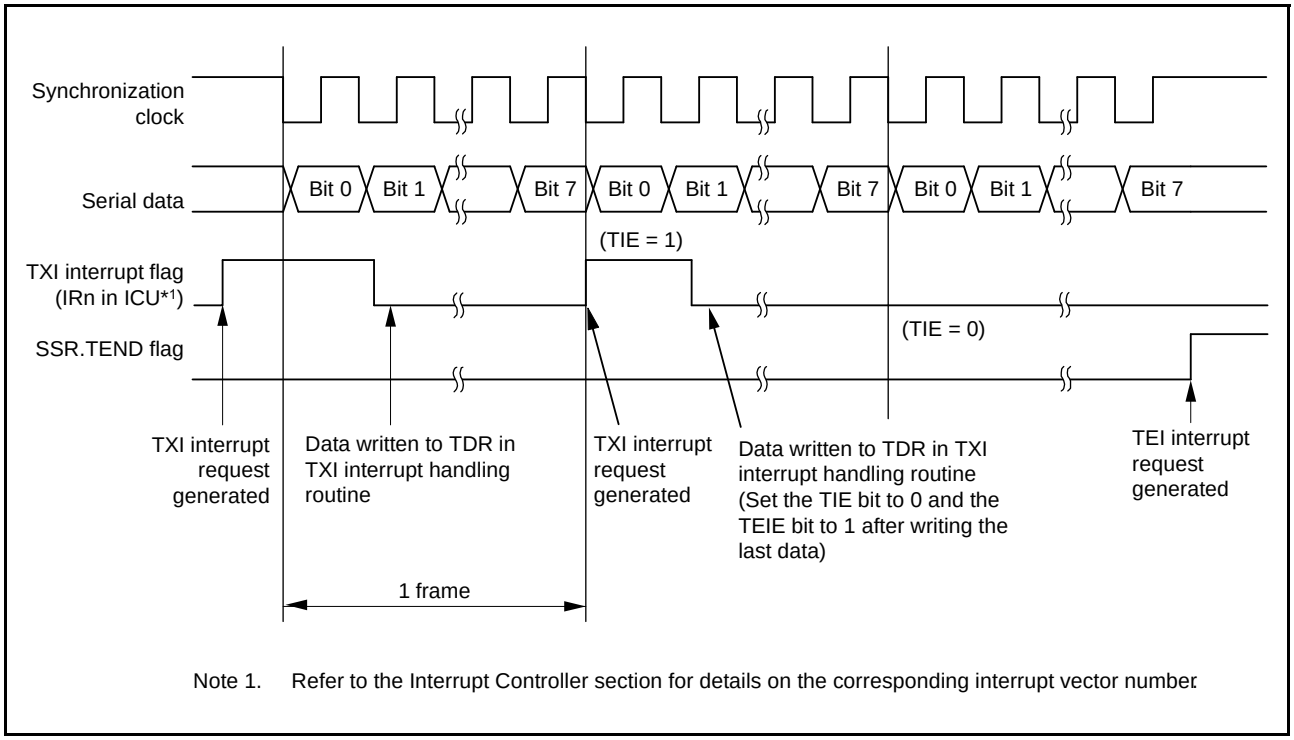
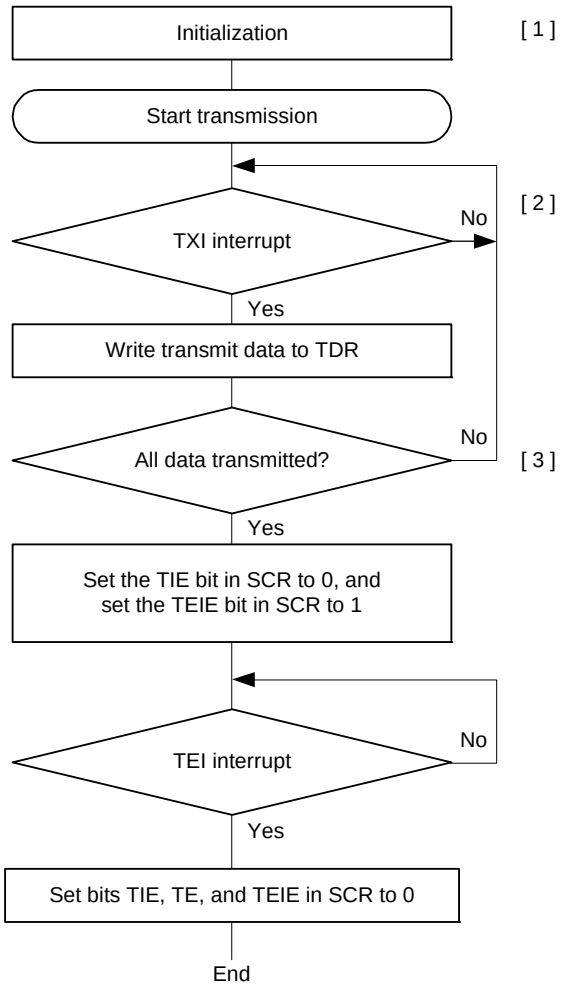


Figure 30.26 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to TDR 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.

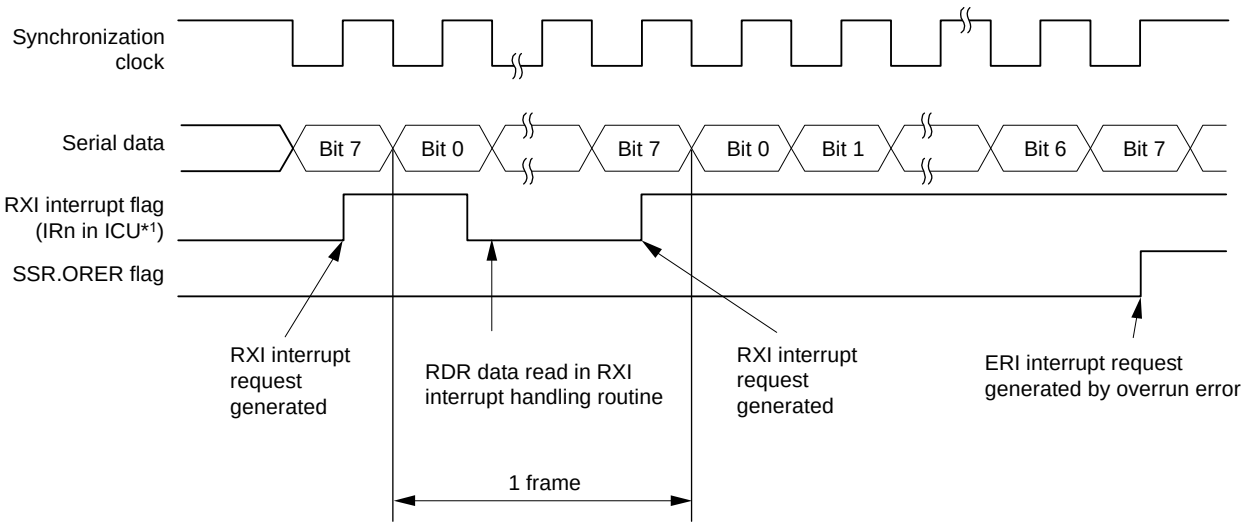
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 30.27 Example Flowchart of Serial Transmission in Clock Synchronous Mode

30.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 30.28 and Figure 30.29 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 RE bit in SCR 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 RSR.
3. If an overrun error occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR 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 RDR causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to the Interrupt Controller section for the corresponding interrupt vector number.

Figure 30.28 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

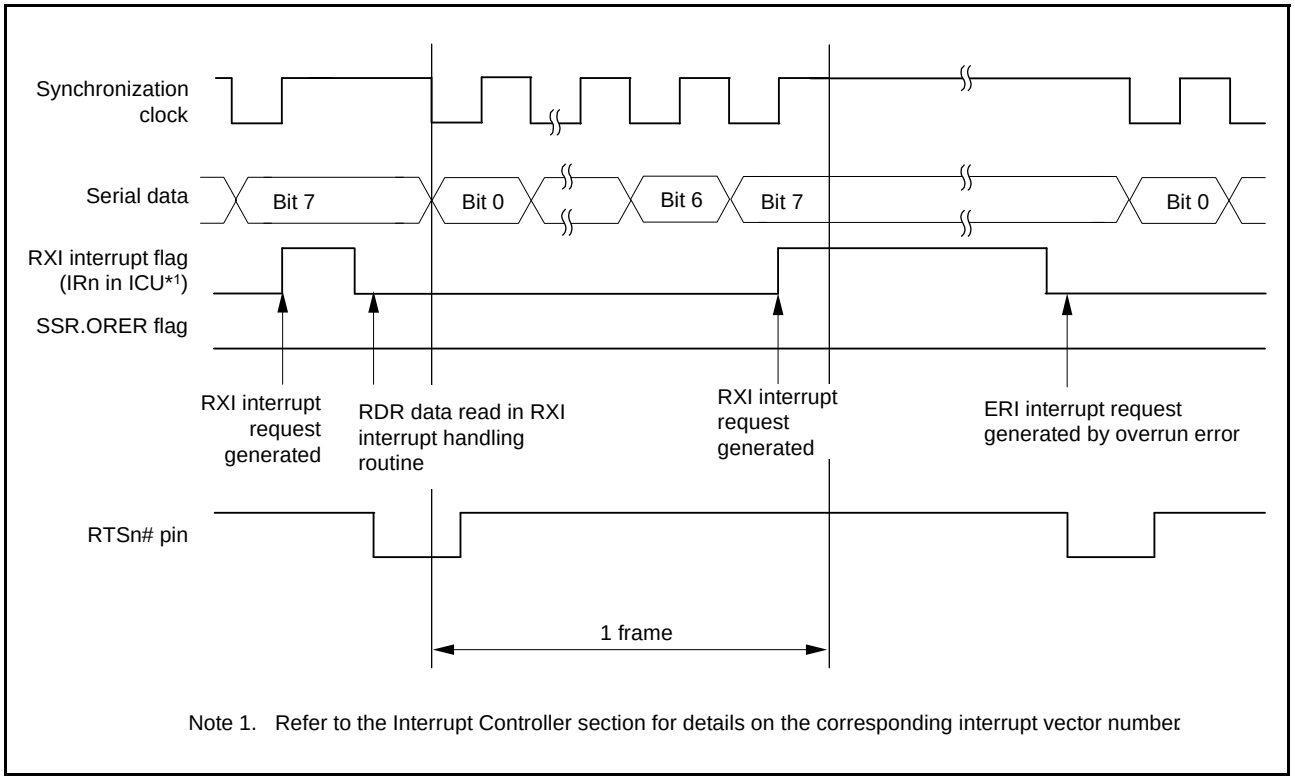
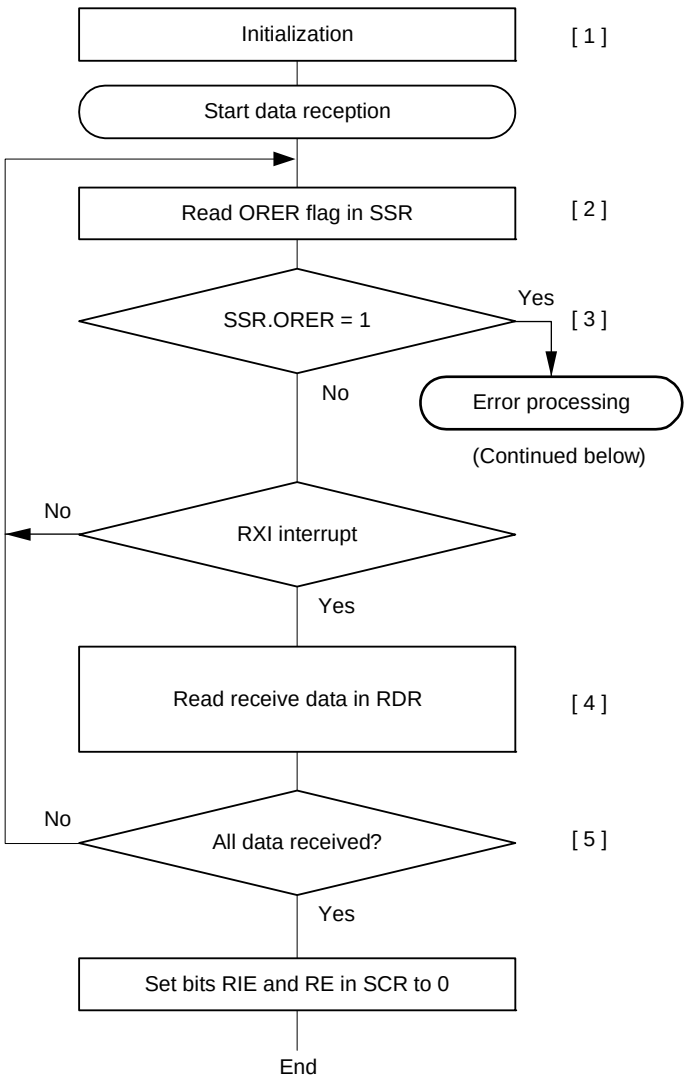
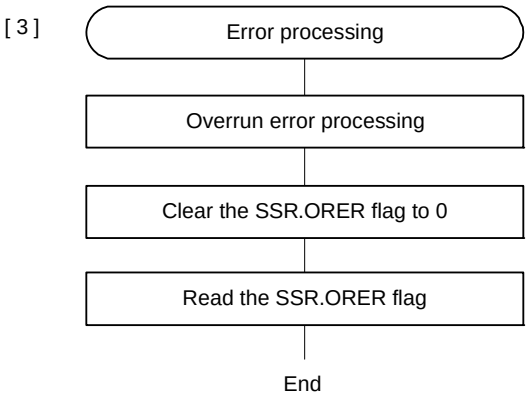


Figure 30.29 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read RDR because received data which has not yet been read may be left in RDR.

Figure 30.30 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 ORER flag in SSR, 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 RDR 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 RDR. 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. 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 30.30 Example Flowchart of Serial Reception in Clock Synchronous Mode

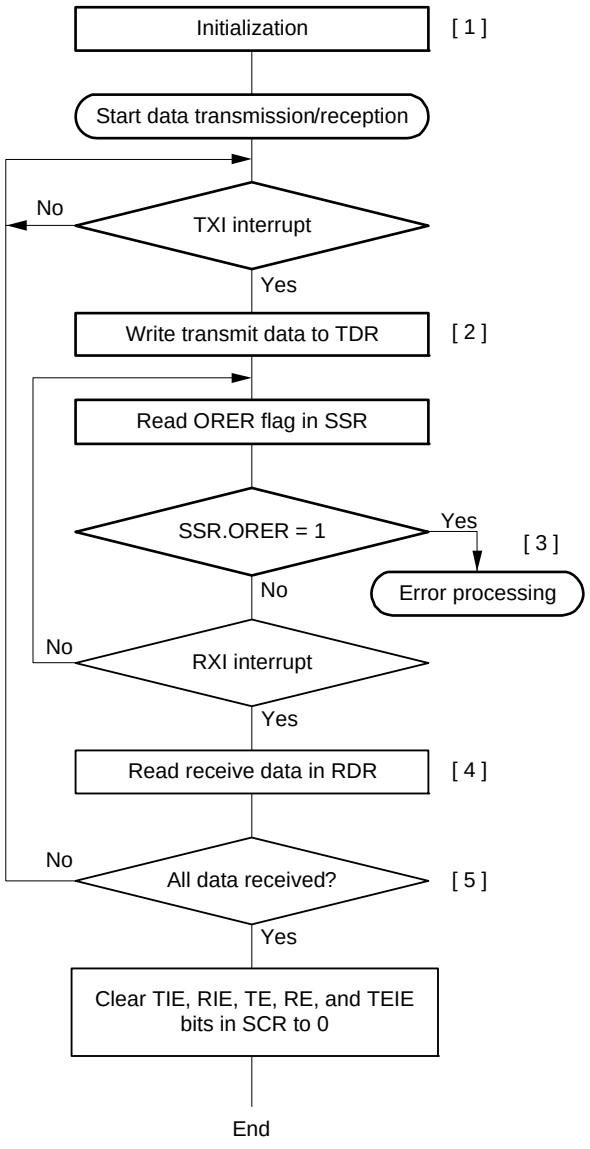
30.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 30.31 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 TEND flag in SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR 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 SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR 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 TDR once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, 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 RDR 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 RDR by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to TDR by the TXI interrupt.
Transmit data can also be written to TDR 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 SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 30.31 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

30.6 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.

30.6.1 Sample Connection

Figure 30.32 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 SCR 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.

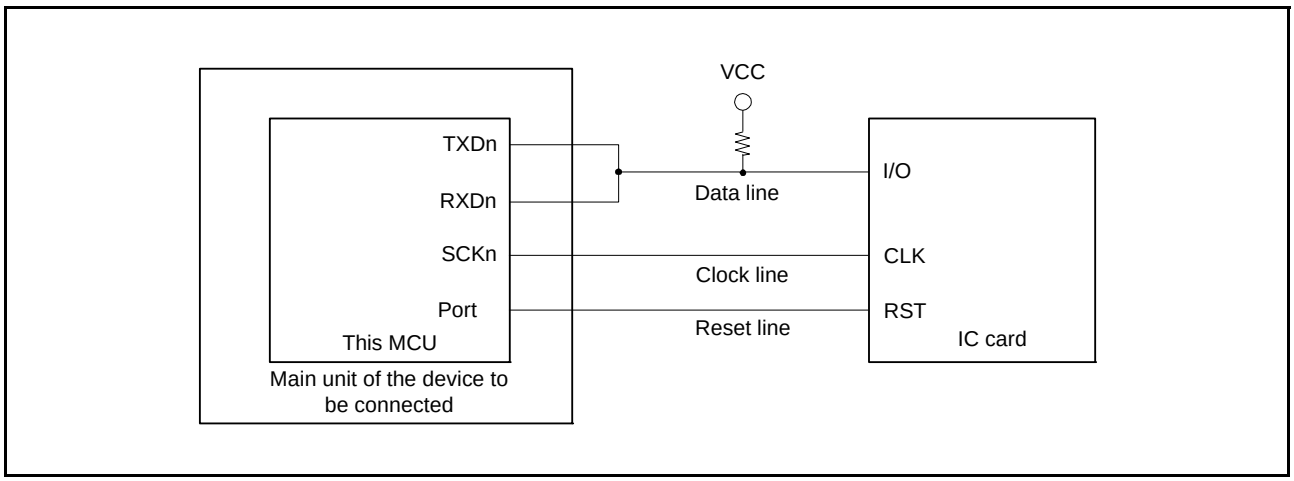


Figure 30.32 Sample Connection with a Smart Card (IC Card)

30.6.2 Data Format (Except in Block Transfer Mode)

Figure 30.33 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.

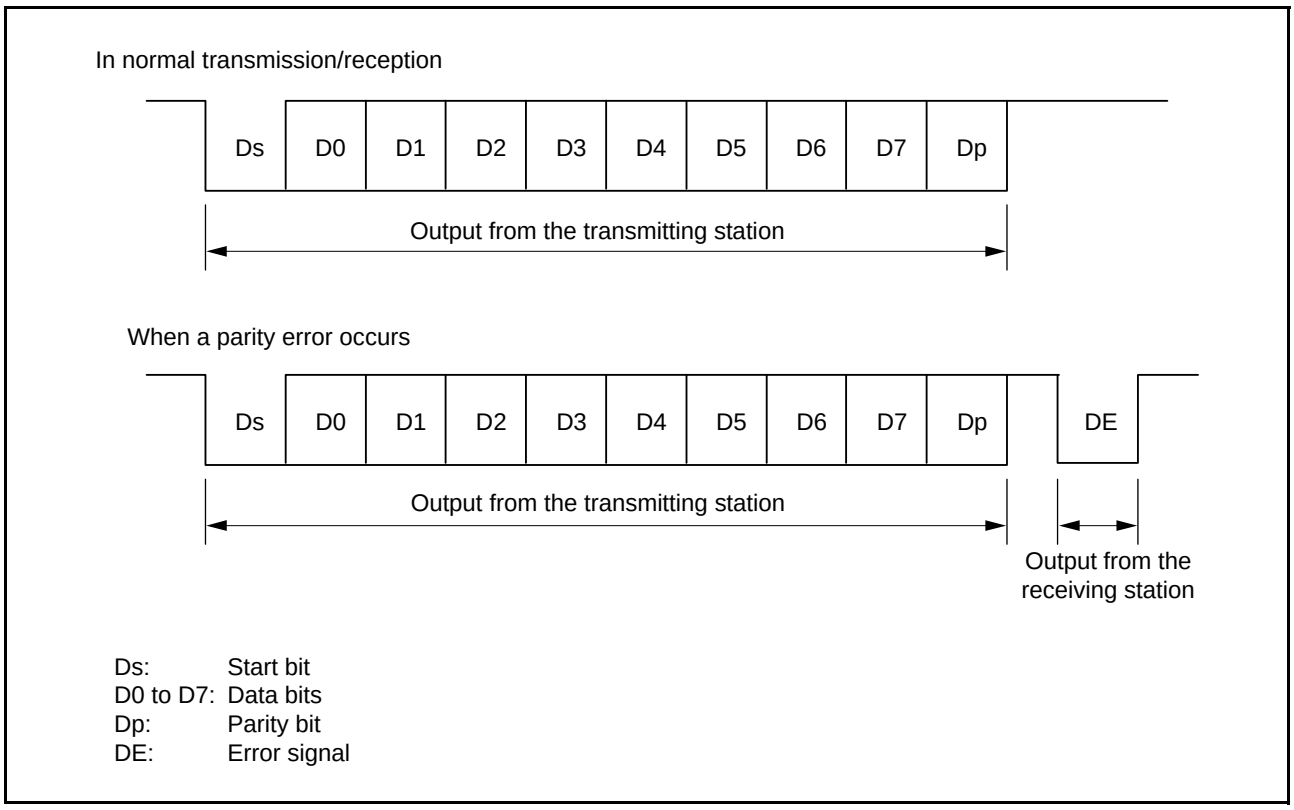


Figure 30.33 Data Formats in Smart Card Interface Mode

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 30.34. 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 SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.

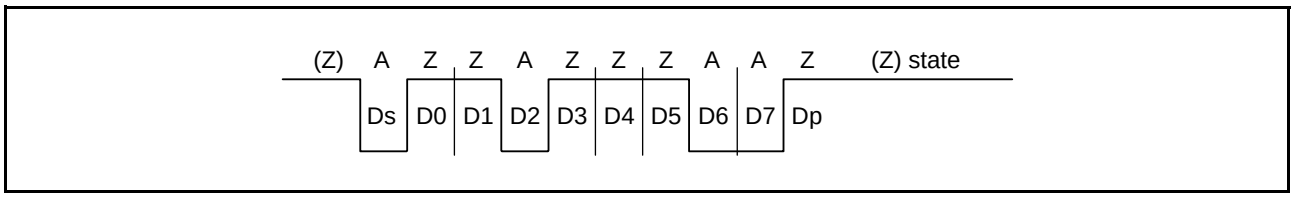


Figure 30.34 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

(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 30.35. 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 SCMR. 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 PM bit in SMR to invert the parity bit for both transmission and reception.

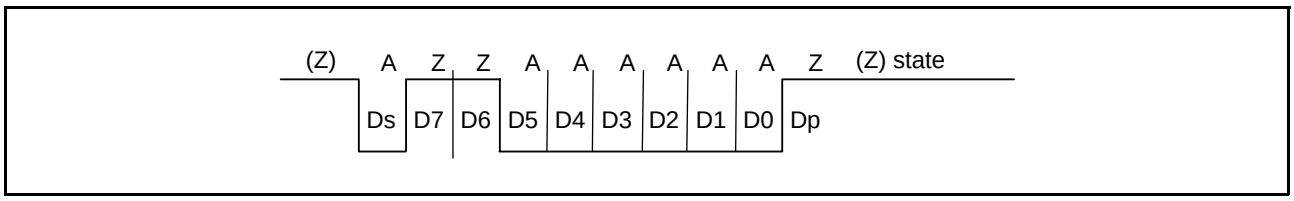


Figure 30.35 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

30.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit 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 SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred.

30.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer 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 BCP2 bit in SCMR and the BCP[1:0] bits in SMR (the frequency is always 16 times the bit rate in normal asynchronous mode).

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal 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 30.36. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{\left| D - 0.5 \right|}{N} (1 + F) \right| \times 100\ [\%]$$

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.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

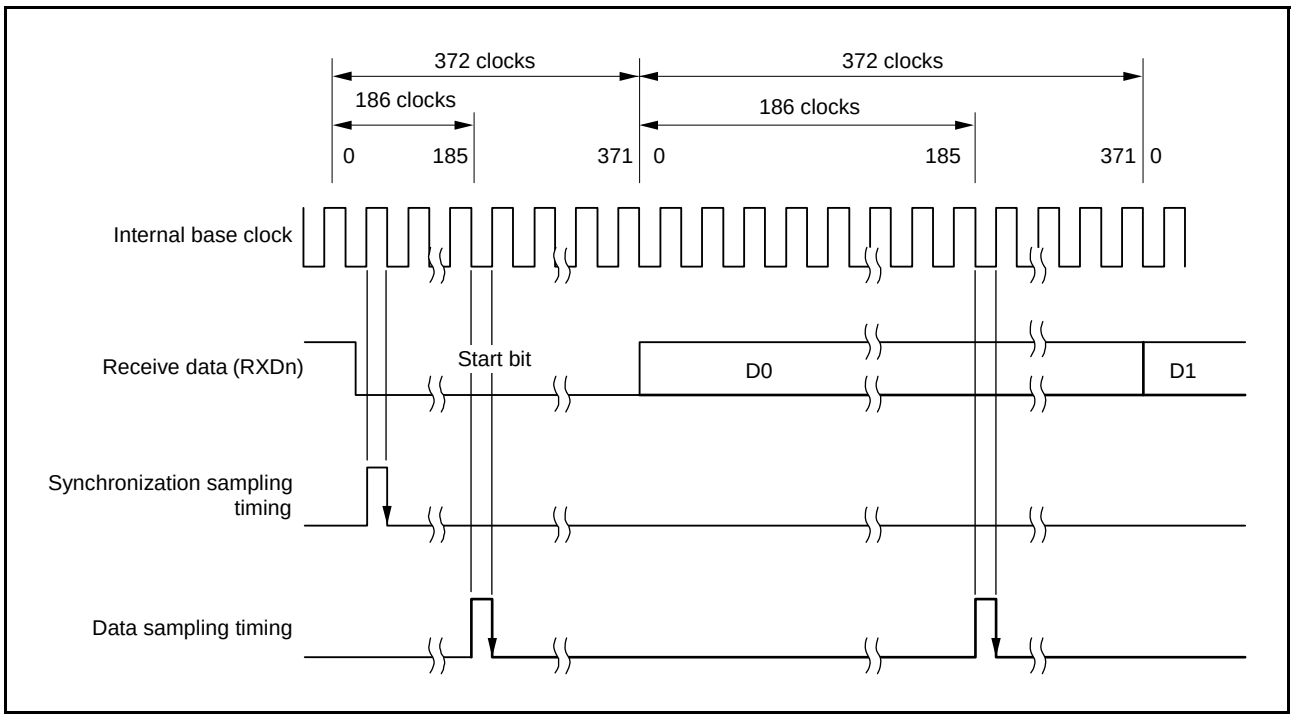


Figure 30.36 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

30.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 30.37.
Be sure to initialize the SCI 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.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.
To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

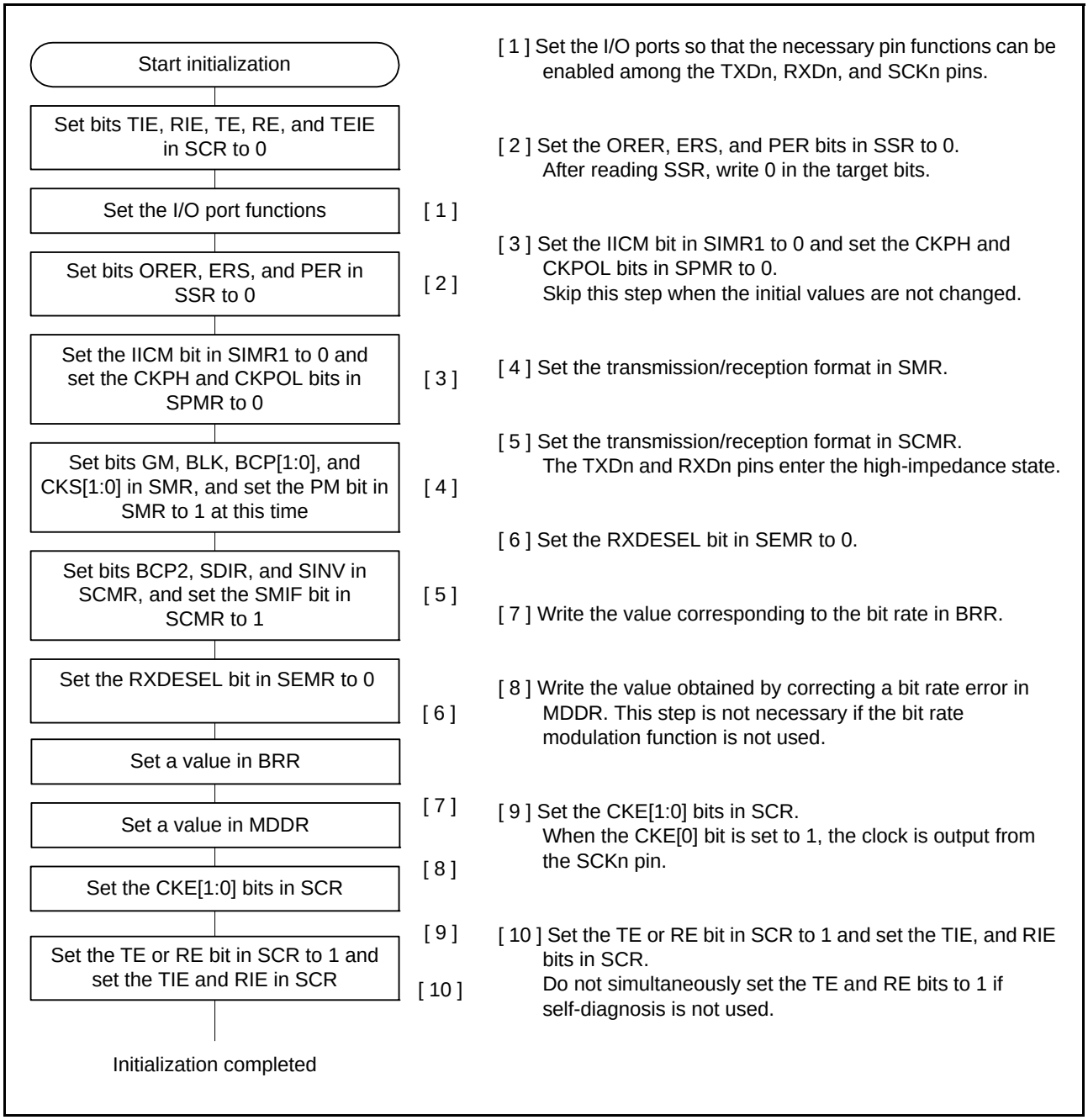


Figure 30.37 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

30.6.6 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 30.38 shows the data retransfer 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 SSR is set to 1. If the RIE bit in SCR 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 SSR is not set. Data is retransferred from TDR to TSR 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 retransfer) has been completed, and the TEND flag is set. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

Figure 30.40 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC.
When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR 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 to 0.
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 17, DMA Controller (DMACA), section 18, Data Transfer Controller (DTCa).

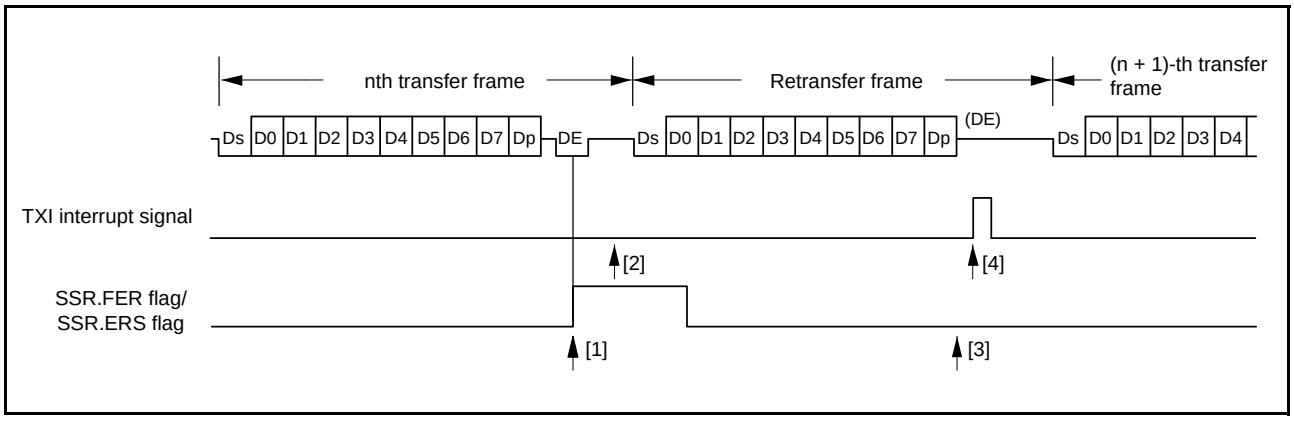


Figure 30.38 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 30.39 shows the TEND flag generation timing.

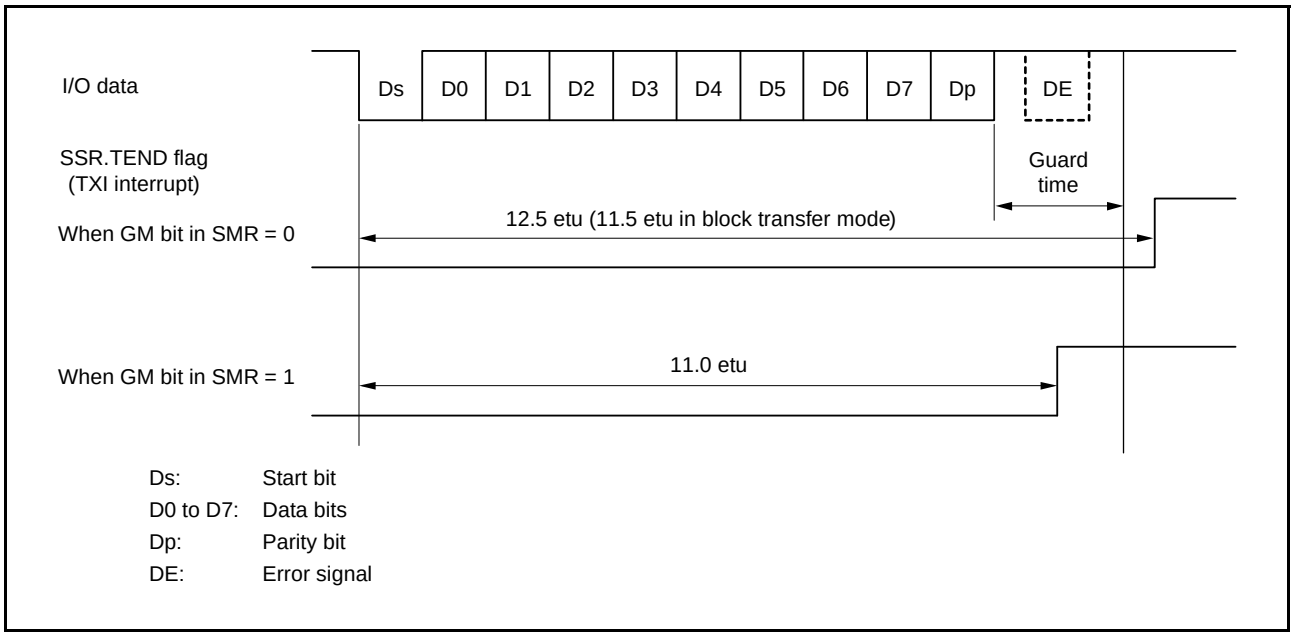


Figure 30.39 SSR.TEND Flag Generation Timing during Transmission

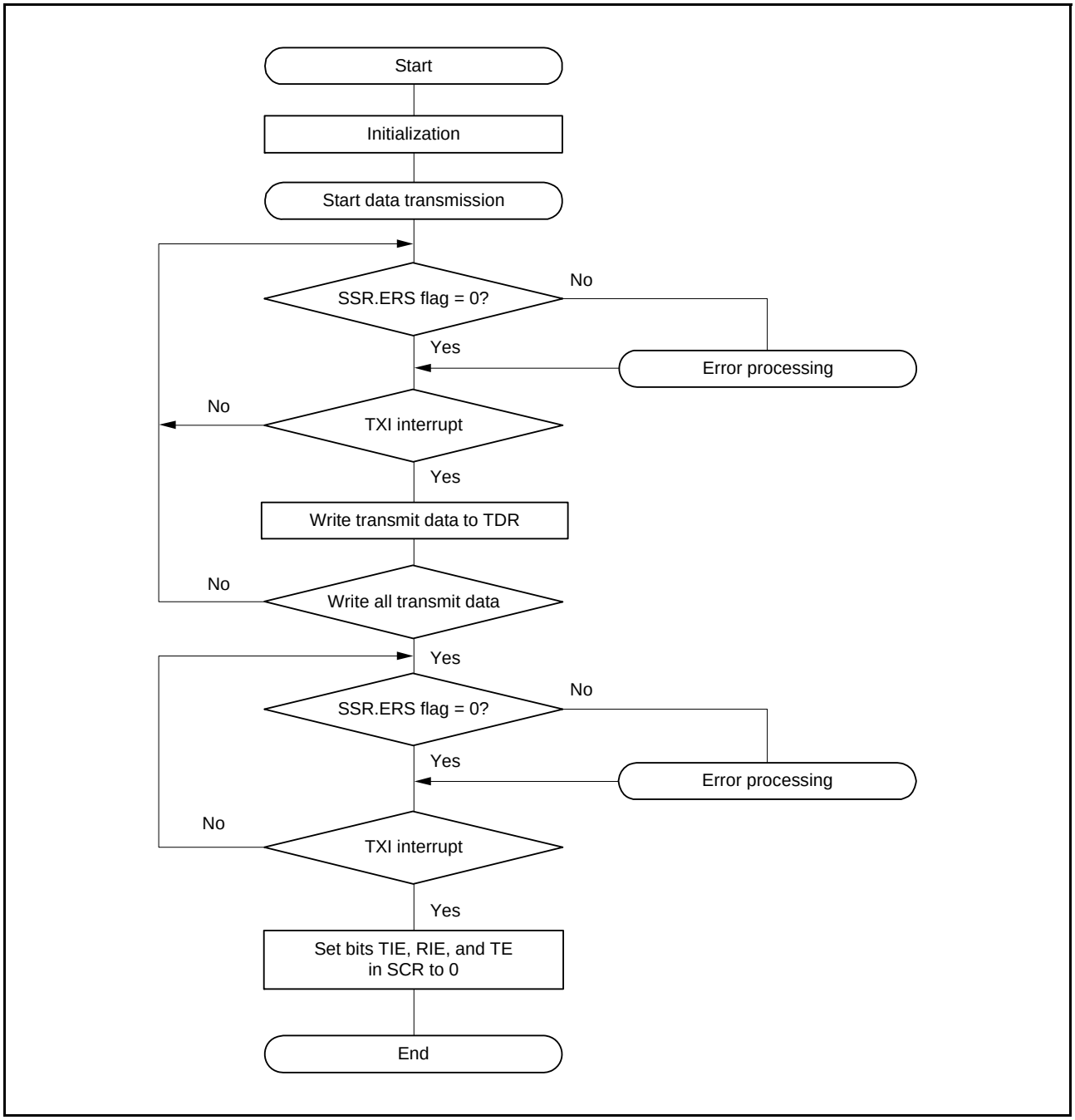


Figure 30.40 Sample Smart Card Interface Transmission Flowchart

30.6.7 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 30.41 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the PER 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 PER flag in SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

Figure 30.42 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 ORER or PER flag in SSR 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 RDR, thus allowing the data to be read.

Note 1. For operations in block transfer mode, refer to section 30.3, Operation in Asynchronous Mode.

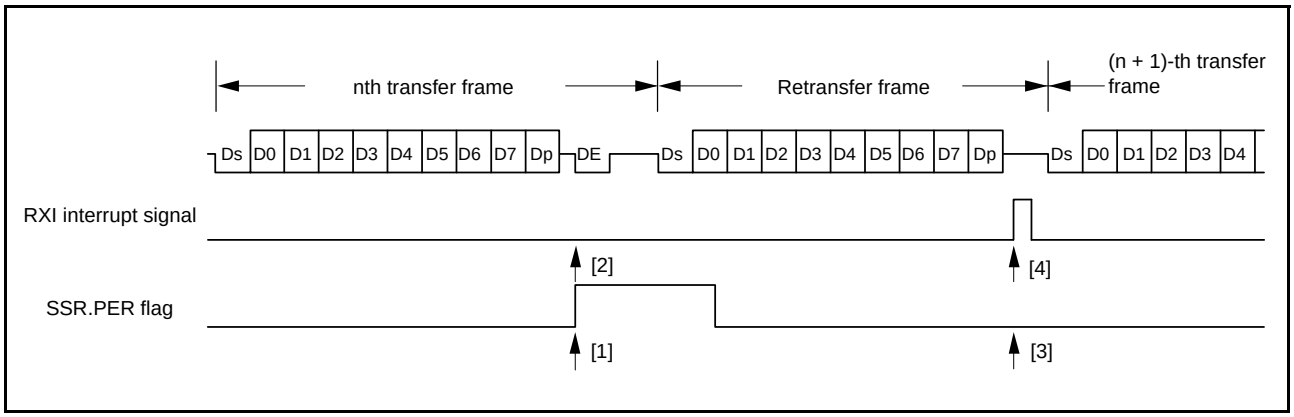


Figure 30.41 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)

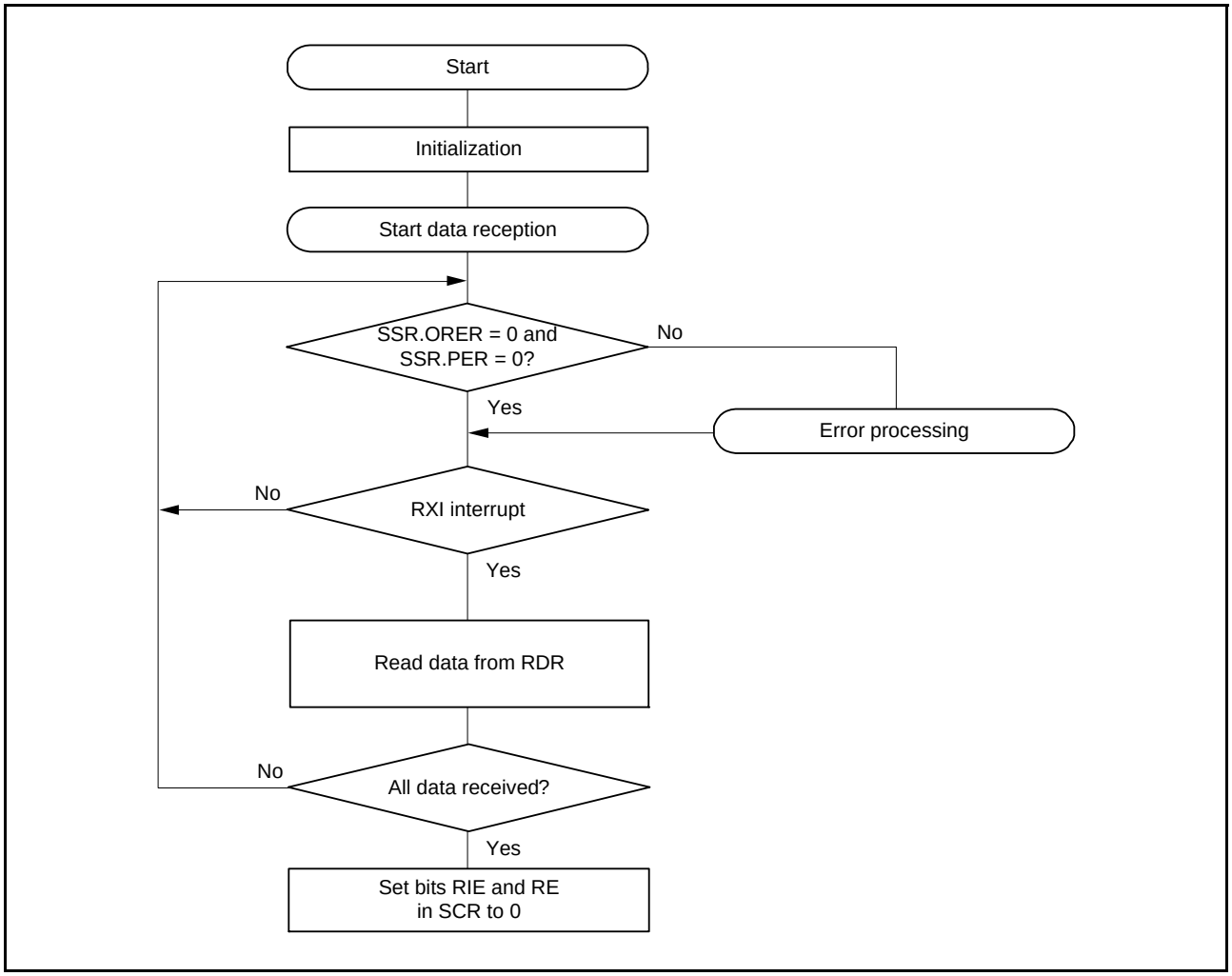


Figure 30.42 Sample Smart Card Interface Reception Flowchart

30.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 30.43 shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.

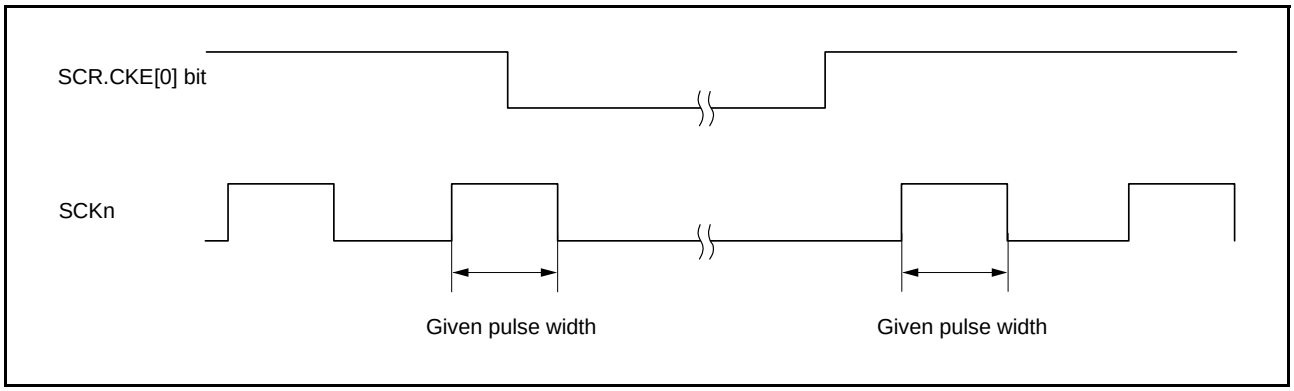


Figure 30.43 Clock Output Fixing Timing

At power-on, use the following procedure to secure the appropriate clock duty cycle.

(1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

30.7 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 format and timing of the I²C bus are shown in Figure 30.44 and Figure 30.45.

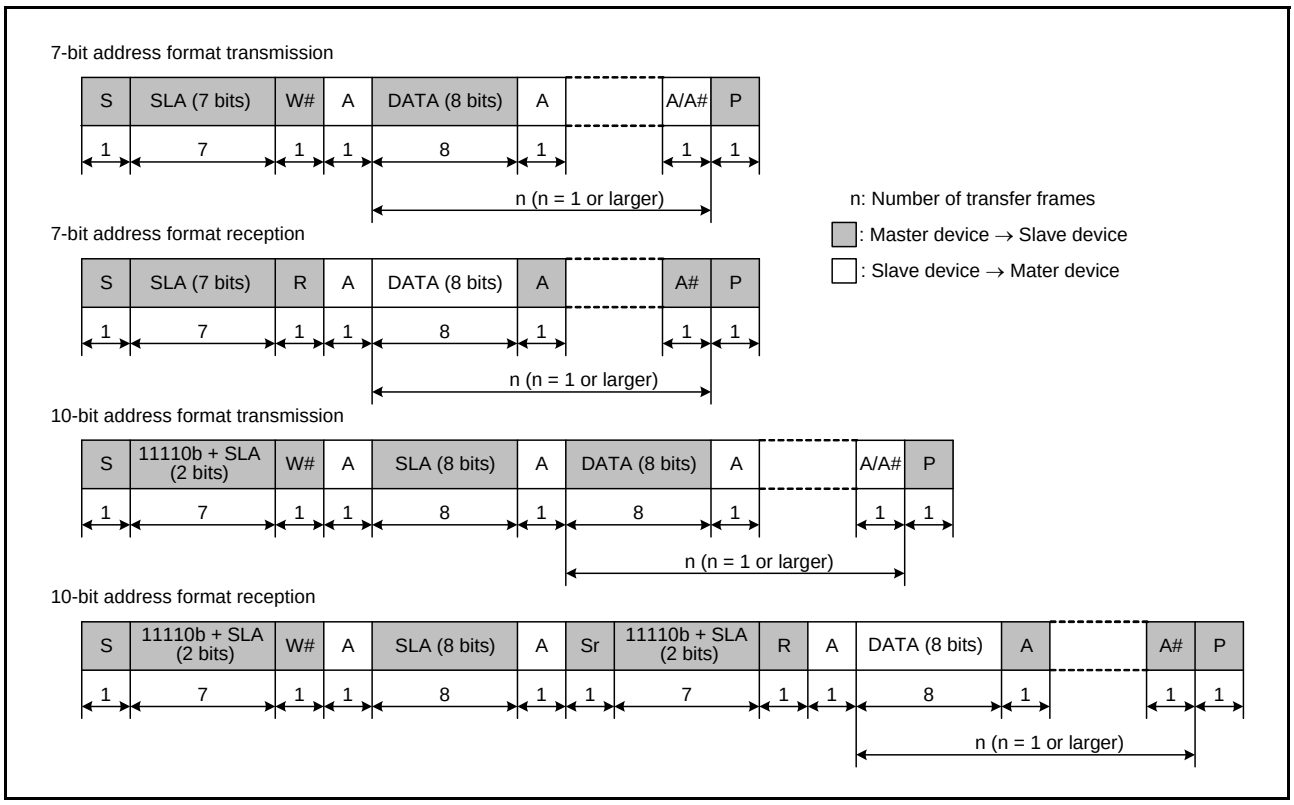


Figure 30.44 I²C Bus Format

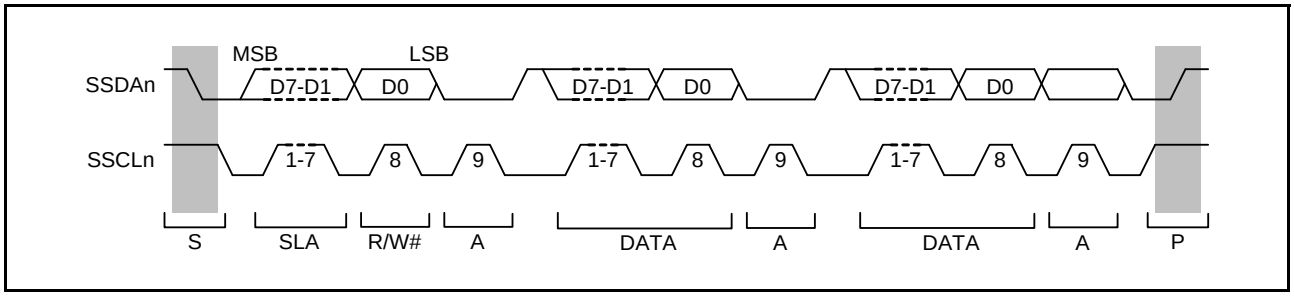


Figure 30.45 I²C Bus Timing (When SLA is 7 Bits)

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.

30.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in SIMR3 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 IICSTAREQ bit in SIMR3 is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in SIMR3 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 IICRSTAREQ bit in SIMR3 is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ bit in SIMR3 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 IICSTPREQ bit in SIMR3 is set (to 0), and a stop-condition generated interrupt is output.

Figure 30.46 shows the timing of operations in the generation of start, restart, and stop conditions.

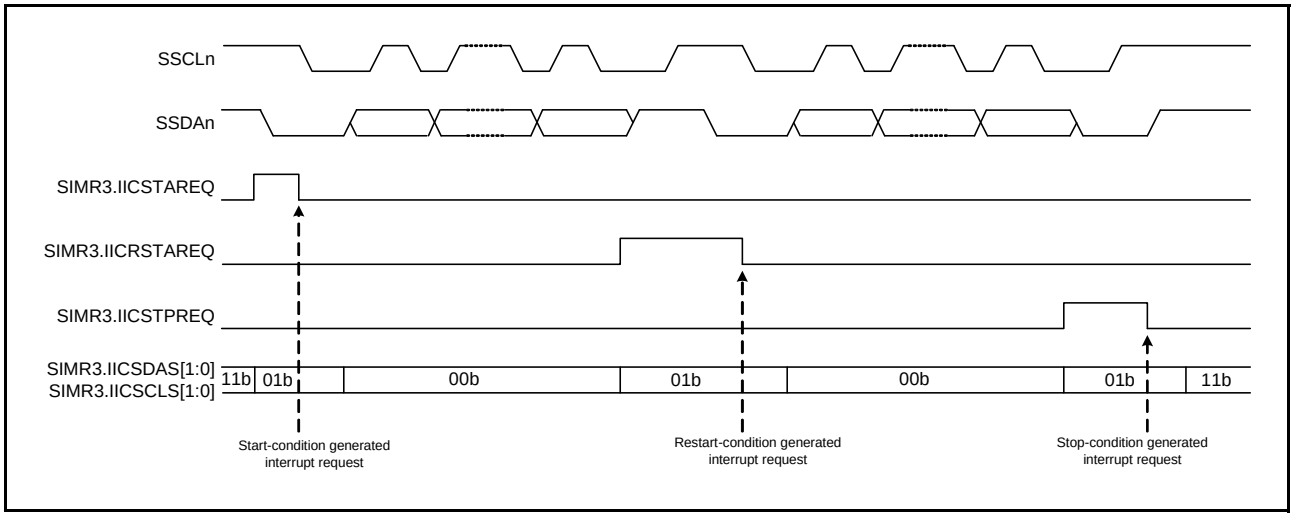


Figure 30.46 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

30.7.2 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 30.47 shows an example of operations to synchronize the clocks.

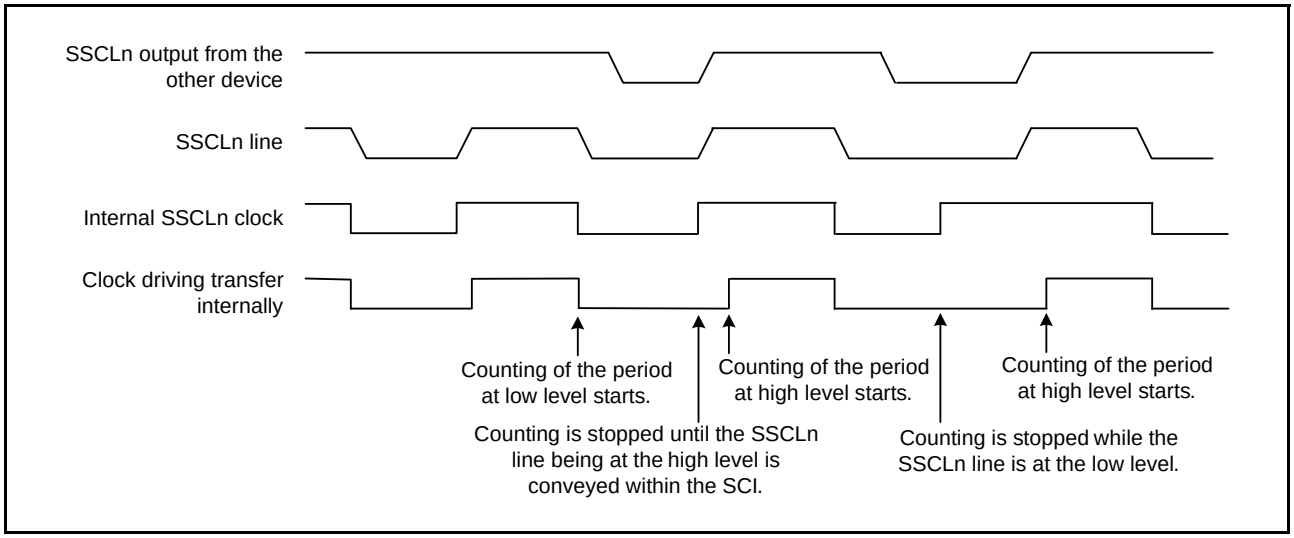


Figure 30.47 Example of Operations for Clock Synchronization

30.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 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 CKS[1:0] bits in SMR). 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 in normal mode and fast mode).

Figure 30.48 shows the timing of delays in SSDA output.

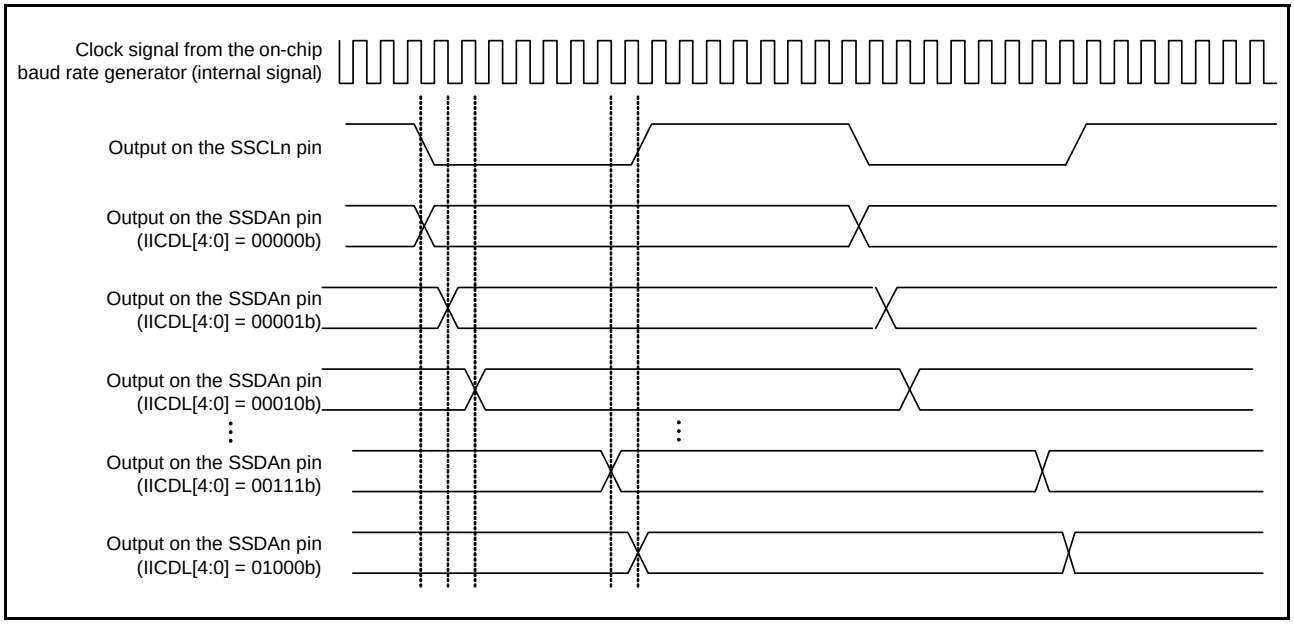


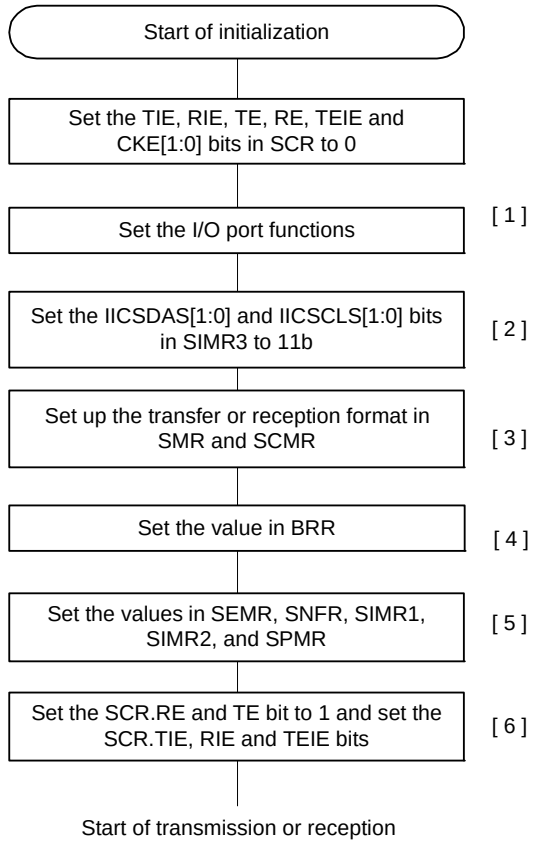
Figure 30.48 Timing of Delays in SSDA Output

30.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface following the example shown in Figure 30.49.

When changing the operating mode, transfer format, and so on, be sure to set SCR 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.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[2] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in SMR and SCMR.
In SMR, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to BRR.

[5] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set the NFEN bit.
In SNFR, set the NFCS[2:0] bits.
In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In SPMR, set all the bits to 0.

[6] Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0). Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Figure 30.49 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

30.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 30.50 and Figure 30.51 show examples of operations in master transmission and Figure 30.52 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (RXI and ERI interrupt requests are disabled). See Table 30.31 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 30.52 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.

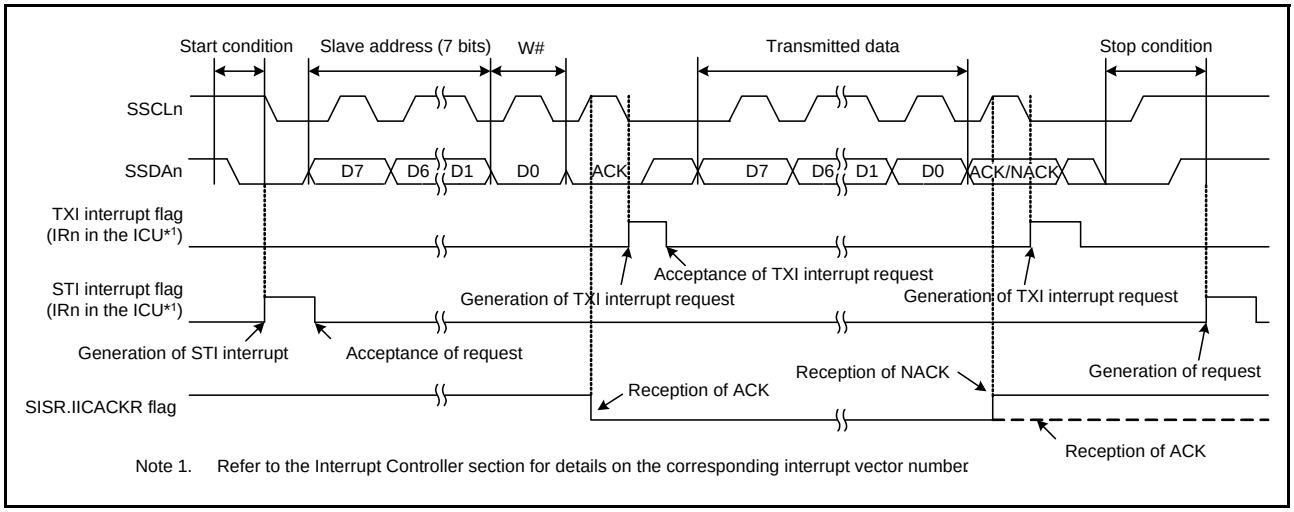


Figure 30.50 Example 1 of Operations for Master Transmission in Simple I2C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

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.

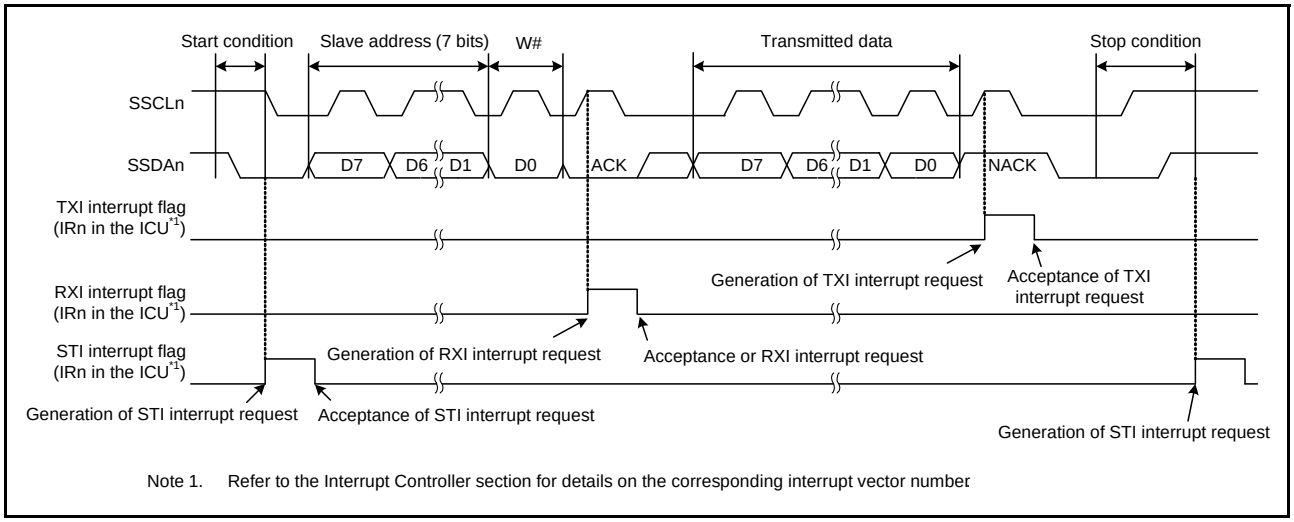
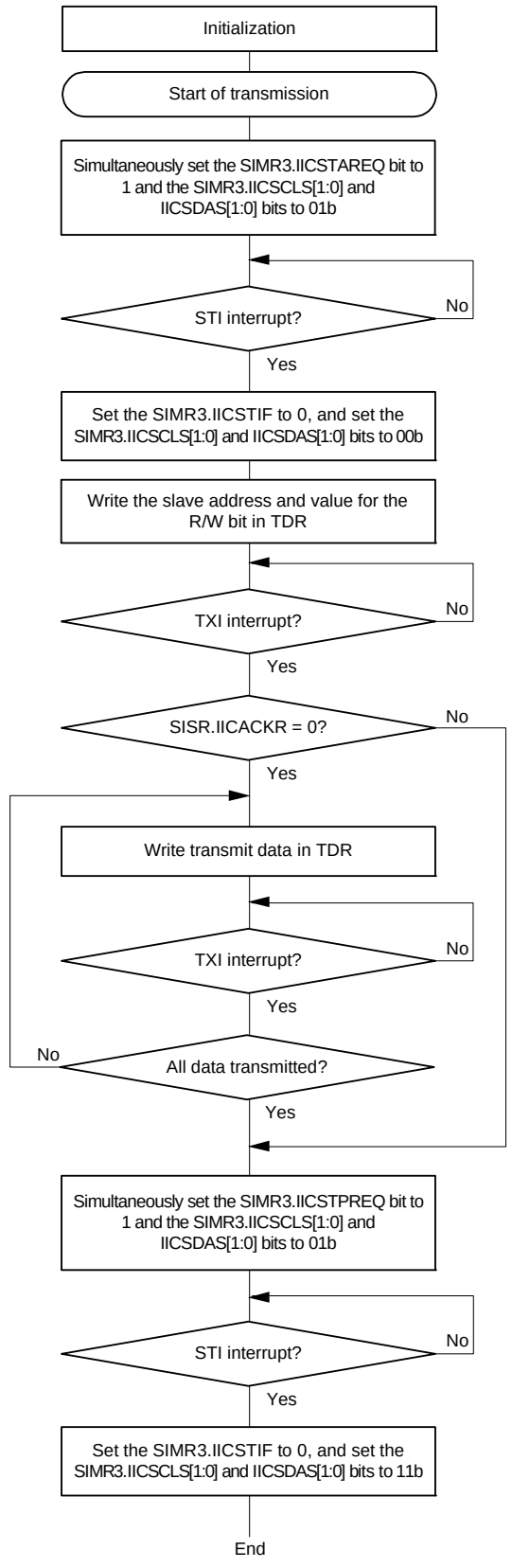


Figure 30.51 Example 2 of Operations for Master Transmission in Simple I2C Bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[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 TDR:
Writing the slave address and value for the R/W bit to TDR.

[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 TDR. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR.

[6] Generation of a stop condition.

[3] [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 30.52 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

30.7.6 Master Reception (Simple I²C Mode)

Figure 30.53 shows an example of operations in simple I²C mode master reception and Figure 30.54 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.

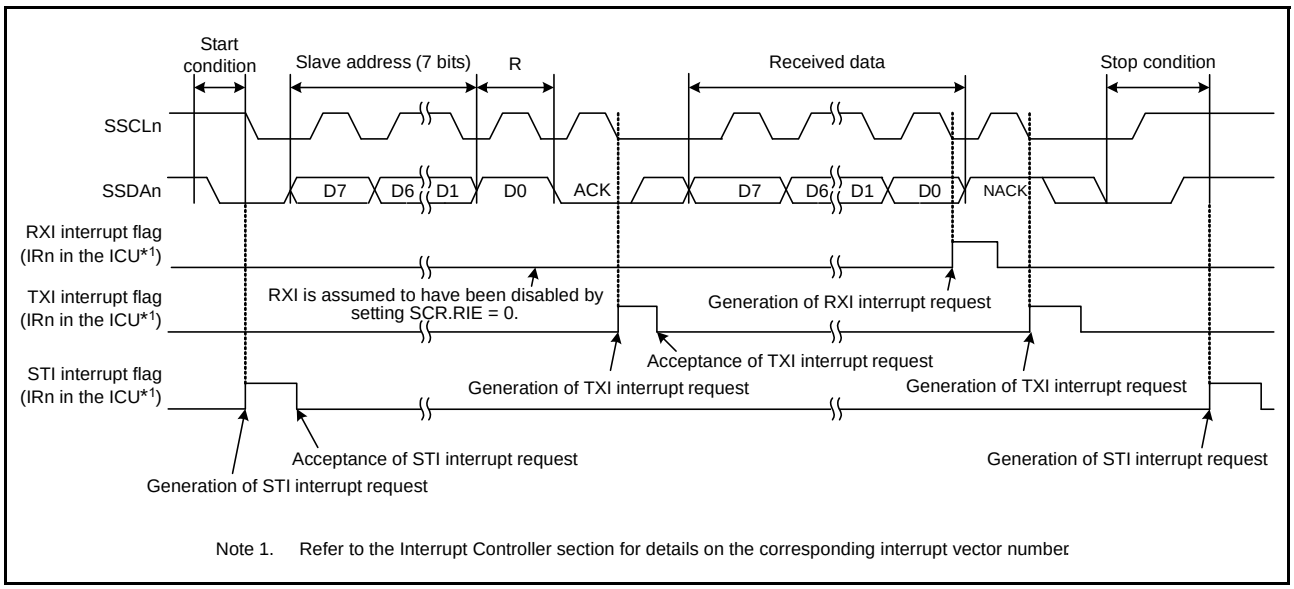
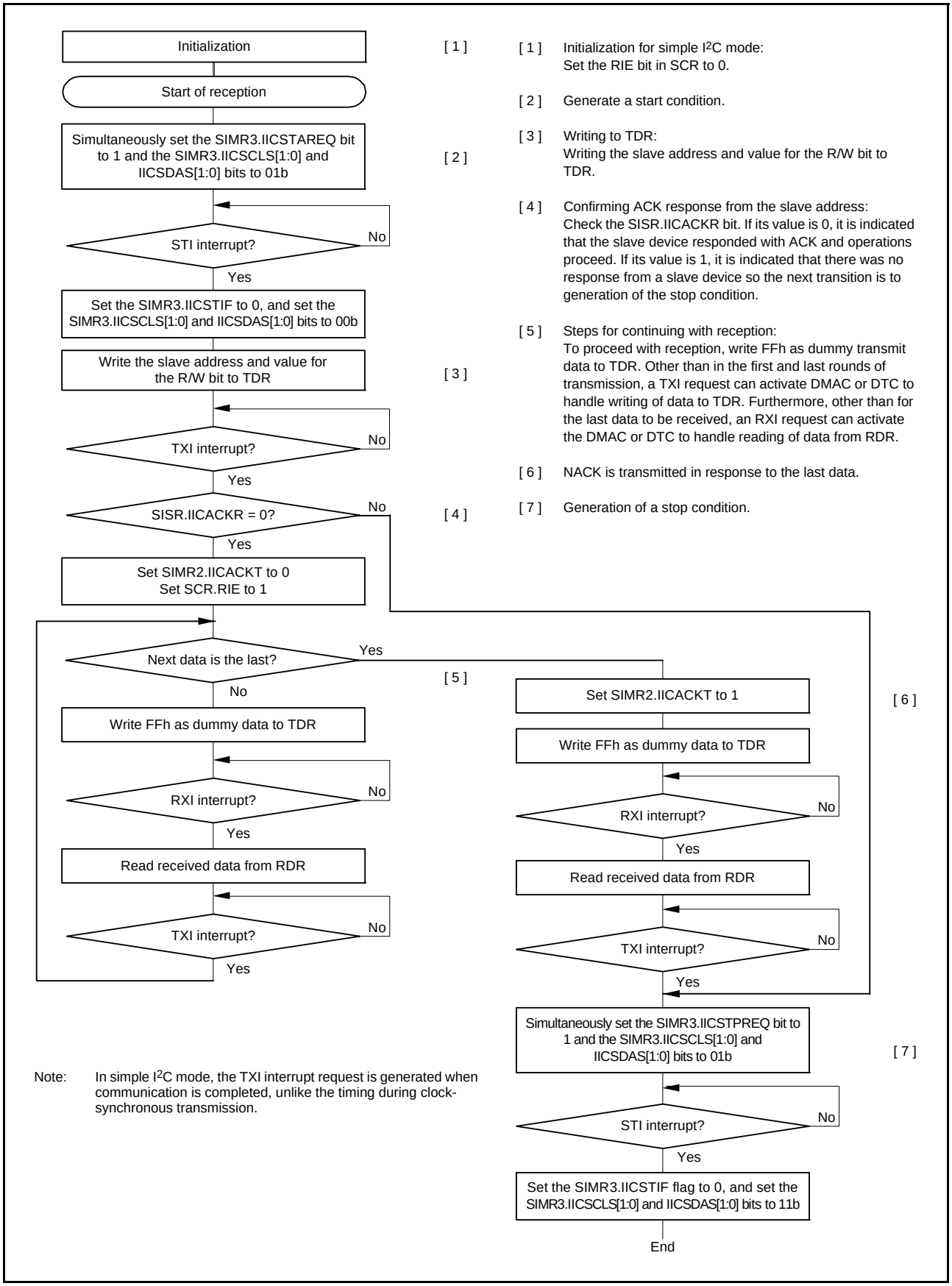


Figure 30.53 Example of Operations for Master Reception in Simple I²C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the RIE bit in SCR to 0.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[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 TDR. Other than in the first and last rounds of transmission, a TXI request can activate DMAC or DTC to handle writing of data to TDR. 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 RDR.

[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 30.54 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

30.8 Operation in Simple SPI Mode

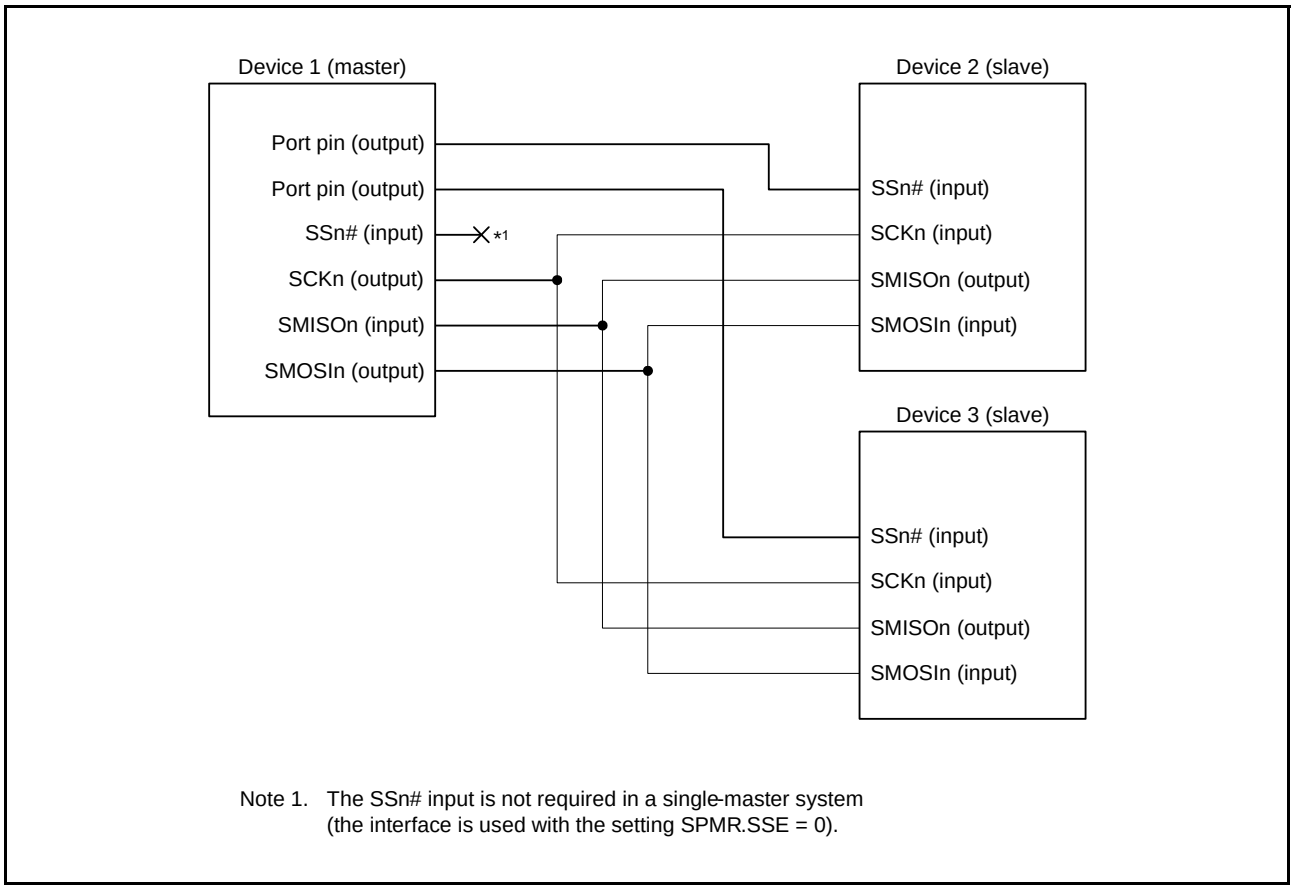
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SPMR 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 SSE bit in the SPMR to 0 in such cases.

Figure 30.55 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 30.55 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

30.8.1 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 30.27 lists the states of pins according to the mode and the level on the SSn# pin.

Table 30.27 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 can proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer cannot 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).

30.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b and the MSS bit in the SPMR 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.

30.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10b and the MSS bit in the SPMR 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.

30.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR 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 30.56. The relation is the same for both master and slave operation. This is the same as when the level on the SSn# pin is high. The SSn# pin can be used for another purpose. For details, refer to section 30.8.2, SS Function in Master Mode.

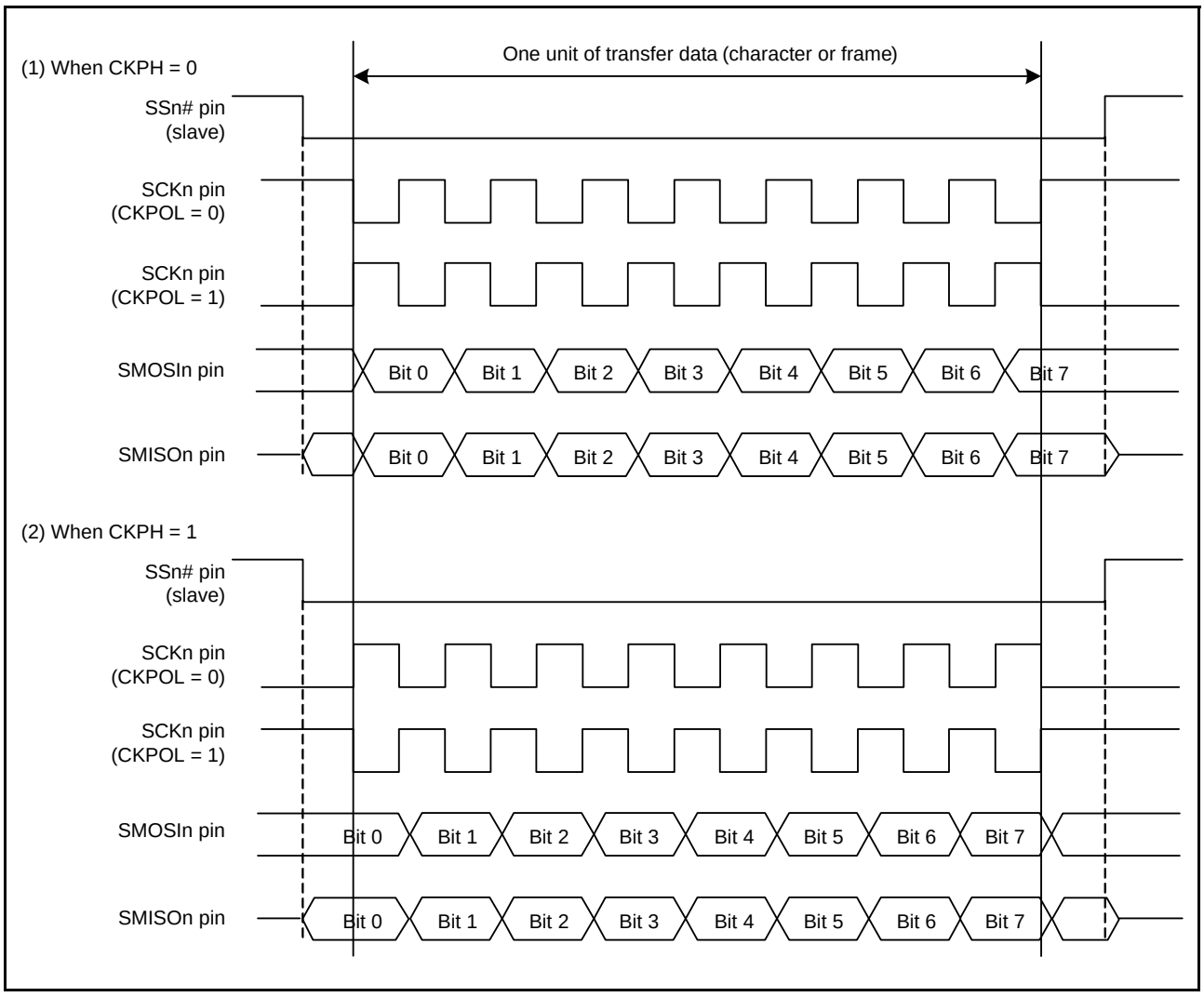


Figure 30.56 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

30.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 30.23, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR 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, are not initialized. Note that changing the value of the TE bit from 1 to 0 or from 0 to 1 will lead to the generation of a transmit data empty interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

30.8.6 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.

30.9 Extended Serial Mode Control Section: Description of Operation

30.9.1 Serial Transfer Protocol

In conjunction with the SCIe module, the extended serial mode control section of the SCIf module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 30.57.
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.

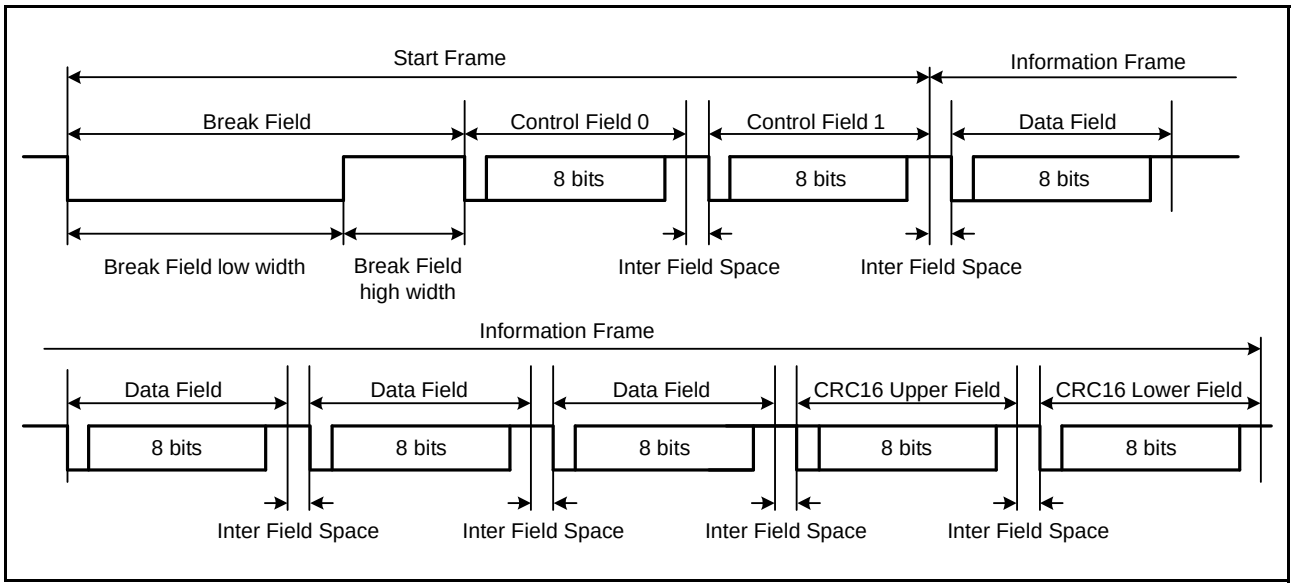


Figure 30.57 Protocol for Serial Transfer by the Extended Serial Mode Control Section

30.9.2 Transmitting a Start Frame

Figure 30.58 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 30.59 and Figure 30.60 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 TCST bit in TCR starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to the TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) Writing 0 to the TCST bit in TCR stops counting by the timer, and SCI2 is used to send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) Once the data for Control Field 0 have been transmitted, SCI12 is used to send the data for Control Field 1.
(5) Once the data for Control Field 1 have been transmitted, SCI12 is used to send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

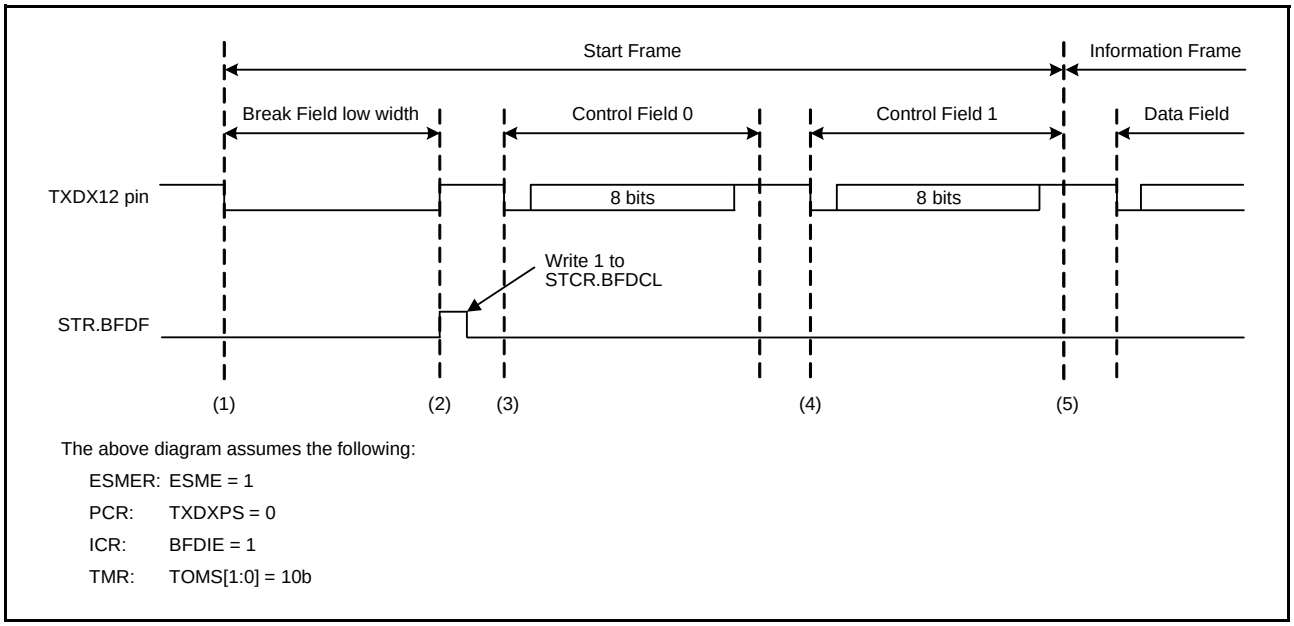


Figure 30.58 Example of Operations When Transmitting a Start Frame

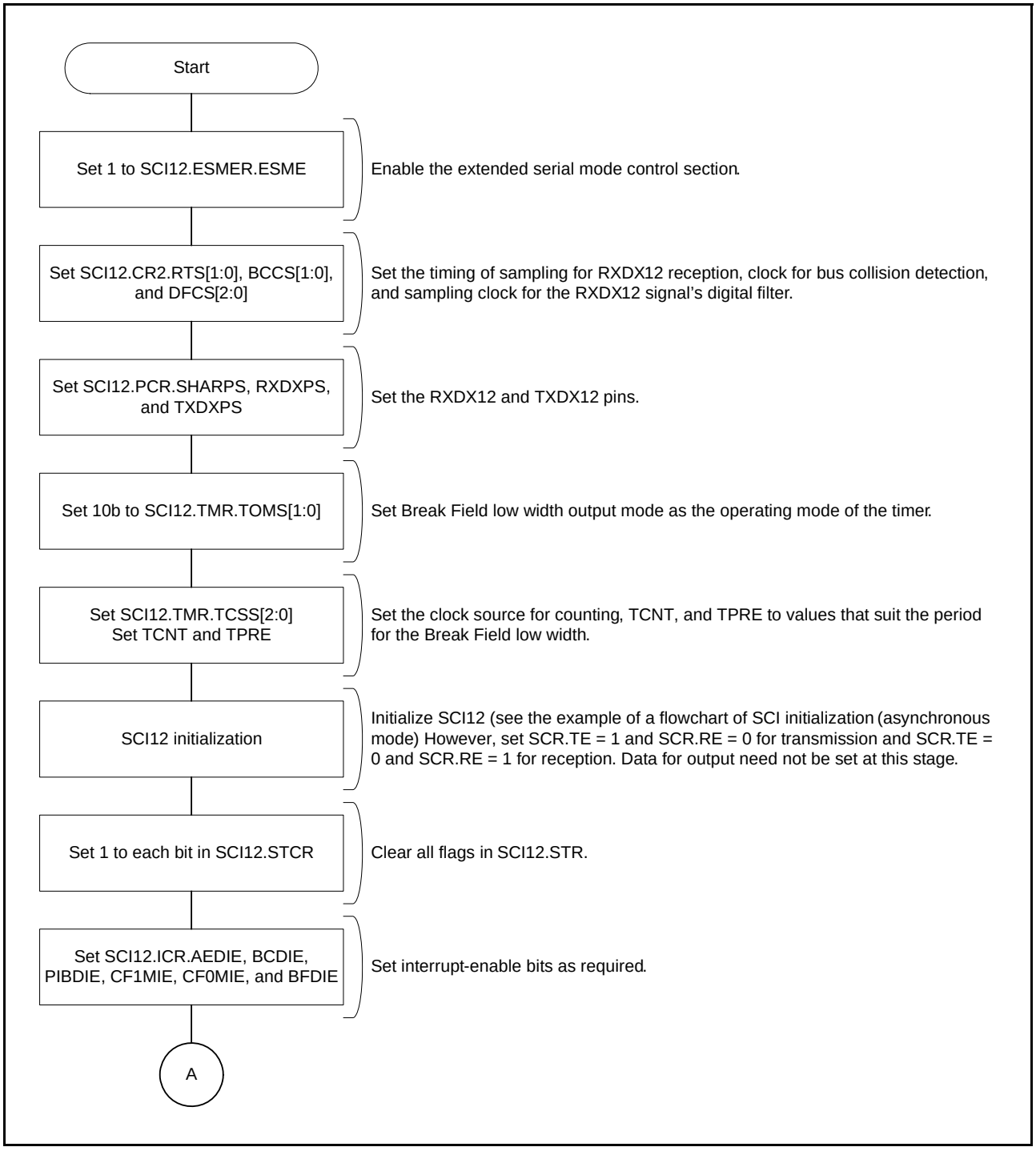


Figure 30.59 Example of Start Frame Transmission (1/2)

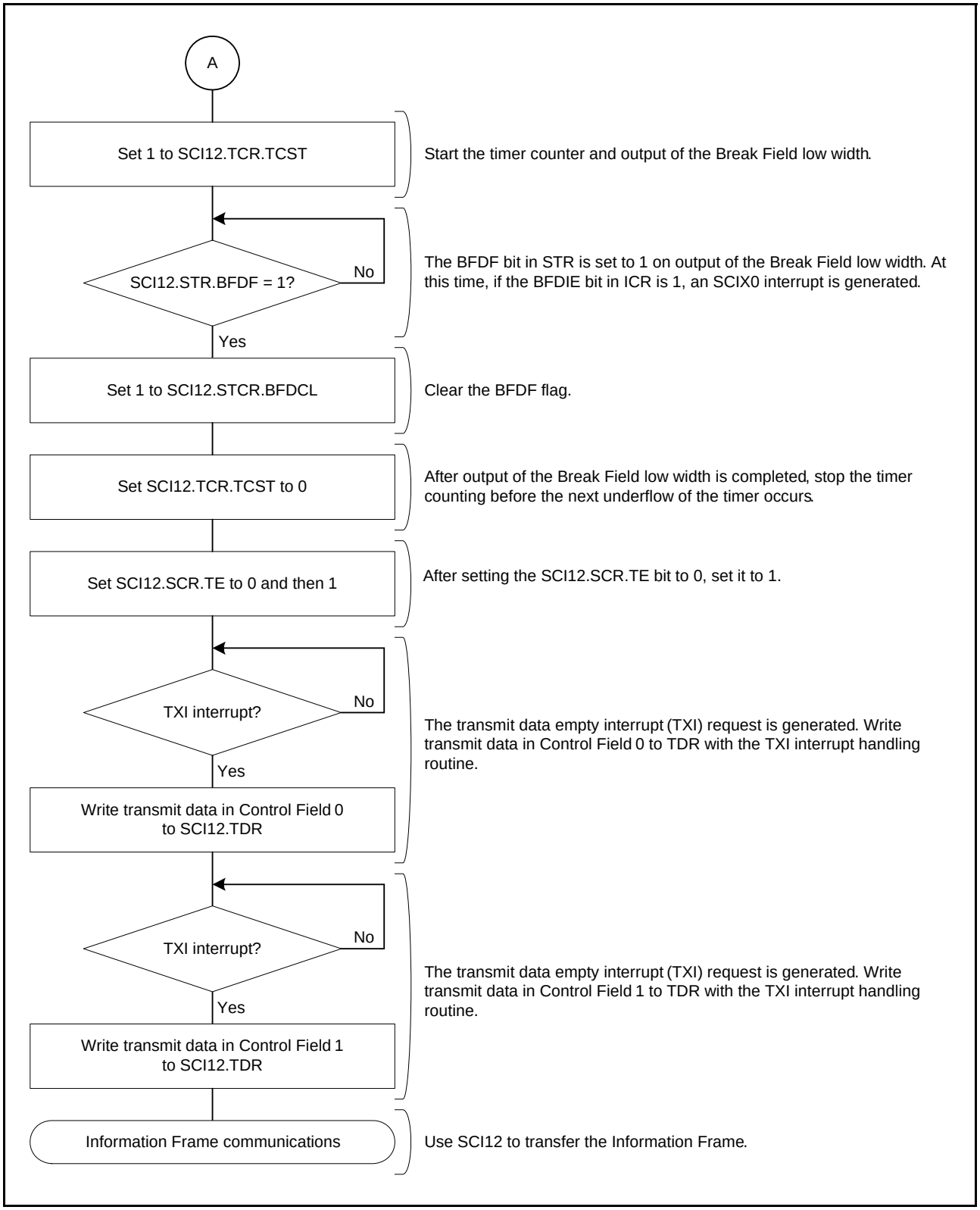


Figure 30.60 Example of Start Frame Transmission (2/2)

30.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 30.28.

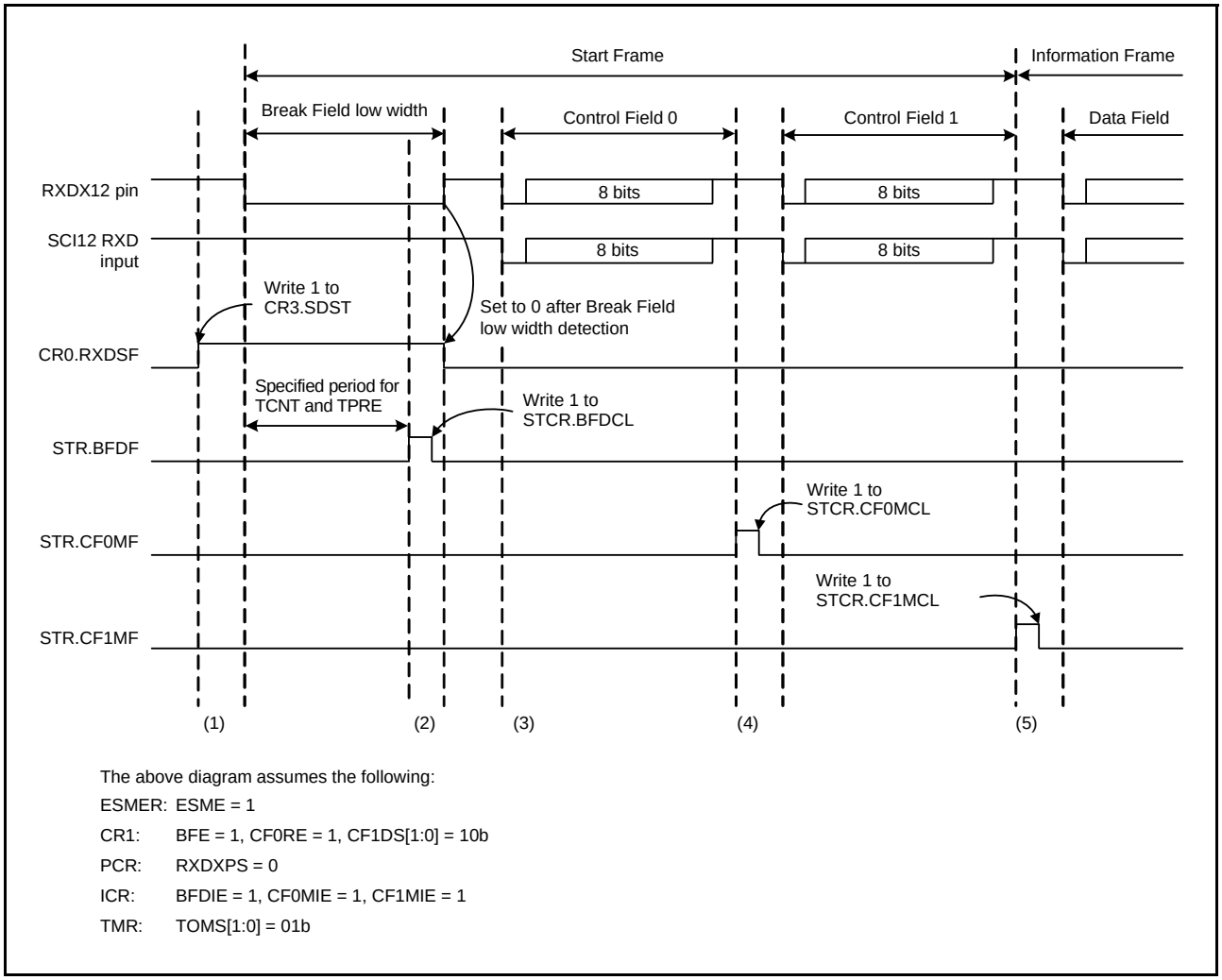
Table 30.28 Structures of Start Frames

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 30.61 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 30.62 and Figure 30.63 are flowcharts for the reception of a Start Frame, and Figure 30.64 is a state transition diagram for the extended serial mode control section.
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 SDST bit in CR3 enables detection of the Break Field low width. RXDX12 input to the SCI12 is disabled at this time.
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of TCNT and TPRE is detected as the Break Field low width. At this time, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the RXDSF bit in CR0 becomes 0 and reception of Control Field 0 by the SCI12 starts.
(4) If the data received in Control Field 0 match the data set in CF0DR, the CF0MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF0MIE bit in ICR is 1. Reception of Control Field 1 by the SCI12 starts after that. If the data received in Control Field 0 do not match the data set in CF0DR, 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 PCF1DR and SCF1DR, the CF1MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF1MIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of 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.



The above diagram assumes the following:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS[1:0] = 10b

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS[1:0] = 01b

Figure 30.61 Example of Operations at the Time of Start Frame Reception

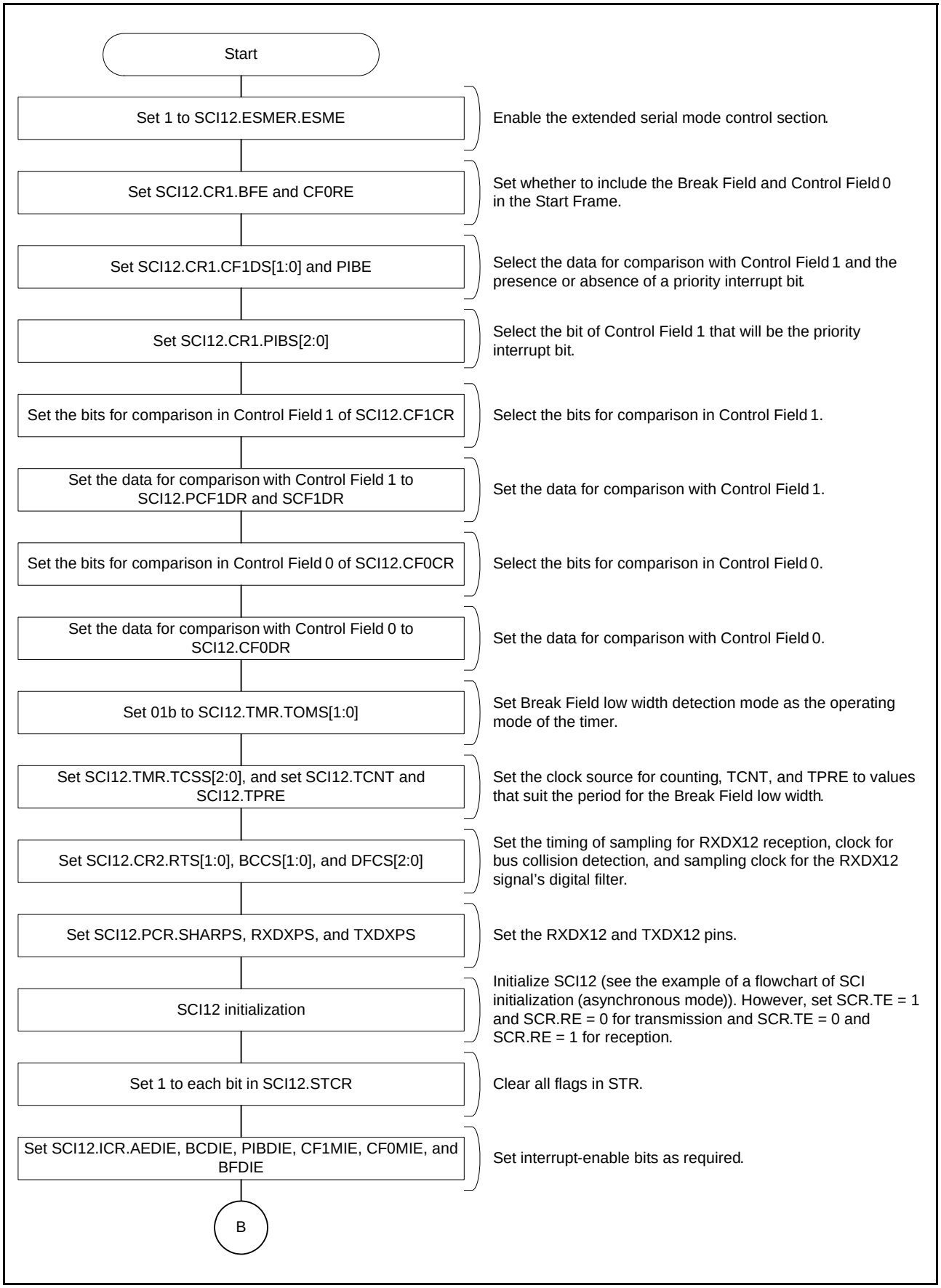


Figure 30.62 Sample Flowchart for Reception of a Start Frame (1)

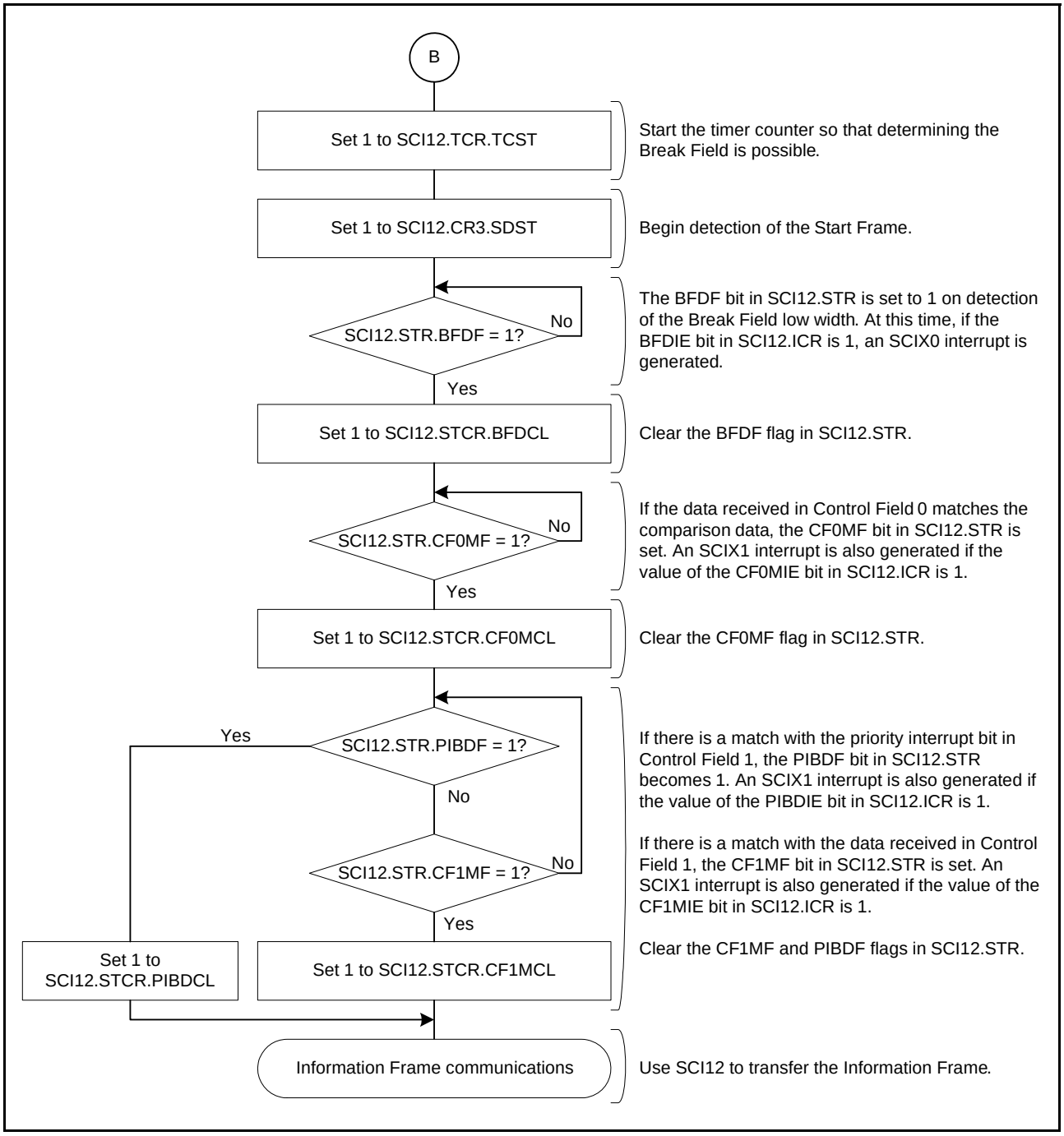


Figure 30.63 Sample Flowchart for Reception of a Start Frame (2)

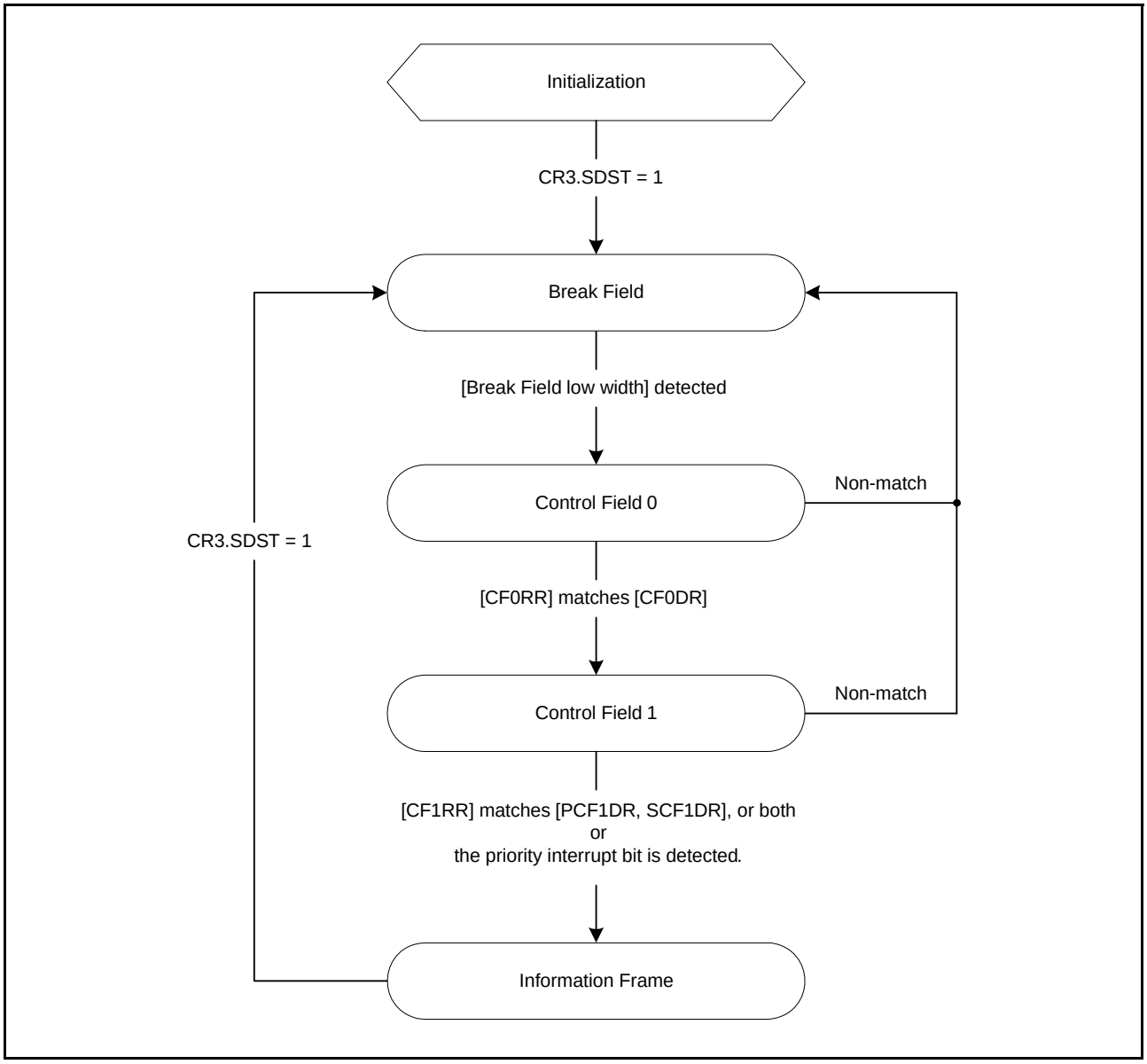


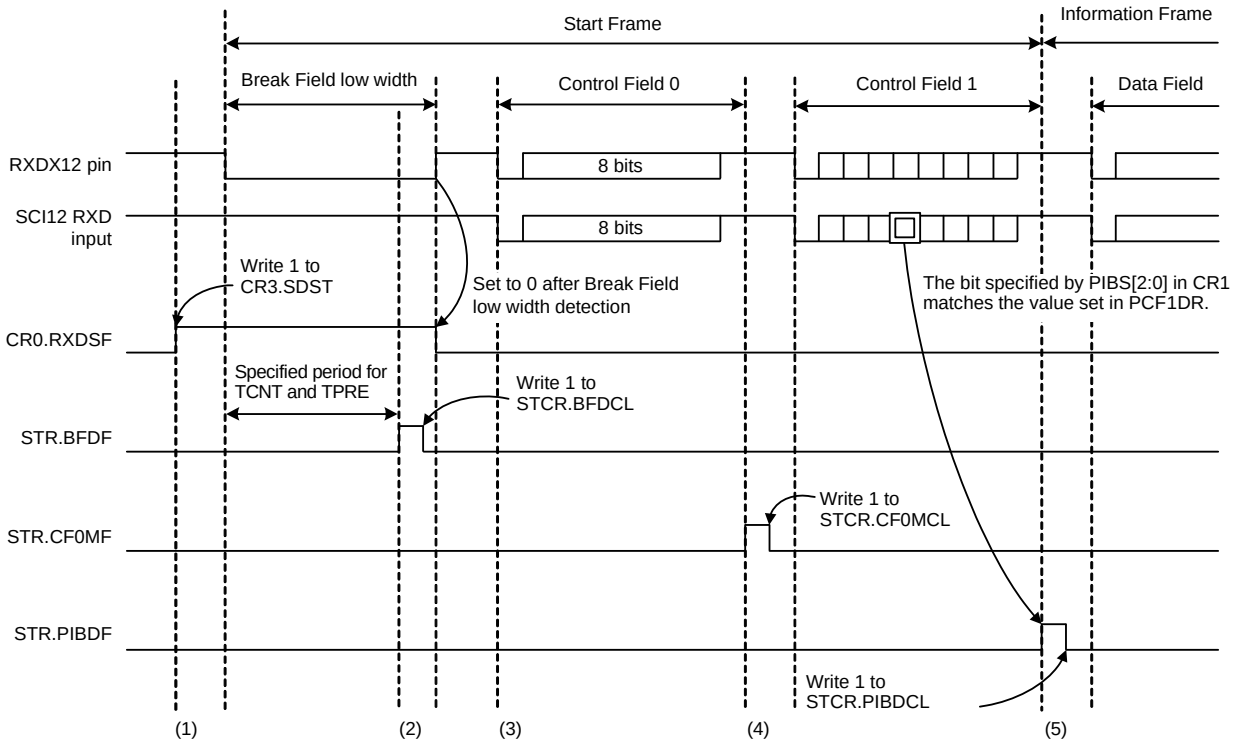
Figure 30.64 State Transitions When Receiving a Start Frame

30.9.3.1 Priority Interrupt Bit

Figure 30.65 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the PIBE bit in CR1 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 30.61, for Start Frame reception.

(5) If the value of the bit selected by the PIBS[2:0] bits in CR1 matches the corresponding bit in PCF1DR, the PIBDF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the PIBDIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.



The above diagram assumes the following:
ESMER: ESME = 1
CR1: BFE = 1, PIBE = 1, CF0RE = 0, CF1DS[1:0] = 10b, PIBS[2:0] = 011b
PCR: RXDXPS = 0
ICR: BFDIE = 1, CF0MIE = 1, PIBDIE = 1
TMR: TOMS[1:0] = 01b

Figure 30.65 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

30.9.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data by the SCI12 are in progress when the ESMER.ESME and the SCI12.SCI.TE are set to 1.

Figure 30.66 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 CR2.BCCS[1:0] as the sampling clock, and the BCDF bit in STR 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 BCDIE bit in ICR is 1.

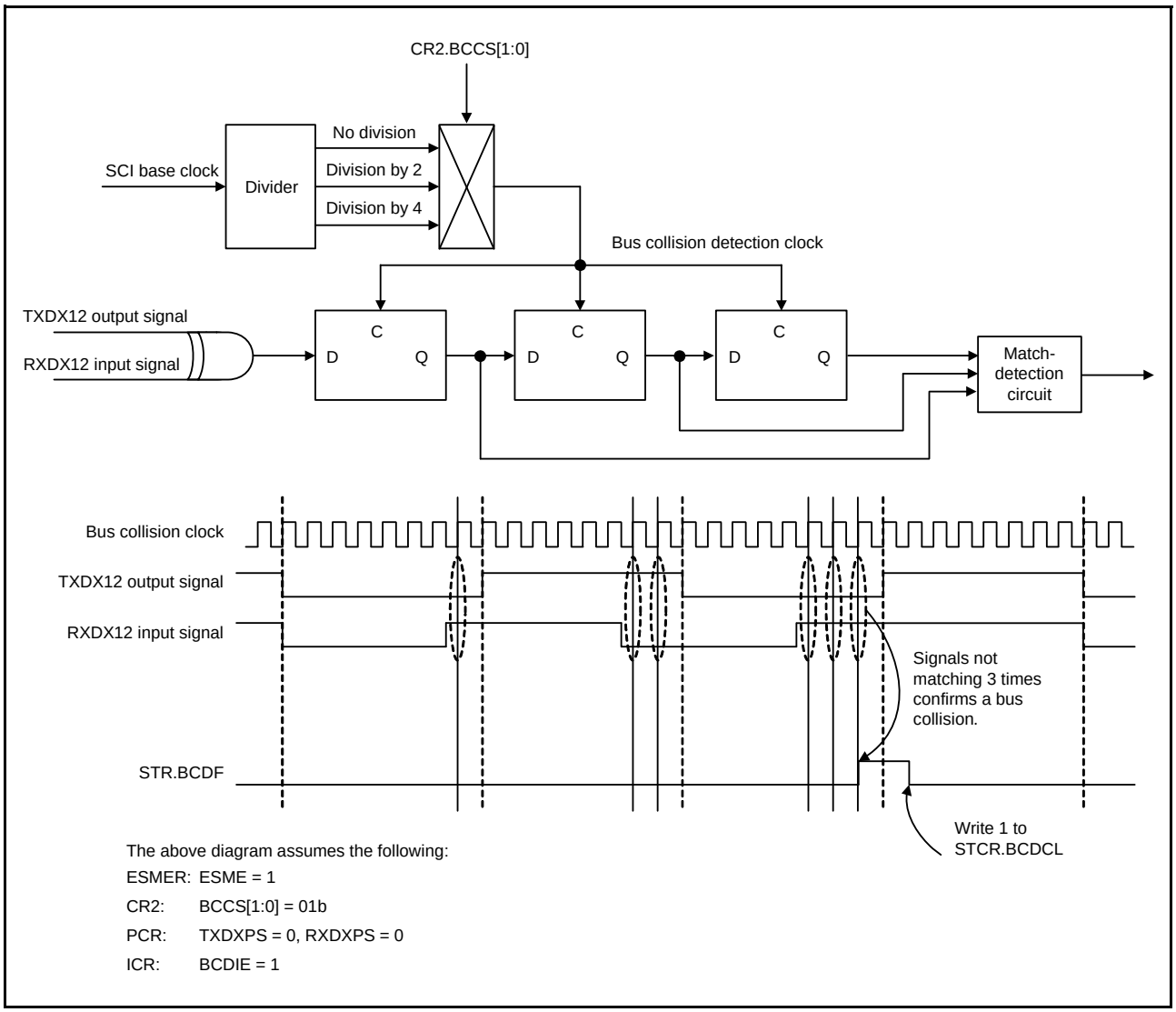


Figure 30.66 Example of Operations with Bus Collision Detection

30.9.5 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 DFCS[2:0] bits in CR2 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 30.67 shows an example of operations with the digital filter.

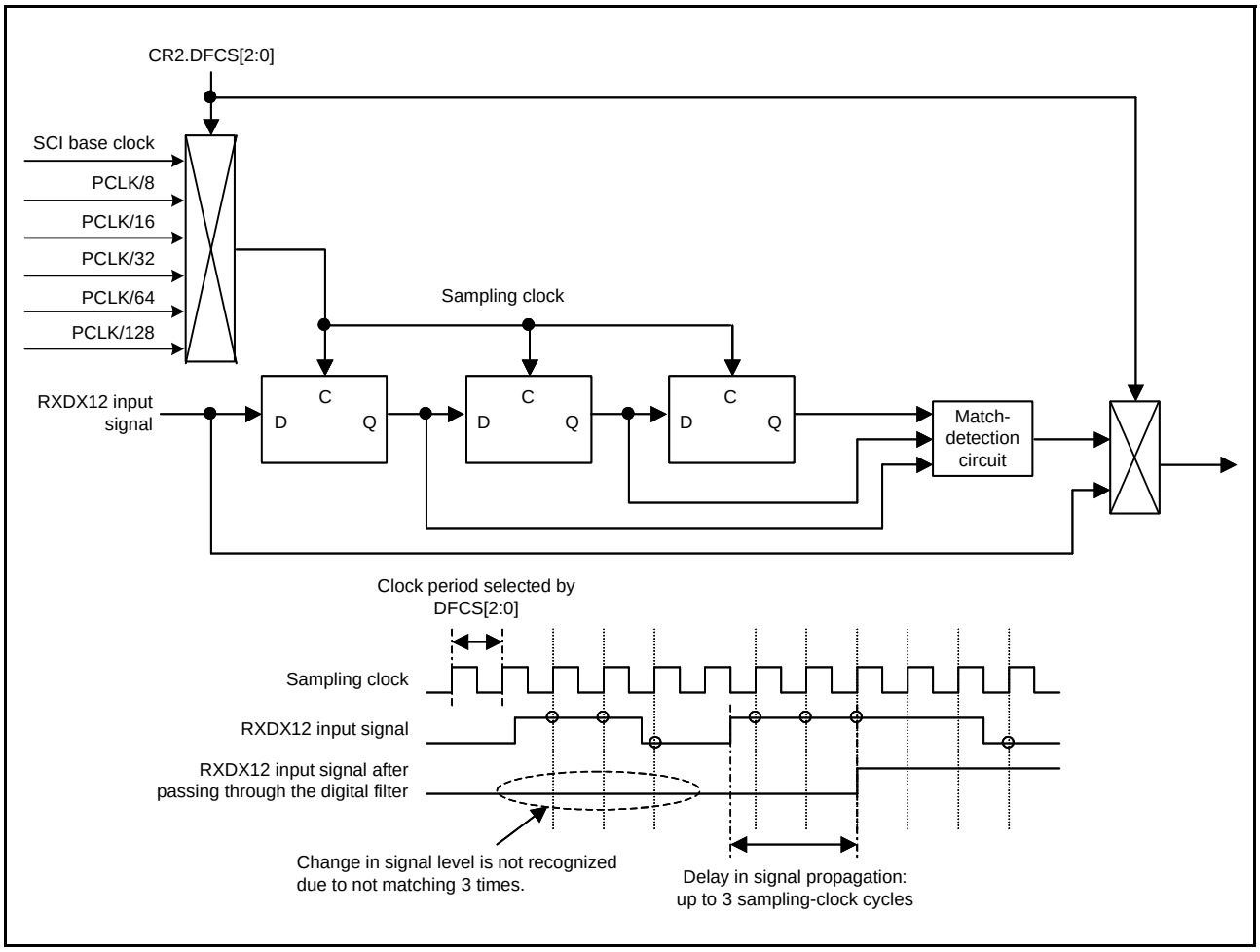


Figure 30.67 Example of Operations with the Digital Filter

30.9.6 Bit Rate Measurement

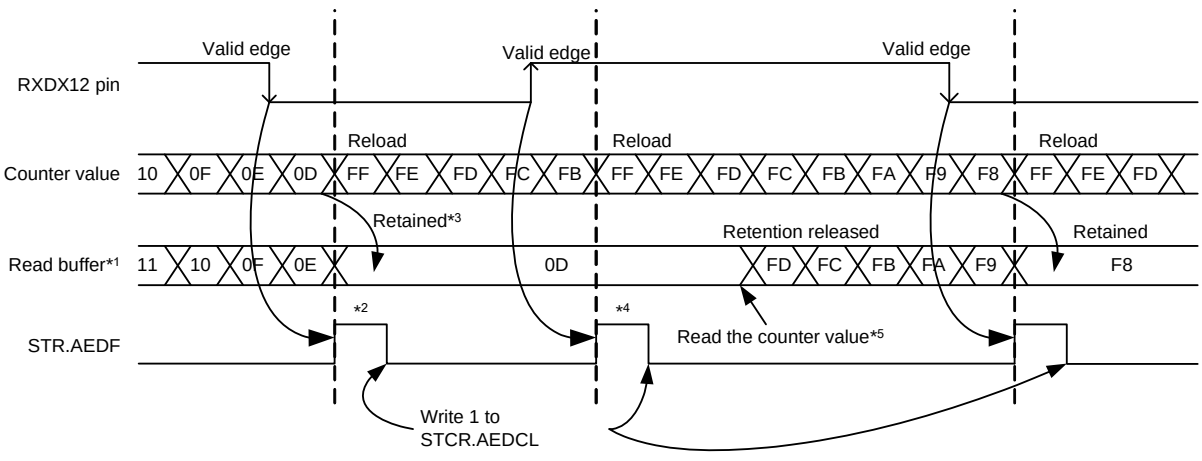
The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 30.68 shows an example of operations for bit rate measurement.

(1) Writing 1 to the BRME bit in CR0 enables bit rate measurement. Only set BRME to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if BRME 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 AEDIE bit in ICR is 1. Retention by 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 SCI12. To disable the bit rate measurement after a match with Control Field 1, write 0 to the BRME bit in CR0.



Note 1. The contents of the read buffers are read by reading TCNT and TPRE.
Note 2. The AEDF bit in STR becomes 1 on detection of a valid edge.
Note 3. The current counter value is retained in the read buffers on detection of a valid edge.
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 TCNT or TPRE.

Figure 30.68 Example of Operations for Bit Rate Measurement

30.9.7 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 of an SCI12 by setting the CRS.RTS[1:0] bits to select the rising edges of 8, 10, 12, or 14 cycles of the SCI base clock. If the value of the ABCS bit in SEMR is 1, the bits select the rising edges of 4, 5, 6, or 7 cycles of the PCLK clock of the SCI12. Figure 30.69 shows timing for the sampling of data received through RXDX12.

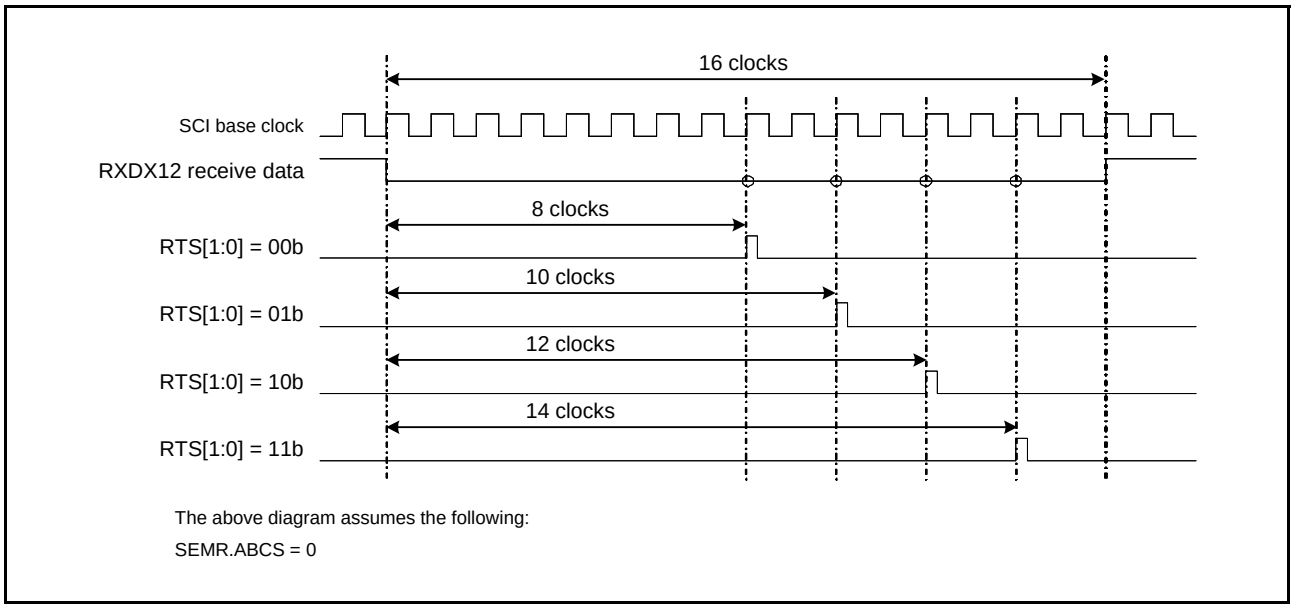


Figure 30.69 Timing for Sampling of Data Received through RXDX12

30.9.8 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 TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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 BFDF bit in the STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1. When 0 is written to the TCST bit in TCR, counting stops after reloading of TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 30.70 shows an example of operations in Break Field low width output mode.

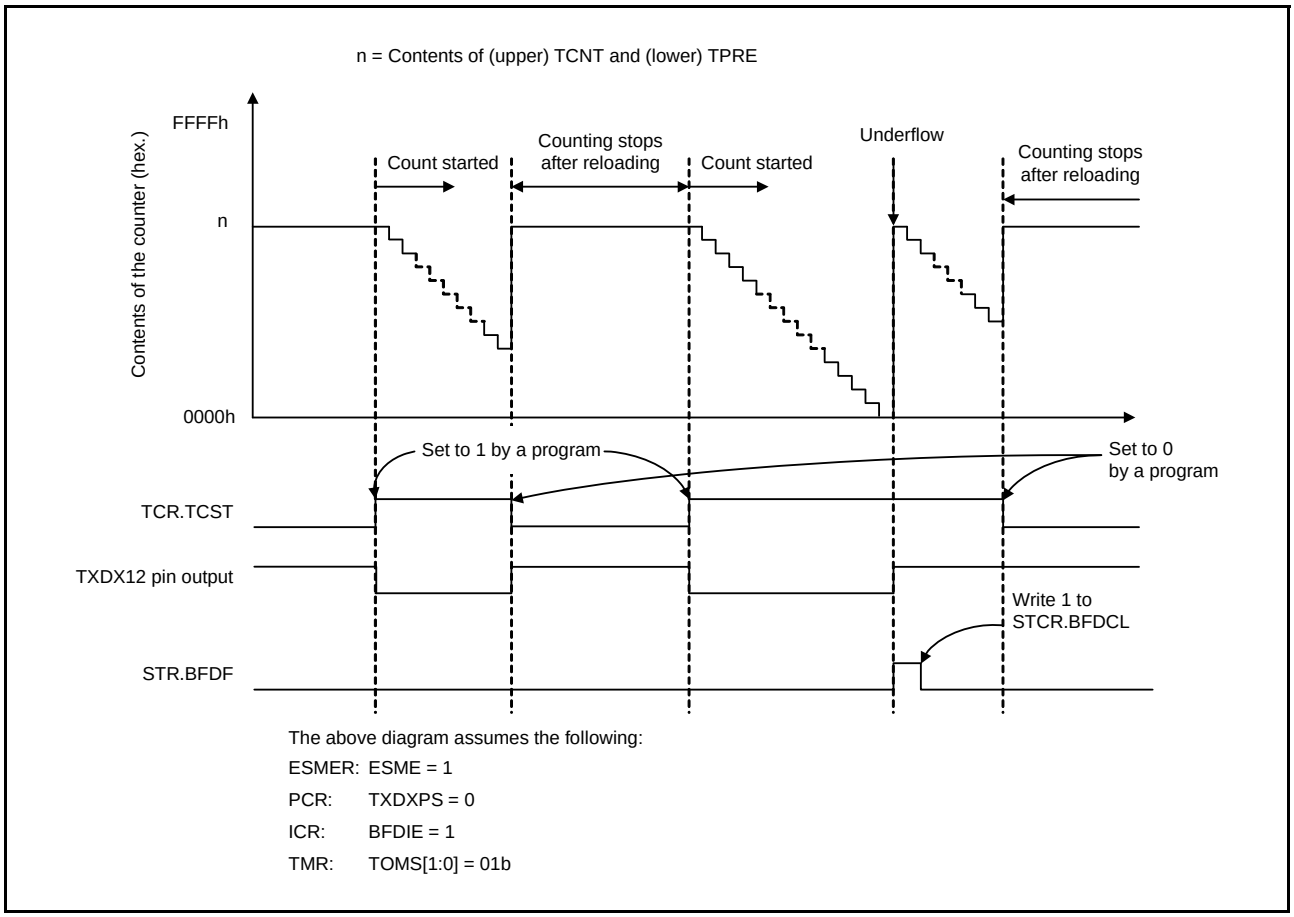


Figure 30.70 Example of Operations in Break Field Low Width Output Mode

(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 TOMS0 to 1 and TOMS1 to 0 in TMR switches operation to Break Field low width determination mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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, 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 BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR 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 30.71 shows an example of operations in Break Field low width output mode.

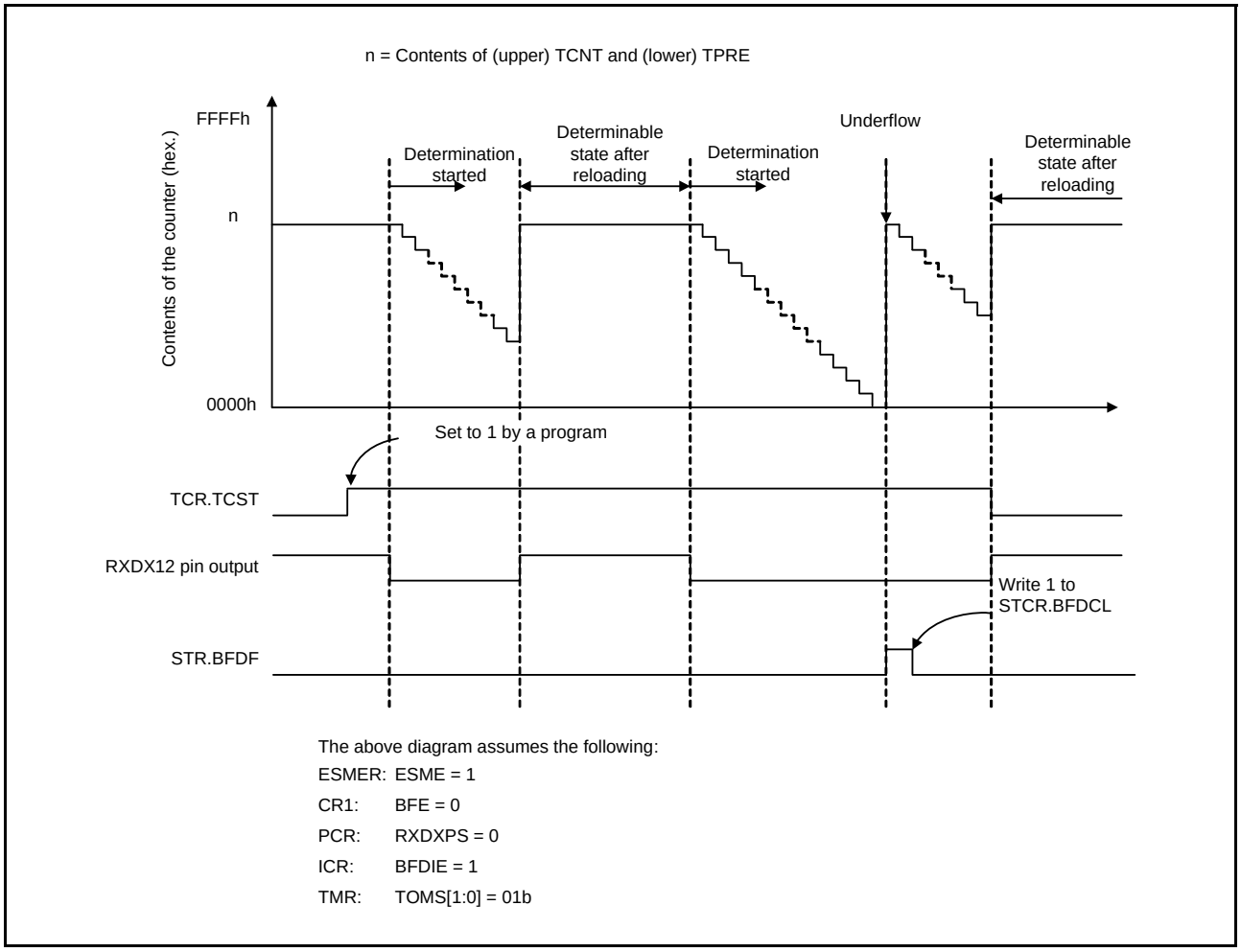


Figure 30.71 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting TOMS0 to 0 and TOMS1 to 0 in TMR switches operation to timer mode. The TCSS[2:0] bits in TMR select the clock source for the counter. Counting starts when 1 is written to the TCST bit in TCR and stops when 0 is written to TCST. TPRE and TCNT both count down. TPRE counts cycles of the clock source for counting, and underflows of TPRE provide the clock source for counting by TCNT. When the timer underflows, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.

30.10 Noise Cancellation Function

Figure 30.72 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.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

In simple I2C 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.

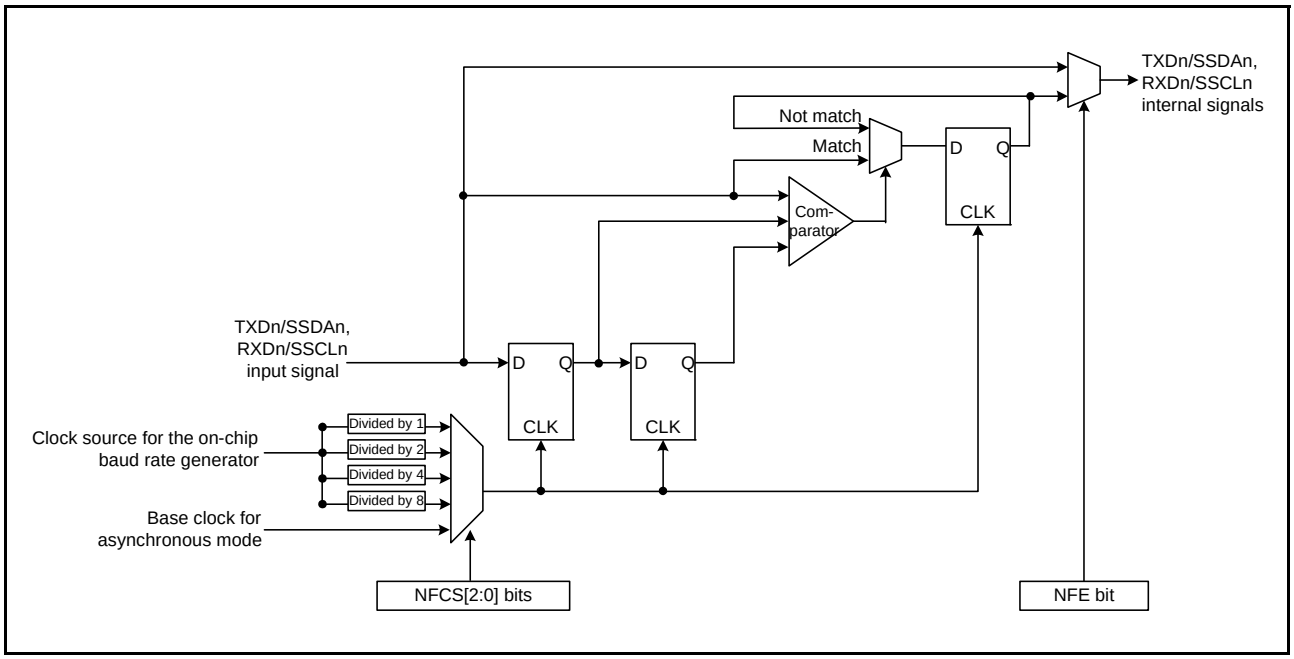


Figure 30.72 Block Diagram of Digital Noise Filter Circuit

30.11 Interrupt Sources

30.11.1 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 interrupt controller 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 interrupt controller 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) can also be used to discard an internally retained interrupt request.

30.11.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 30.29 lists interrupt sources in asynchronous 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 SCR.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR to the TSR. 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.*1

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, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR 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. 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 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. 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 transmission-completed interrupt, control prohibiting and permitting 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 30.29 Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	DMAC Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	—	Possible	Possible	
TEI	Transmit end	TEND	Not possible	Not possible	Low

30.11.3 Interrupts in Smart Card Interface Mode

Table 30.30 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 30.30 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	TEND	Possible	Possible	Low

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 TEND flag in SSR 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 ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, refer to section 17, DMA Controller (DMACA) and section 18, Data Transfer Controller (DTCa).

In reception, an RXI interrupt request is generated when receive data is set to RDR. 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.

30.11.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 30.31. 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 IICINTM bit in SIMR2 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 IICINTM bit in SIMR2 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 SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 30.31 SCI Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	DMAC Activation	Priority
RXI	Reception, ACK detection	—	Possible	Possible	High
TXI	Transmission, NACK detection	—	Possible*1	Possible*1	↑
STI	Completion of generation of a start, restart, or stop condition	IICSTIF	Not possible	Not possible	Low

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

30.11.5 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 30.32.

Table 30.32 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 CF0DR
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in PCF1DR or SCF1DR
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in PCF1DR
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

30.12 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).
- 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) 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.

30.13 Usage Notes

30.13.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) and module stop control register C (MSTPCRC) are 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.

30.13.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error has occurred), and the PER flag in SSR may also be set to 1 (parity error has occurred). 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 is set to 1 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.

30.13.3 Mark State and Production of Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (serial transmission is enabled), the I/O port function is used to set the TXDn pin to output 1 and set the pin mode to a general I/O port pin, and thus place the communication line in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output 0 and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

30.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. 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 RE bit in SCR is set to 0 (serial reception is disabled).

30.13.5 Writing Data to TDR

Data can be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request handling routine.

30.13.6 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 TDR by the CPU, DMAC, or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (see Figure 30.73).

(2) Continuous transmission

(a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (see Figure 30.73).

(b) When updating TDR after bit 7 has started to transmit, update TDR 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 (see Figure 30.73).

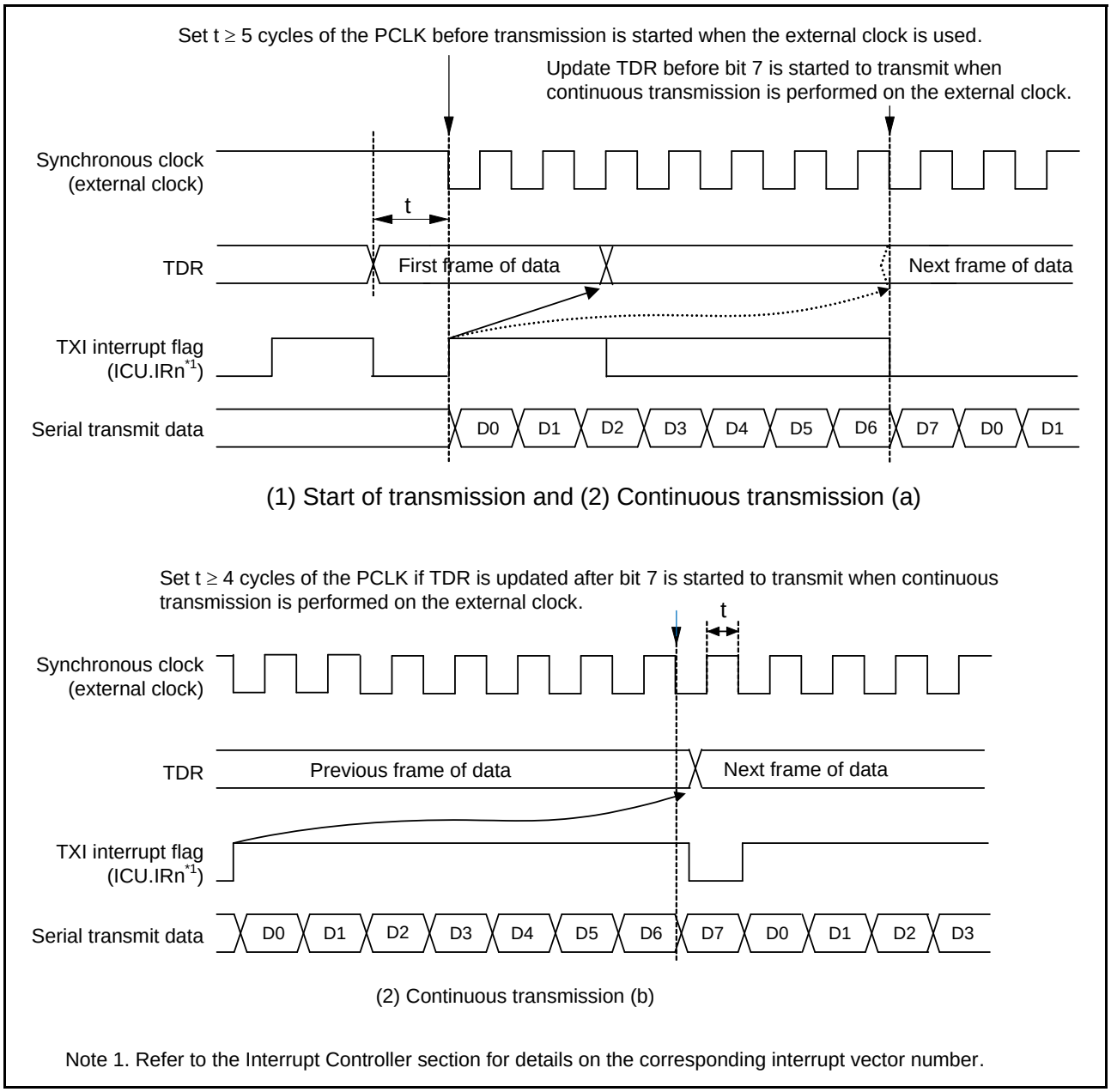


Figure 30.73 Restrictions on Use of External Clock in Clock Synchronous Transmission

30.13.7 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

30.13.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) 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 (ICUb).

- 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 bit) in the interrupt controller to 0.

30.13.9 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 to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the TEND bit in the SSR. Depending on the port settings, 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 transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

Figure 30.74 shows a sample flowchart for transition to software standby mode during transmission. Figure 30.75 and Figure 30.76 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 transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (RE = 0 in SCR). If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception 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 reception mode, initialize the SCI first.

Figure 30.77 shows a sample flowchart for transition to software standby mode during reception.

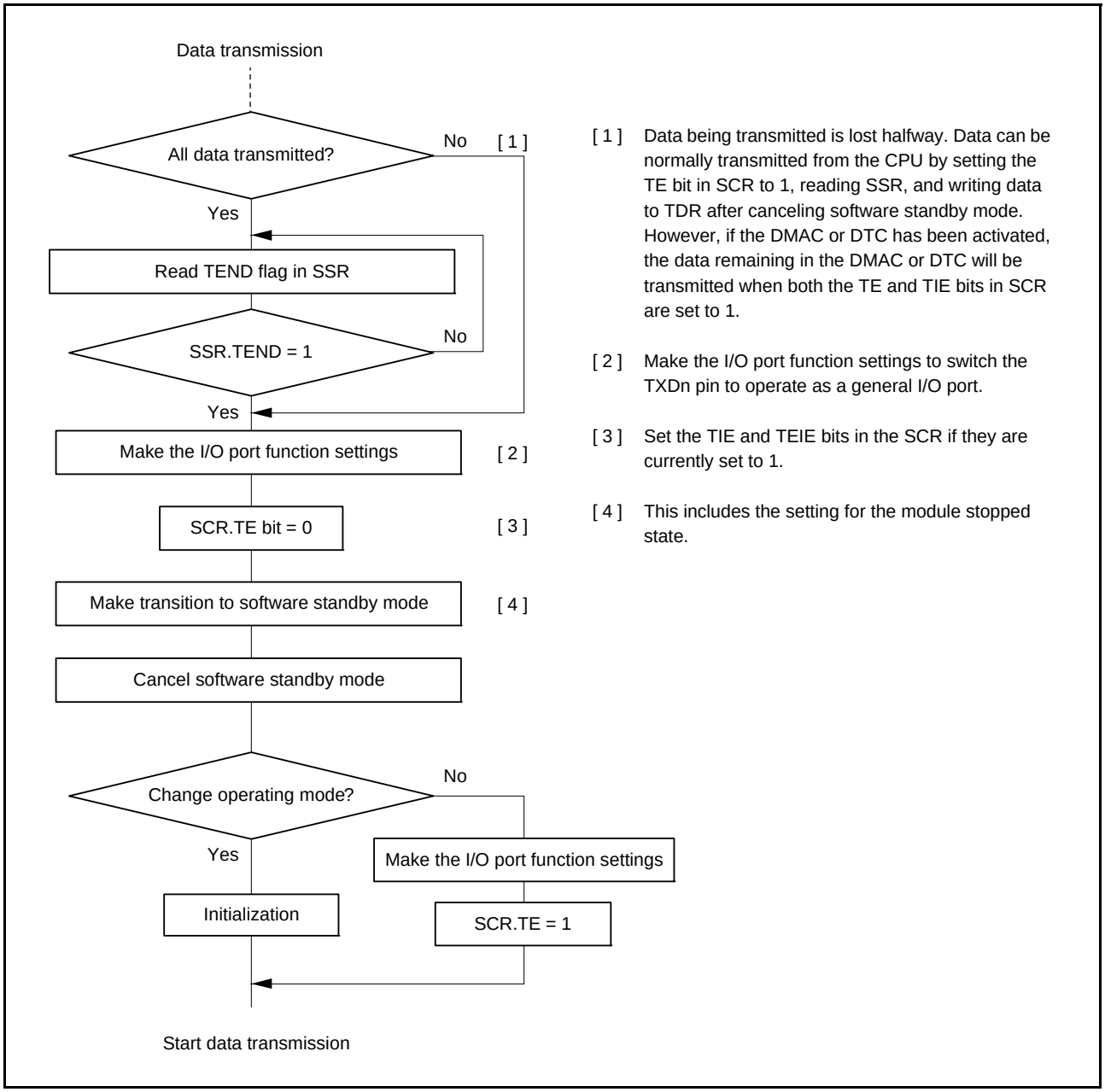


Figure 30.74 Example of Flowchart for Transition to Software Standby Mode during Transmission

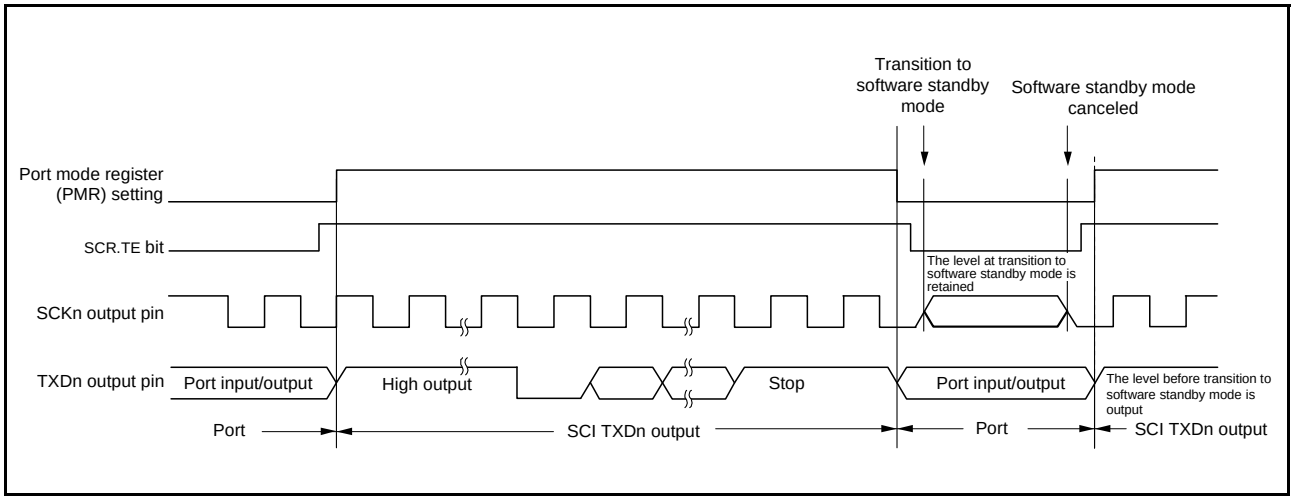


Figure 30.75 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

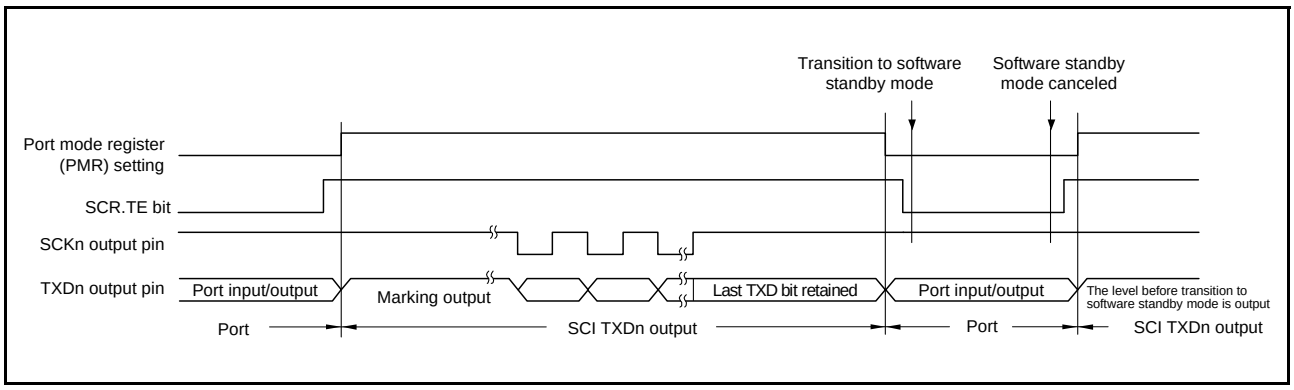


Figure 30.76 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

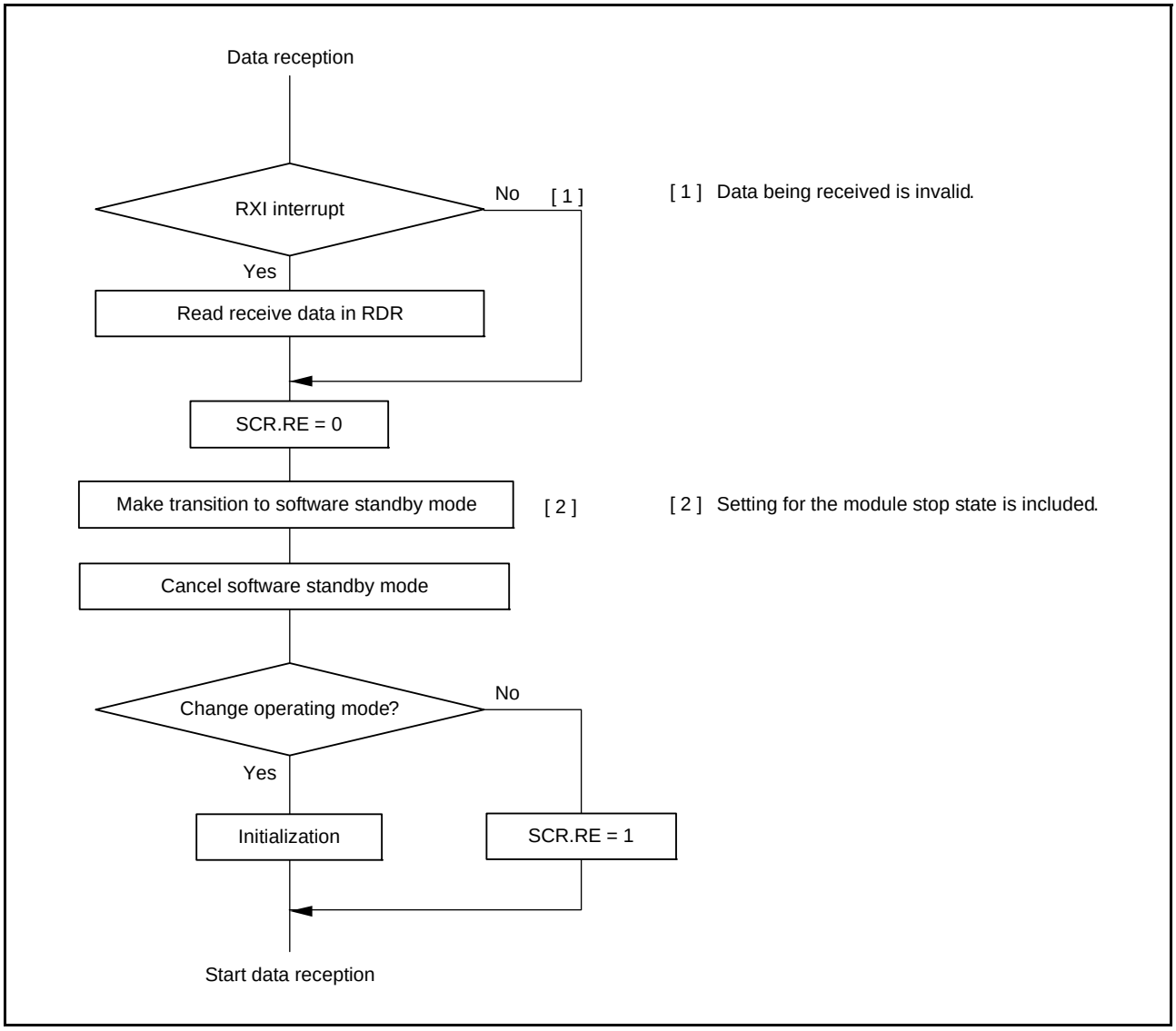


Figure 30.77 Example of Flowchart for Transition to Software Standby Mode during Reception

30.13.10 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

30.13.11 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 30.78. If the TE and RE bits in the SCR 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.

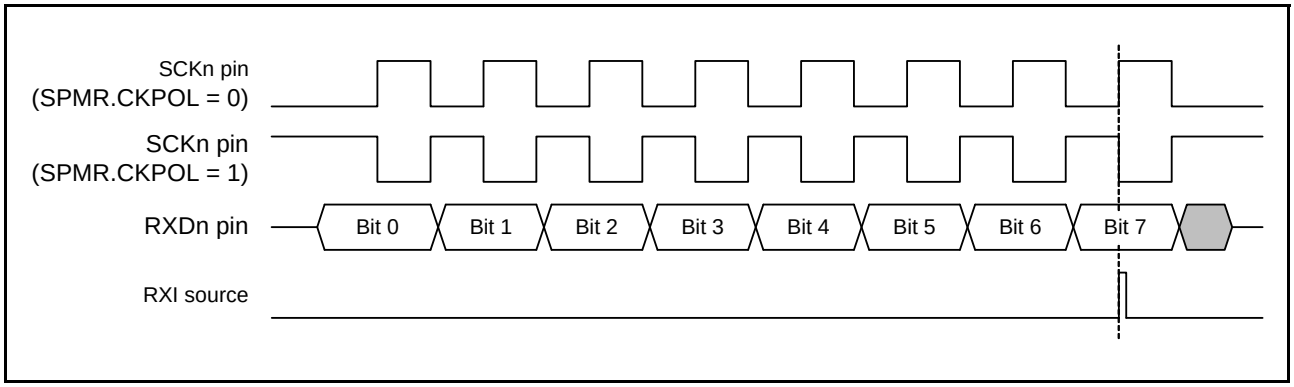


Figure 30.78 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Provide an external clock signal to the master the same as the data length for transfer.
- 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 to 0 and, after remaking the settings, restart transfer of the first byte.

30.13.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the SHARPS bit in PCR is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer of the SCIf module is in Break Field low width output mode and the value of the TCST bit in TCR 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 TE bit in SCI12.SCR is 1.

30.13.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SCIe interrupt request is generated even if the extended serial mode is enabled. However, the SCIe interrupt should not be used during reception of a Start Frame because SCIf uses an SCIe interrupt request.
The two ways of dealing with this are described below. When a receive error is detected, clear the error flag of the SCIe and initialize the control section of the SCIf.

(1) Set the SCR.RIE bit of the SCIe to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SCIe 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 of the SCIe to 1 by the time the first byte of the Information Frame is received.

(2) Set the SCR.RIE bit of the SCIe 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.

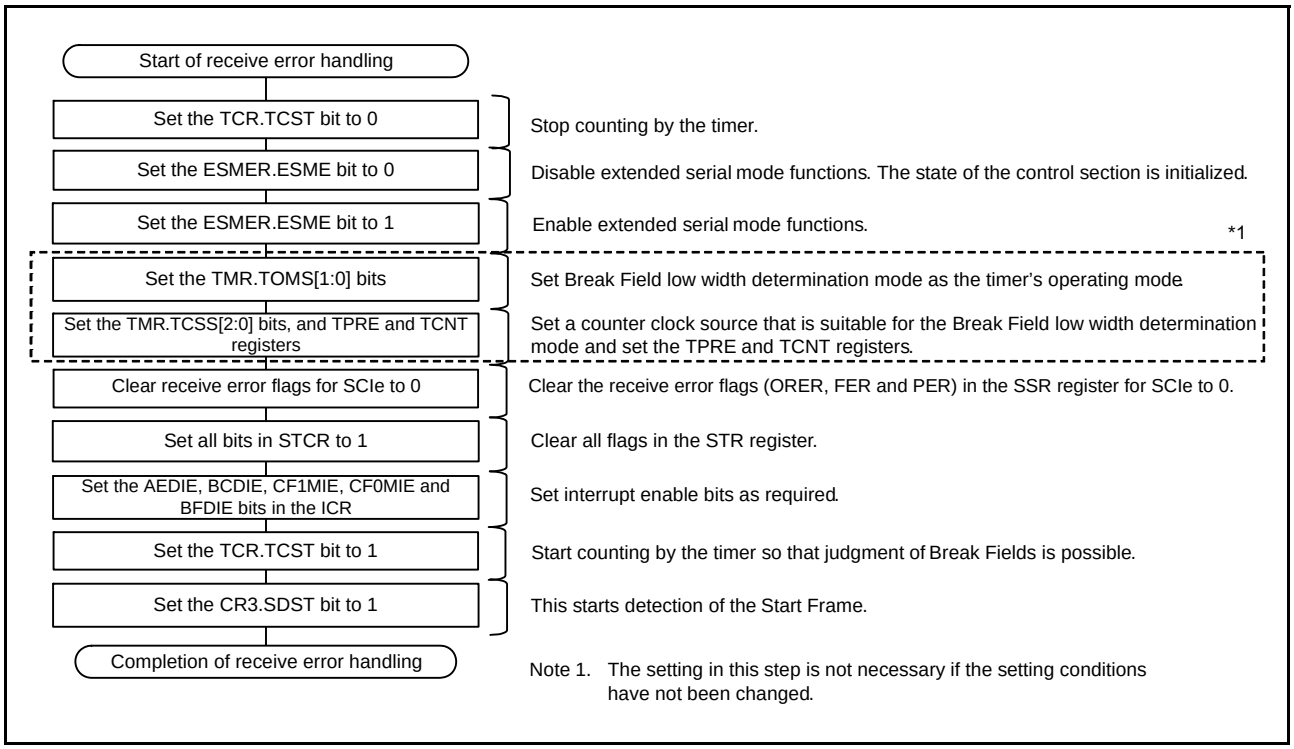


Figure 30.79 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

30.13.14 Note on Transmit Enable Bit (TE Bit)

When setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is “TXDn”, output of the pin becomes high impedance.

Prevent the TXDn line from becoming high impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.

(2) Change the pin function to “general-purpose I/O port, output” before setting the SCR.TE bit to 0.
Set the SCR.TE bit to 1 before changing the pin function to “TXDn”.

31. Remote Control Signal Receiver (RCR)

This MCU has a two-channel remote control signal receiver (RCR). The RCR can receive data by checking the width and period of an external pulse input signal.

31.1 Overview

Table 31.1 lists the RCR specifications. Figure 31.1 shows a block diagram of the RCR.

Table 31.1 RCR Specifications

Item	Description	
	RCR0	**RCR1**
External pulse input	PMC0	PMC1
Count clock sources	• RCRILCLK*1 • RCRMCLK*2 • TMR compare match output (TMO0) • PCLKB	• RCRILCLK*1 • RCRMCLK*2 • TMR compare match output (TMO2) • PCLKB
Count operation	Up-count	
Detection patterns	• Header pattern • Data 0 pattern • Data 1 pattern • Special data pattern	
Receive buffer	8 bytes (64 bits)	
Interrupt request	RCRI0	RCRI1
Interrupt request generation timing	• Compare match • Receive error • Completion of data reception • Receive buffer full • Header pattern match • Data 0 pattern or data 1 pattern match • Special data pattern match	
Selectable functions	• Input signal inversion • Digital filter (matching three or two times) • Pattern end setting	
Low power consumption	• Module stop state can be set for each channel. • Signal reception during low power consumption state and recovery from low power consumption state in response to the RCR interrupt request are available.	

Note 1. RCRILCLK is an operating clock supplied from the IWDT-dedicated on-chip oscillator.
Note 2. RCRMCLK is an operating clock supplied from the main clock oscillator.

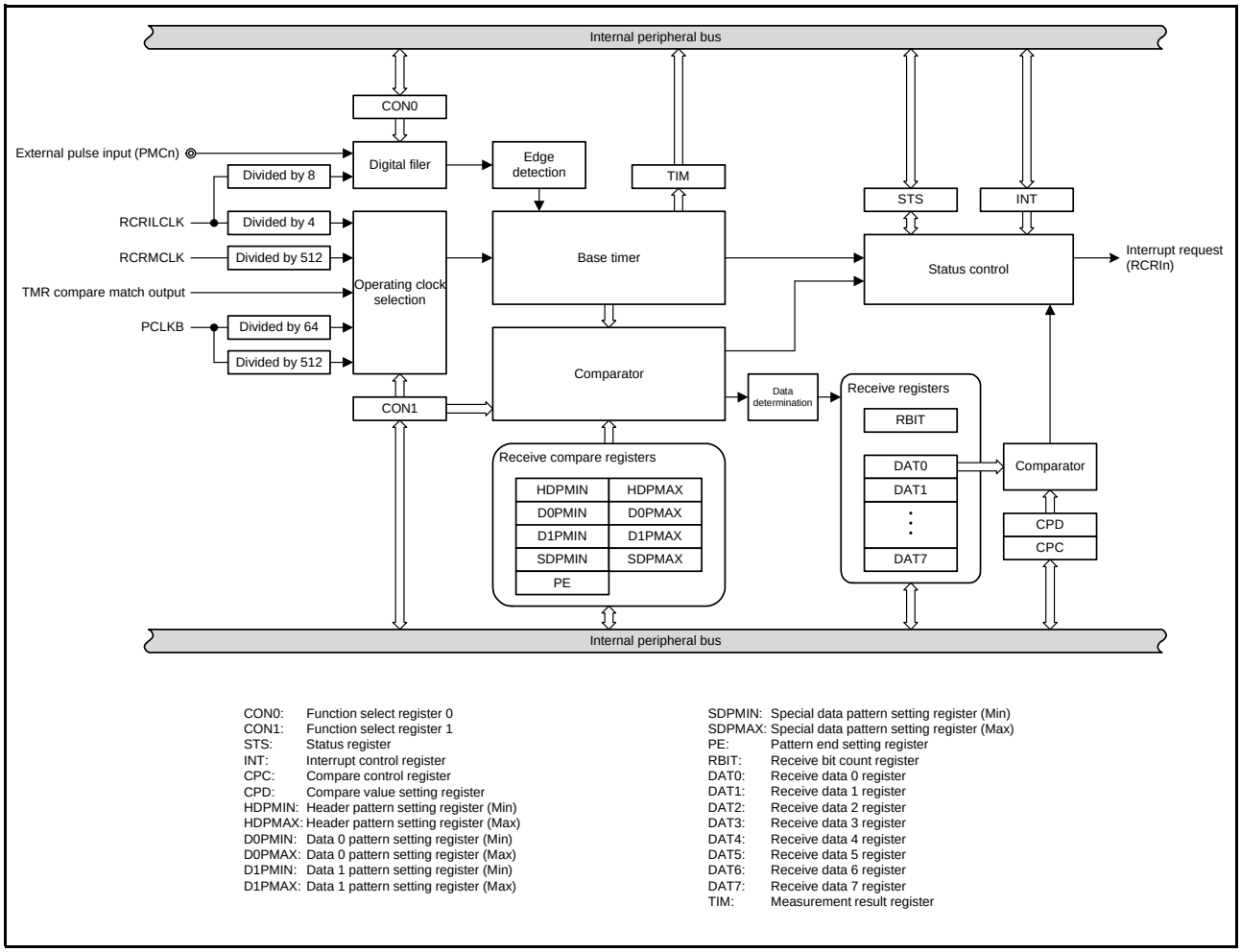


Figure 31.1 RCR Block Diagram

Table 31.2 lists the input pins used for the RCR.

Table 31.2 RCR Pin Configuration

Channel	Pin Name	I/O	Function
RCR0	PMC0	Input	External pulse signal input
RCR1	PMC1	Input	External pulse signal input

31.2 Registers

31.2.1 Function Select Register 0 (CON0)

Address(es): RCR0.CON0 000A 0B00h, RCR1.CON0 000A 0B80h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	FILSEL	—	EC	INFLG	FIL	INV	ENFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENFLG	Remote Control Status Flag	0: Stopped 1: Operating	R*2
b1	INV	Input Signal Inversion	0: Not inverted 1: Inverted	R/W*1
b2	FIL	Digital Filter Enable/Disable Setting	0: Disables the digital filter for matches three or two times. 1: Enables the digital filter for matching three or two times.	R/W*1
b3	INFLG	Input Signal Flag	0: The level of the internal input signal of the remote control signal receiver is low. 1: The level of the internal input signal of the remote control signal receiver is high.	R*2
b4	EC	Receive Error Capture Operation Select	0: Captures the data after an error pattern is received. 1: Does not capture the data after an error pattern is received.	R/W*1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	FILSEL	Digital Filter Function Select	0: Digital filter for matching three times 1: Digital filter for matching two times	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the ENFLG flag are both 0 (RCR is stopped).
Note 2. These flags are all initialized when the CON1.EN bit is 0.

ENFLG Flag (Remote Control Status Flag)

This flag can be used to confirm whether the remote control signal receiver is stopped or operating.
This flag changes after zero to one clock when a value is written to the CON1.EN bit.

FIL Bit (Digital Filter Enable/Disable Setting)

This bit enables or disables the digital filter. To enable the digital filter, it is necessary to supply the RCRILCLK used as the operating clock for the digital filter. For details on supplying the operating clock, refer to section 31.3.3, Operating Clocks.

INFLG Flag (Input Signal Flag)

This flag can be used to confirm the level of the internal input signal of the remote control signal receiver. The confirmed level is the result set by the INV and FIL bits.

EC Bit (Receive Error Capture Operation Select)

This bit can be used to set capture operation to the RBIT and DATj registers (j = 0 to 7) after an error pattern is received.

31.2.2 Function Select Register 1 (CON1)

Address(es): RCR0.CON1 000A 0B01h, RCR1.CON1 000A 0B81h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CSRC[2:0]			—	EN	TYP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TYP[1:0]	Receive Mode Select	These bits can be used to select the format for capturing the remote control signal waveform. b1 b0 0 0: Format A shown in section 31.3.2, Pattern Setting. 0 1: Format B shown in section 31.3.2, Pattern Setting. 1 0: Format C shown in section 31.3.2, Pattern Setting. 1 1: Setting prohibited	R/W*1
b2	EN	Remote Control	0: Operation disabled 1: Operation enabled	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSRC[2:0]	Count Source Clock Select	b6 b4 x 0 0: RCRILCLK/4 x 0 1: TMR compare match output x 1 0: RCRMCLK/512 0 1 1: PCLKB/64 1 1 1: PCLKB/512	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

x: Don't care

Note 1. To rewrite the TYP[1:0] bits when the EN bit or CON0.ENFLG flag is 1 (RCR is operating), change the values of these bits one bit at a time.

EN Bit (Remote Control)

This bit enables or disables RCR operation.

Use the CON0.ENFLG flag to confirm whether operation has started or not.

CSRC[2:0] Bits (Count Source Clock Select)

These bits select the operating clock source for the RCR. These bits can be rewritten when the EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.3 Status Register (STS)

Address(es): RCR0.STS 000A 0B02h, RCR1.STS 000A 0B82h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDFLG	D1FLG	D0FLG	HDFLG	BFULFLG	DRFLG	REFLG	CPFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPFLG	Compare Match Flag	0: Mismatch 1: Match	R
b1	REFLG	Receive Error Flag*2	0: No error has occurred. 1: An error has occurred.	R
b2	DRFLG	Data Receiving Flag	0: Waiting for data reception. 1: Data is being received.	R
b3	BFULFLG	Receive Buffer Full Flag	0: Receive buffer is empty. 1: Receive buffer is full (64 bits received).	R/W*1
b4	HDFLG	Header Pattern Match Flag	0: Mismatch 1: Match	R
b5	D0FLG	Data 0 Pattern Match Flag	0: Mismatch 1: Match	R
b6	D1FLG	Data 1 Pattern Match Flag	0: Mismatch 1: Match	R
b7	SDFLG	Special Data Pattern Match Flag	0: Mismatch 1: Match	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 31.4.8, Reading Registers.
Note: The values of the STS register are initialized when the CON1.EN bit is 0.
Note 1. The flag can be cleared only by writing 0. However, if this flag is written when changing the CON0.INFLG flag, the value read from this flag may become undefined.
Note 2. The conditions for changing the REFLG flag differ depending on the CON1.TYP[1:0] bits (receive format select). Table 31.3 lists the conditions for changing the REFLG flag.

CPFLG Flag (Compare Match Flag)

This flag indicates the comparison result between the value of the CPD register specified by the CPC.CPN bit and the data to be stored in DAT0.

[Setting condition]

- When the value of the CPD register matches the value to be stored in the DAT0 register (when the setting value of the CPC.CPN[2:0] bits is n, bits n to 0 in the CPD register match bits n to 0 in the DAT0 register)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the HDFLG flag changes from 0 to 1

DRFLG Flag (Data Receiving Flag)

This flag indicates the state of receiving the remote control signal.

[Setting condition]

- Rising edge of RCR internal input signal (when the CON0.INV bit is 0)

[Clearing condition]

- This flag becomes 0 after one cycle of the count source when the value of the base timer is larger than any value of the HDPMAX, D0PMAX, D1PMAX, SDPMAX, and PE register.

BFULFLG Flag (Receive Buffer Full Flag)

[Setting condition]

- When the value of the RBIT register becomes 64

[Clearing conditions]

- When the HDFLG flag changes from 0 to 1
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- This flag becomes 0 after one to two cycles when 0 is written to the BFULFLG flag.

HDFLG Flag (Header Pattern Match Flag)

[Setting condition]

- See Table 31.4, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 31.4, Measurement Results and Flags.

D0FLG Flag (Data 0 Pattern Match Flag)

[Setting condition]

- See Table 31.4, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 31.4, Measurement Results and Flags.

D1FLG Flag (Data 1 Pattern Match Flag)

[Setting condition]

- See Table 31.4, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 31.4, Measurement Results and Flags.

SDFLG Flag (Special Data Pattern Match Flag)

[Setting condition]

- See Table 31.4, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 31.4, Measurement Results and Flags.

Table 31.3 Conditions for Changing REFLG Flag

CON1.TYP[1:0]	Conditions for Changing REFLG Flag to 1
00b	• The data 0, data 1, or special data pattern is detected prior to receiving the header pattern • The width between a rising edge and the next rising edge of the input signal is not the data 0, data 1, or special data pattern (when the CON0.INV bit is 0) • A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes
01b	• The data 0, data 1, or special data pattern is detected prior to receiving the header pattern • The width between a falling edge and the next falling edge of the input signal is not the data 0, data 1, or special data pattern (when the CON0.INV bit is 0) • A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes
10b	• The width between a rising edge and the next rising edge of the input signal is not the data 0, data 1, or special data pattern (when the CON0.INV bit is 0) • A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes
11b	Setting prohibited
CON1.TYP[1:0]	**Conditions for Changing REFLG Flag to 0**
00b	• The header pattern is detected • The DRFLG flag changes from 0 to 1 (next frame reception starts)
01b	
10b	
11b	Setting prohibited

Table 31.4 Measurement Results and Flags

Comparison Result between TIM Register Value (Measurement Result) and Each Register	Flag Value			
	HDFLG	D0FLG	D1FLG	SDFLG
Between HDPMIN and HDPMAX	1	0	0	0
Between D0PMIN and D0PMAX	0	1*1	0	0
Between D1PMIN and D1PMAX	0	0	1*1	0
Between SDPMIN and SDPMAX	0	0	0	1*1
Values not listed above	0	0	0	0

Note 1. When the CON1.TYP[1:0] bits are 00b or 01b, the D0FLG, D1FLG, and SDFLG flags remain unchanged until the header pattern is detected.

31.2.4 Interrupt Control Register (INT)

Address(es): RCR0.INT 000A 0B03h, RCR1.INT 000A 0B83h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDINT	—	DINT	HDINT	BFULINT	DRINT	REINT	CPINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPINT	Compare Match Interrupt Enable	0: Disabled 1: Enabled	R/W*1
b1	REINT	Receive Error Interrupt Enable	0: Disabled 1: Enabled	R/W*1
b2	DRINT	Data Reception Complete Interrupt Enable	0: Disabled 1: Enabled	R/W
b3	BFULINT	Receive Buffer Full Interrupt Enable	0: Disabled 1: Enabled	R/W*1
b4	HDINT	Header Pattern Match Interrupt Enable	0: Disabled 1: Enabled	R/W*1
b5	DINT	Data 0 Pattern or Data 0 Pattern Match Interrupt Enable	0: Disabled 1: Enabled	R/W
b6	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b7	SDINT	Special Data Pattern Match Interrupt Enable	0: Disabled 1: Enabled	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.5 Compare Control Register (CPC)

Address(es): RCR0.CPC 000A 0B04h, RCR1.CPC 000A 0B84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CPN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CPN[2:0]	Compare Bit Count Specification	When the setting value of the CPN[2:0] bits is n, bits n to 0 are compared. Example 1) Setting value: 0 Bit 0 in the CPD register and bit 0 in the DAT0 register are compared Example 2) Setting value: 7 Bits 7 to 0 in the CPD register and bits 7 to 0 in the DAT0 register are compared	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.6 Compare Value Setting Register (CPD)

Address(es): RCR0.CPD 000A 0B05h, RCR1.CPD 000A 0B85h

	b7	b6	b5	b4	b3	b2	b1	b0
	CPD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CPD[7:0]	Compare Value Setting	Set the value to be compared with the data in the DAT0 register when the compare function is used. The CPC.CPN[2:0] bits can be used to set the number of bits to be compared.	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.7 Header Pattern Setting Register (Min) (HDPMIN)

Address(es): RCR0.HDPMIN 000A 0B06h, RCR1.HDPMIN 000A 0B86h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HDPMIN[10:0]										
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
b10 to b0	HDPMIN[10:0]	Header Pattern Minimum Width Setting	Set the minimum width of header pattern. Setting range: 000h to 7FFh	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.8 Header Pattern Setting Register (Max) (HDPMAX)

Address(es): RCR0.HDPMAX 000A 0B08h, RCR1.HDPMAX 000A 0B88h

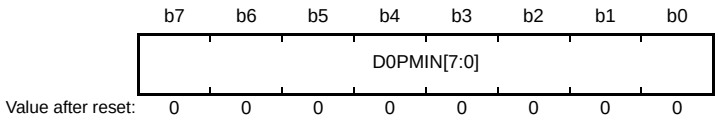
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HDPMAX[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	HDPMAX[10:0]	Header Pattern Maximum Width Setting	Set the maximum width of header pattern. Setting range: 000h to 7FFh	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.9 Data 0 Pattern Setting Register (Min) (D0PMIN)

Address(es): RCR0.D0PMIN 000A 0B0Ah, RCR1.D0PMIN 000A 0B8Ah

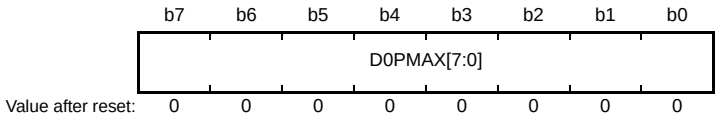


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMIN[7:0]	Data 0 Pattern Minimum Width Setting	Set the minimum width of data 0 pattern. Setting range: 00h to FFh	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.10 Data 0 Pattern Setting Register (Max) (D0PMAX)

Address(es): RCR0.D0PMAX 000A 0B0Bh, RCR1.D0PMAX 000A 0B8Bh

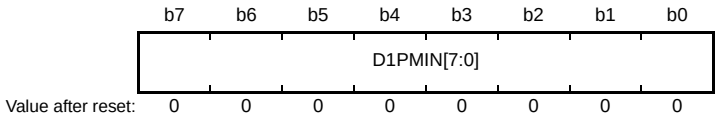


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMAX[7:0]	Data 0 Pattern Maximum Width Setting	Set the maximum width of data 0 pattern. Setting range: 00h to FFh	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.11 Data 1 Pattern Setting Register (Min) (D1PMIN)

Address(es): RCR0.D1PMIN 000A 0B0Ch, RCR1.D1PMIN 000A 0B8Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D1PMIN[7:0]	Data 1 Pattern Minimum Width Setting	Set the minimum width of data 1 pattern. Setting range: 00h to FFh	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.12 Data 1 Pattern Setting Register (Max) (D1PMAX)

Address(es): RCR0.D1PMAX 000A 0B0Dh, RCR1.D1PMAX 000A 0B8Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	D1PMAX[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D1PMAX[7:0]	Data 1 Pattern Maximum Width Setting	Set the maximum width of data 1 pattern. Setting range: 00h to FFh	R/W*1

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.13 Special Data Pattern Setting Register (Min) (SDPMIN)

Address(es): RCR0.SDPMIN 000A 0B0Eh, RCR1.SDPMIN 000A 0B8Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDPMIN[10:0]										
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
b10 to b0	SDPMIN[10:0]	Special Data Pattern Minimum Width Setting	Set the minimum width of special data pattern. Setting range: 000h to 7FFh	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.14 Special Data Pattern Setting Register (Max) (SDPMAX)

Address(es): RCR0.SDPMAX 000A 0B10h, RCR1.SDPMAX 000A 0B90h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDPMAX[10:0]										
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
b10 to b0	SDPMAX[10:0]	Special Data Pattern Maximum Width Setting	Set the maximum width of special data pattern. Setting range: 000h to 7FFh	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.15 Pattern End Setting Register (PE)

Address(es): RCR0.PE 000A 0B12h, RCR1.PE 000A 0B92h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PE[10:0]										
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
b10 to b0	PE[10:0]	Pattern End Width Setting	Set the width of pattern end. Setting range: 000h to 7FFh These bits can be used to set the timing at which the STS.DRFLG flag changes from 1 to 0.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the CON1.EN bit and the CON0.ENFLG flag are both 0 (RCR is stopped).

31.2.16 Receive Bit Count Register (RBIT)

Address(es): RCR0.RBIT 000A 0B15h, RCR1.RBIT 000A 0B95h

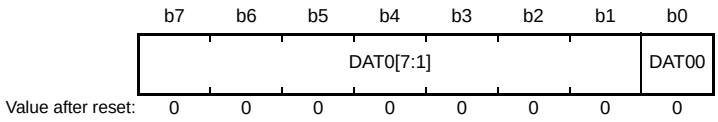
	b7	b6	b5	b4	b3	b2	b1	b0
	—	RBIT[6:1]						RBIT0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RBIT0	Receive Bit Count Check 0	Receive bit count can be read.	R/W
b6 to b1	RBIT[6:1]	Receive Bit Count Check 6 to 1	These bits indicate the bit position of the buffer to be stored by counting the detected data 0 pattern or data 1 pattern. • When the receive bit count exceeds 64, the value returns to 1. • The header pattern and special data pattern are not counted. • If an error is detected while the CON0.EC bit is 1, the value is not incremented even when the data 0 pattern or data 1 pattern is detected. • The RBIT register becomes 00h when the STS.DRFLG flag changes from 0 to 1. • The RBIT register becomes 00h when the STS.HDFLG flag changes from 0 to 1. When 0 is written to the RBIT.RBIT0 bit, the value of the RBIT register becomes 00h after one to two cycles of the operating clock.	R
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 31.4.8, Reading Registers.
Note: The values of this register are initialized when the CON1.EN bit is 0.

31.2.17 Receive Data 0 Register (DAT0)

Address(es): RCR0.DAT0 000A 0B16h, RCR1.DAT0 000A 0B96h

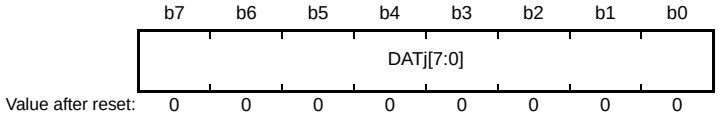


Bit	Symbol	Bit Name	Description	R/W
b0	DAT00	Receive Data 0 Store Bit 0	Receive data is stored. The values of the DAT0 to DAT7 registers become all 00h after one to two cycles of the operating clock when 0 is written to bit 0 in the DAT0 register.	R/W
b7 to b1	DAT0[7:1]	Receive Data 0 Store Bits 7 to 1		R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 31.4.8, Reading Registers.
Note: The values of this register are initialized when the CON1.EN bit is 0.

31.2.18 Receive Data j Register (DATj) (j = 1 to 7)

Address(es): RCR0.DAT1 000A 0B17h, RCR0.DAT2 000A 0B18h, RCR0.DAT3 000A 0B19h, RCR0.DAT4 000A 0B1Ah, RCR0.DAT5 000A 0B1Bh, RCR0.DAT6 000A 0B1Ch, RCR0.DAT7 000A 0B1Dh, RCR1.DAT1 000A 0B97h, RCR1.DAT2 000A 0B98h, RCR1.DAT3 000A 0B99h, RCR1.DAT4 000A 0B9Ah, RCR1.DAT5 000A 0B9Bh, RCR1.DAT6 000A 0B9Ch, RCR1.DAT7 000A 0B9Dh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATj[7:0]	Receive Data j Store	Receive data is stored.	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 31.4.8, Reading Registers.
Note: The values of this register are initialized when the CON1.EN bit is 0.

When data 0 pattern or data 1 pattern is detected, the result is stored bit by bit as received data. For details on storing received data, see section 31.3.7, Receive Data Buffer.

31.2.19 Measurement Result Register (TIM)

Address(es): RCR0.TIM 000A 0B1Eh, RCR1.TIM 000A 0B9Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TIM[10:0]										
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
b10 to b0	TIM[10:0]	Measurement Result	The measurement result of each event width can be read. The value of the base timer is captured when one of the following events occur. • Header pattern detection • Data 0 pattern detection • Data 1 pattern detection • Special data pattern detection • Data pattern detection other than the above (receive error)	R
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 31.4.8, Reading Registers.

Note: The values of this register are initialized when the CON1.EN bit is 0.

31.3 Operation

31.3.1 Overview of RCR Operation

Figure 31.2 shows an example of the remote control signal. The signal begins with a header, followed by a sequence of data. This header differs from the subsequent sequence of data in waveform, allowing the header and the data to be distinguished. The sequence of data contains custom code and data code, and 0 or 1 is distinguished depending on the bit length. After a stop bit, there is an interval during which the signal does not change (frame space), thus constituting a frame.

The time between the edges of the external input signal is measured using the base timer in the RCR. The patterns of the remote control signal are detected and the data is captured according to the measurement results.

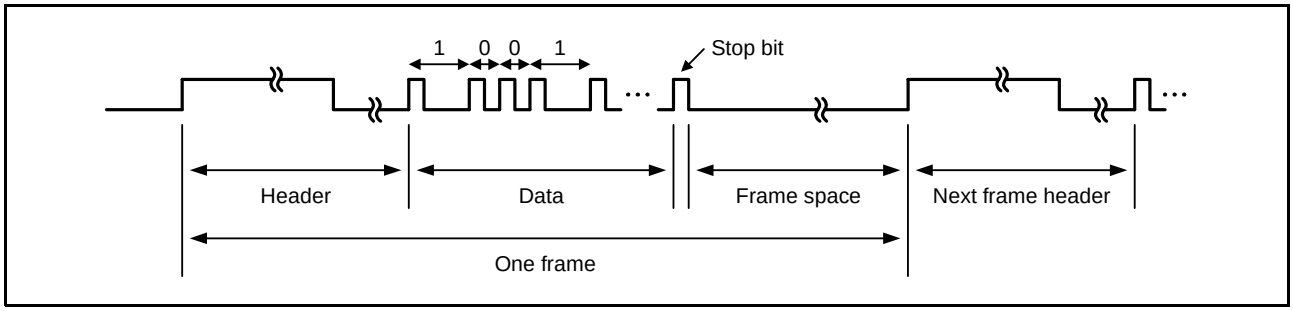


Figure 31.2 Example of Remote Control Signal

31.3.2 Pattern Setting

The format for capturing the remote control signal reception waveform can be set by setting the CON1.TYP[1:0] bits. Figure 31.3 and Figure 31.4 show examples of a remote control signal reception waveform captured by setting the CON1.TYP[1:0] bits.

When the CON1.TYP[1:0] bits are 00b (format A)

The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal.

When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the rising edge of the internal input signal.

When the CON1.TYP[1:0] bits are 01b (format B)

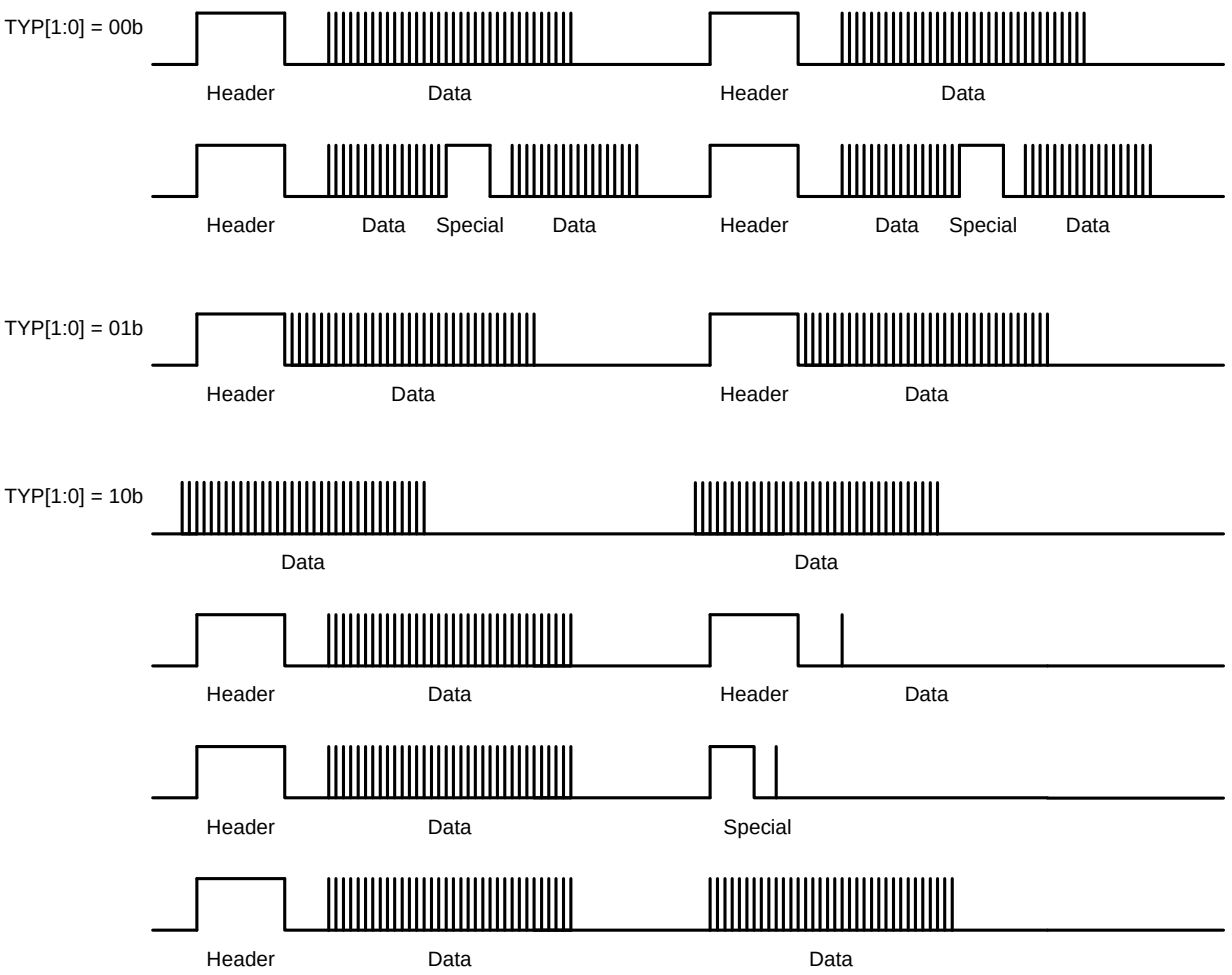
The measured result is determined from the setting value of the header pattern at the falling edge of the internal input signal.

When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the falling edge of the internal input signal.

The header pattern is detected once within one frame.

When the CON1.TYP[1:0] bits are 10b (format C)

The measured result is determined from the setting values of the header, data 0, data 1 and special data patterns at the rising edge of the internal input signal.



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

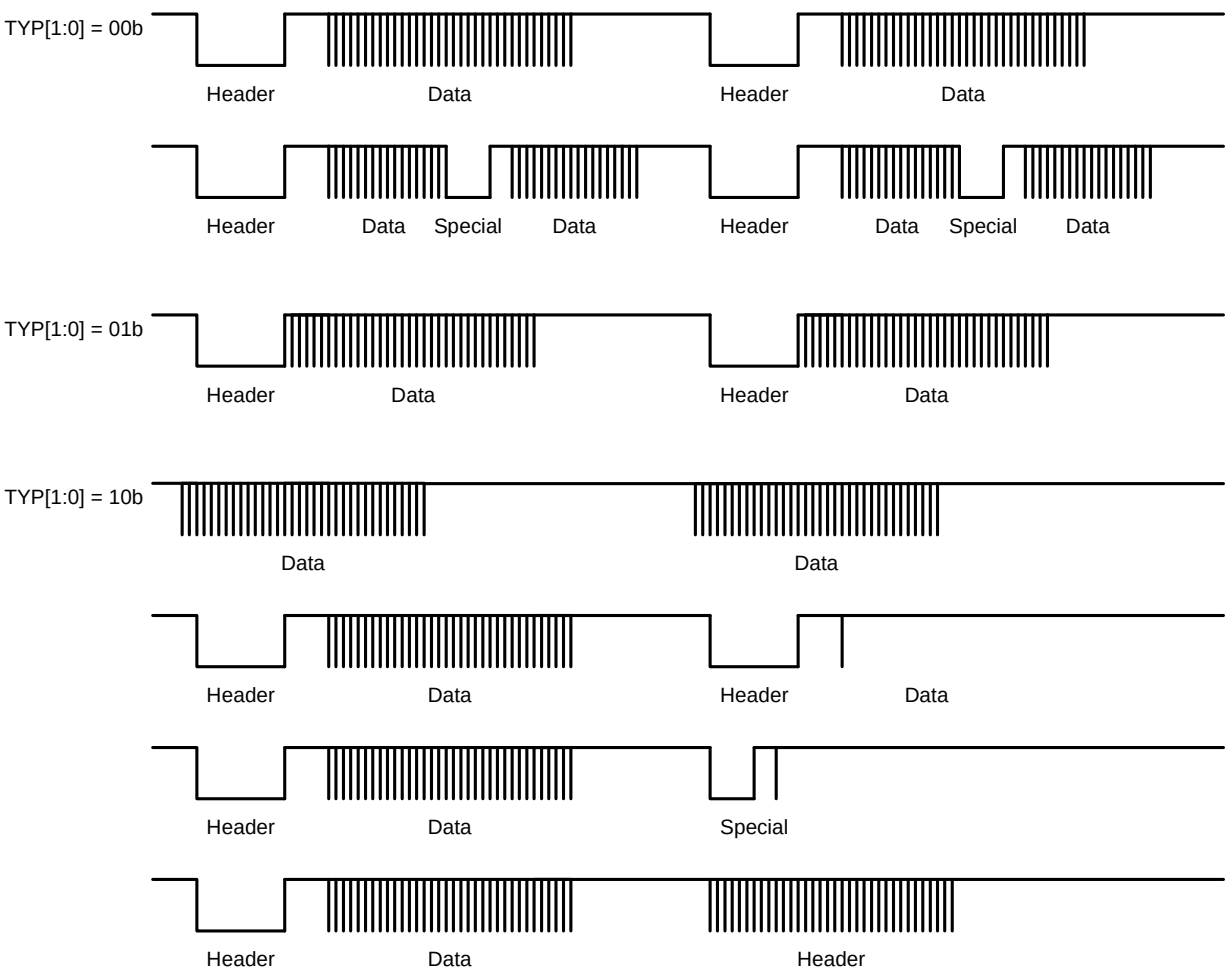
TYP[1:0]: Bits in the CON1 register

↑↓ · · · Measurement starts

↑↓ · · · Pattern measurement

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the CON0.INV bit is 0.

Figure 31.3 Example of Remote Control Signal Reception Waveform Captured by Setting CON1.TYP[1:0] Bits (CON0.INV = 0)



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

TYP[1:0]: Bits in the CON1 register

↑↓ · · · Measurement starts

↑↓ · · · Pattern measurement

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one
The above diagram applies when the CON0.INV bit is 1.

Figure 31.4 Example of Remote Control Signal Reception Waveform Captured by Setting CON1.TYP[1:0] Bits (CON0.INV = 1)

31.3.3 Operating Clocks

The RCR can use one of the following clocks as its operating clock: the divided clock of the peripheral module clock (PCLKB), the divided clock of the RCRILCLK supplied from the IWDT-dedicated on-chip oscillator, the divided clock of the RCRMCLK supplied from the main clock oscillator, or TMR compare match output.
To enable the digital filter, it is necessary to supply the RCRILCLK used as the operating clock. When supplying the RCRILCLK or RCRMCLK to the RCR, take note of the respective procedures for supplying these clocks. The following describes how to supply these clocks.

31.3.3.1 Using RCRILCLK as RCR Operating Clock

This section describes the flow for using the divided clock of the RCRILCLK supplied from the IWDT-dedicated on-chip oscillator as the RCR operating clock.
When the ILOCOCR.ILCSTP bit is set to 0, the IWDT-dedicated on-chip oscillator starts operating. After oscillation starts, the operating clock is supplied to the RCR when the oscillation stabilization wait time has elapsed. When continuing operation of the IWDT-dedicated on-chip oscillator in software standby mode, set the IWDTCSTPR.SLCSTP bit to 0.
For details on the ILOCOCR register, see section 9.2.8, IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR). For details on the IWDTCSTPR register, see section 29.2.5, IWDT Count Stop Control Register (IWDTCSTPR).
Note that the operating clock is also supplied to the IWDT while the IWDT-dedicated on-chip oscillator is operating. When using the divided clock of the RCRILCLK as the RCR operating clock, do not use the IWDT function in order to prevent an unexpected reset or an interrupt from being generated.
Figure 31.5 shows an example of the flowchart for starting the RCRILCLK supply to the RCR.

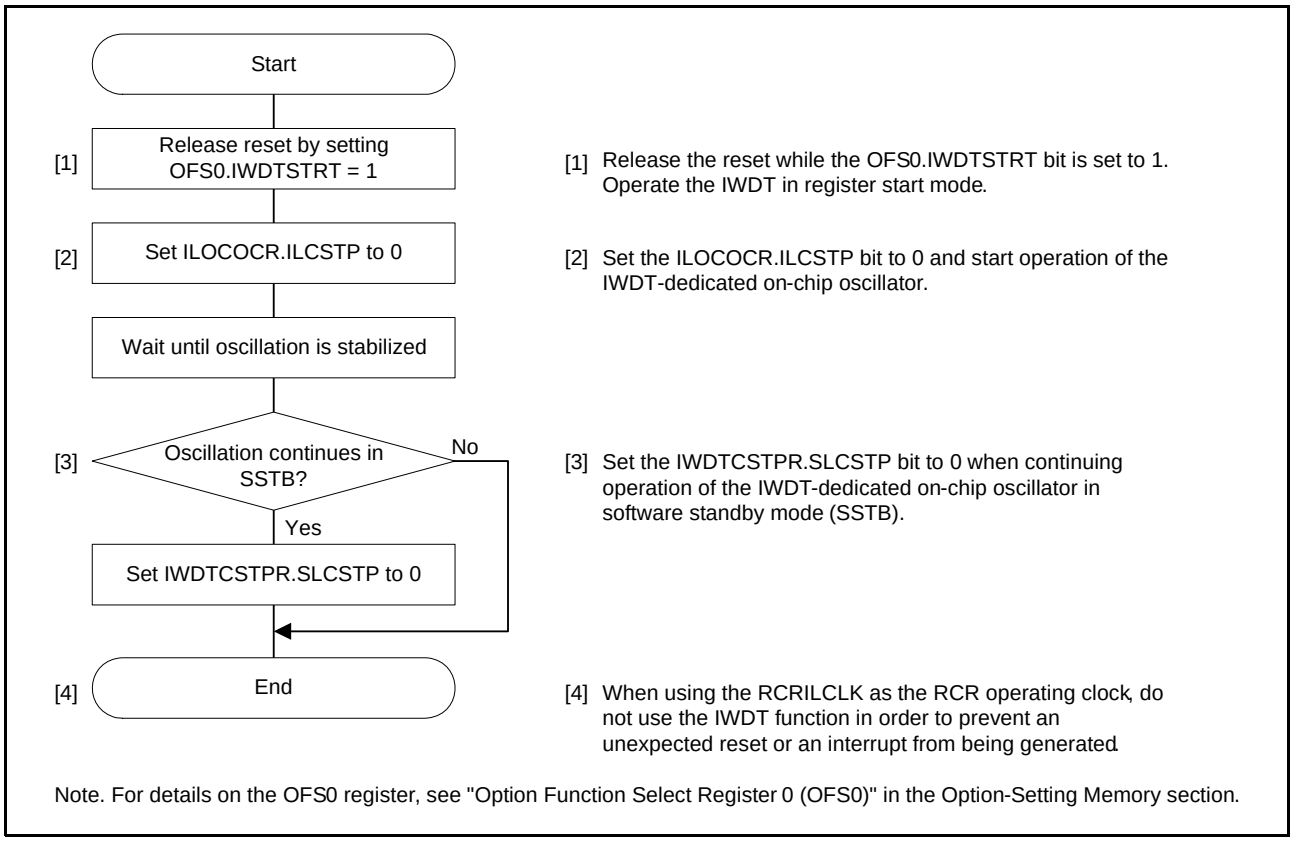


Figure 31.5 Example of Flowchart for Starting RCRILCLK Supply to RCR

31.3.3.2 When Using RCRMCLK as RCR Operating Clock

This section describes the flow for using the divided clock of the RCRMCLK supplied from the main clock oscillator as the RCR operating clock.

When the MOSCCR.MOSTP bit is set to 0, the main clock oscillator starts operating.

Although oscillation can also be started by setting the MOFCR.MOFXIN bit to 1, if this bit is used to start oscillation, the main clock cannot be supplied to the system clock. Therefore, make sure to use the MOSCCR.MOSTP bit to start oscillation. The MOFCR.MOFXIN bit should be set only to continue oscillation in software standby mode. For details on the MOSCCR register, see section 9.2.6, Main Clock Oscillator Control Register (MOSCCR). For details on the MOFCR register, see section 9.2.11, Main Clock Oscillator Forced Oscillation Control Register (MOFCR).

After the main clock oscillator starts operating, the main clock supply to the RCR is enabled when the main clock is counted for the cycles set by MOSCWTCR.MSTS[4:0]. Also, the main clock supply to the system is enabled after the main clock is counted for an additional 16,384 cycles. For details on the MOSCWTCR register, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR).

This waiting operation for oscillation stabilization is the same for the returning operation when oscillation is stopped in software standby mode.

After the main clock oscillation stabilization wait time has elapsed and the main clock supply to the RCR is enabled, set the MOSCR.MOSE bit to 1 to start the main clock supply. For details on the MOSCR register, see section 9.2.12, Main Clock Supply Control Register (MOSCR).

A noise filter is included in the path for supplying the main clock to the RCR. This noise filter can be enabled or disabled by the MONFCR register. For details on the register, see section 9.2.13, Main Clock Noise Filter Control Register (MONFCR). When modifying the MONFCR register, make sure that the MOSCR.MOSE bit is 0 (main clock supply to RCR is stopped).

To continue operation of the main clock oscillator in software standby mode, set the MOFCR.MOFXIN bit to 1 when the main clock is oscillating while the MOSCCR.MOSTP bit is set to 0 (the main clock supply to the RCR is stopped).

Figure 31.6 shows an example of the flowchart for starting the RCRMCLK supply to the RCR, and Figure 31.7 shows the path for supplying the main clock to the RCR.

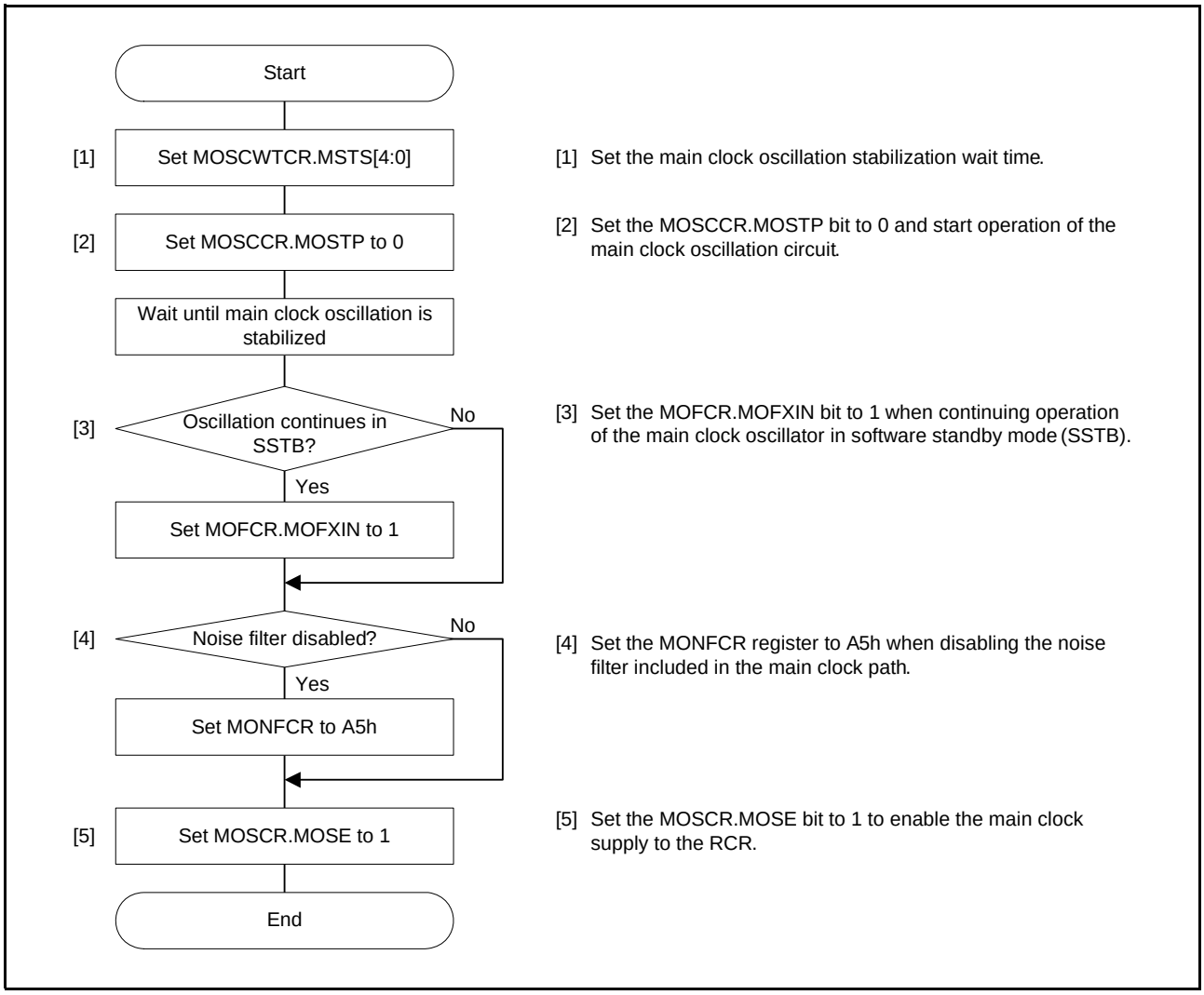


Figure 31.6 Example of Flowchart for Starting RCRMCLK Supply to RCR

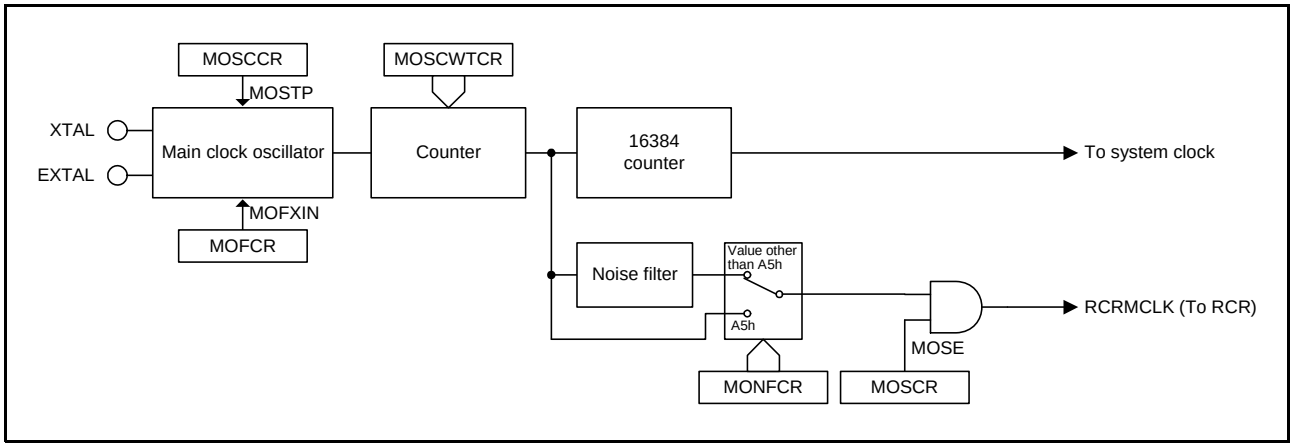


Figure 31.7 Path for Main Clock Supply to RCR

31.3.3.3 Using TMR Compare Match Output as RCR Operating Clock

The TMR compare match output can be supplied as the RCR operating clock. TMO0 and TMO2 can be supplied to the RCR0 and RCR1, respectively. For details on the TMR compare match output, see section 26, 8-Bit Timer (TMR).

31.3.4 PMCn Input

The options below can be selected in PMCn (n = 0, 1) input.

- Input polarity
- Digital filter

Figure 31.8 shows the configuration of PMCn internal input signal generation.

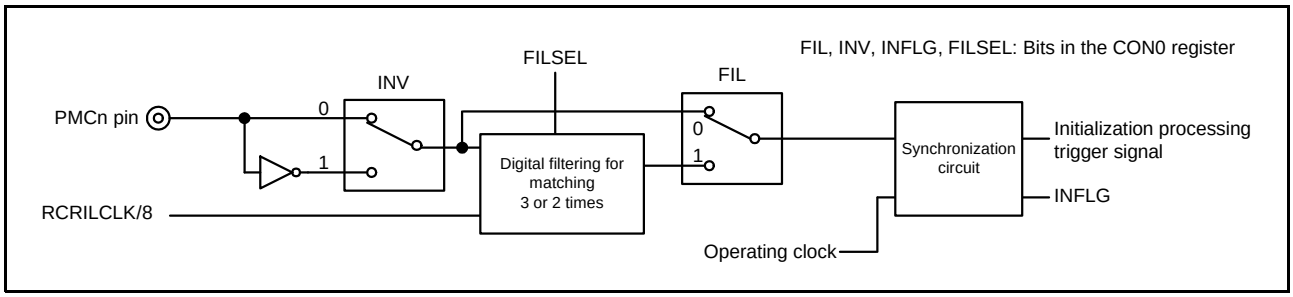
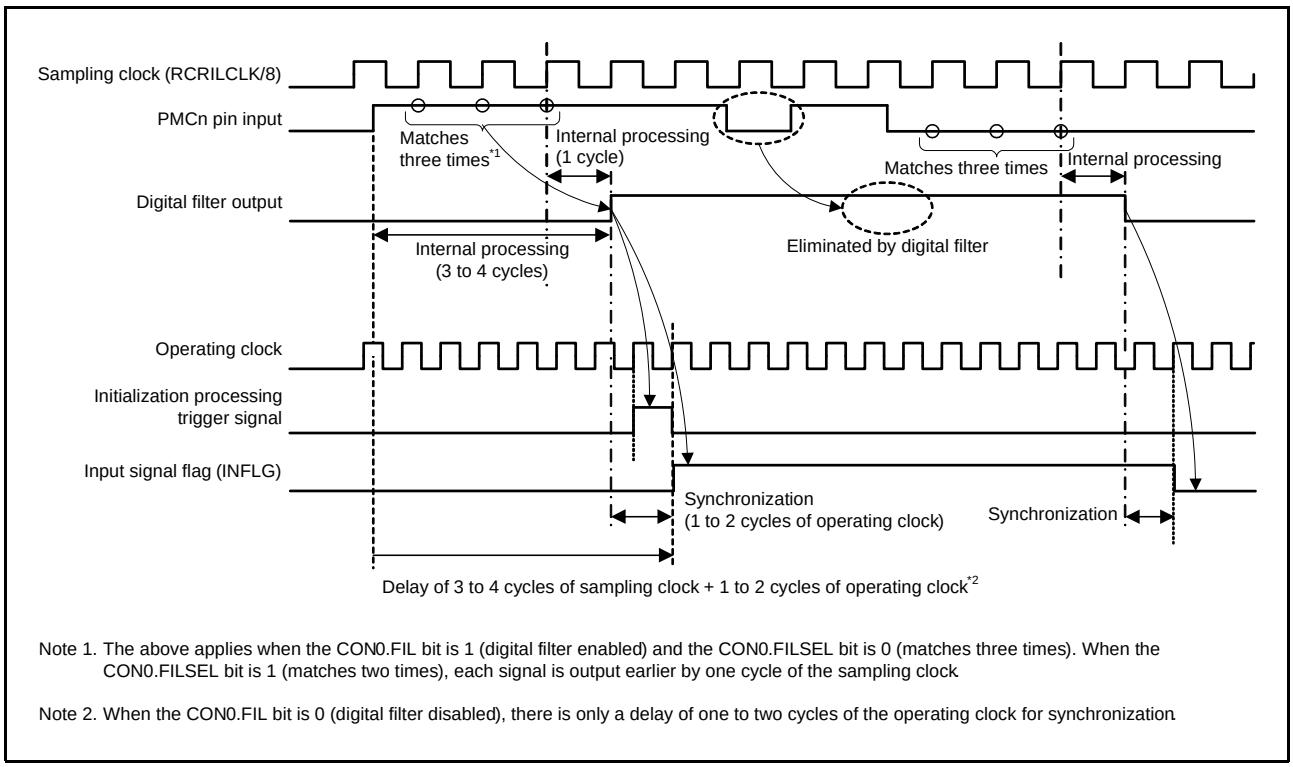


Figure 31.8 PMCn Internal Input Signal Generation Configuration

The input polarity of the PMCn pin can be inverted. Whether to invert or not can be selected by the CON0.INV bit. When the CON0.FIL bit is 1 (digital filter enabled), if the signal input to the PMCn pin holds the same level for k sequential cycles (k = 3 or 2; value selected by the CON0.FILSEL bit), that level is transferred to the internal circuit. This enables noise to be eliminated from k cycles of the sampling clock. The sampling clock of the digital filter is a divided-by-8 clock of the RCRILCLK supplied from the IWDT-dedicated on-chip oscillator.

Input to the PMCn pin is transferred as the CON0.INFLG flag (input signal flag) and the initialization processing trigger signal to the internal circuit in synchronization with the operating clock. The initialization processing trigger signal is used to initialize the internal base timer to the event corresponding to the CON1.TYP[1:0] setting. There is a delay caused by internal processing after the input to the PMCn pin is changed and before these signals are generated. Figure 31.9 shows digital filtering for PMCn input.



Note 1. The above applies when the CON0.FIL bit is 1 (digital filter enabled) and the CON0.FILSEL bit is 0 (matches three times). When the CON0.FILSEL bit is 1 (matches two times), each signal is output earlier by one cycle of the sampling clock.

Note 2. When the CON0.FIL bit is 0 (digital filter disabled), there is only a delay of one to two cycles of the operating clock for synchronization.

Figure 31.9 Digital Filtering for PMCn Input

31.3.5 Pattern Detection

The RCR has a function that detects the following patterns.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern

Using the base timer included in the RCR, the time between the edges of the external input signal is measured to determine which pattern matches the measurement result. This enables detection of the remote control signal and capturing the data. The width for determining each pattern can be set to any value using each pattern setting register. Figure 31.10 shows the waveform of RCR operation.

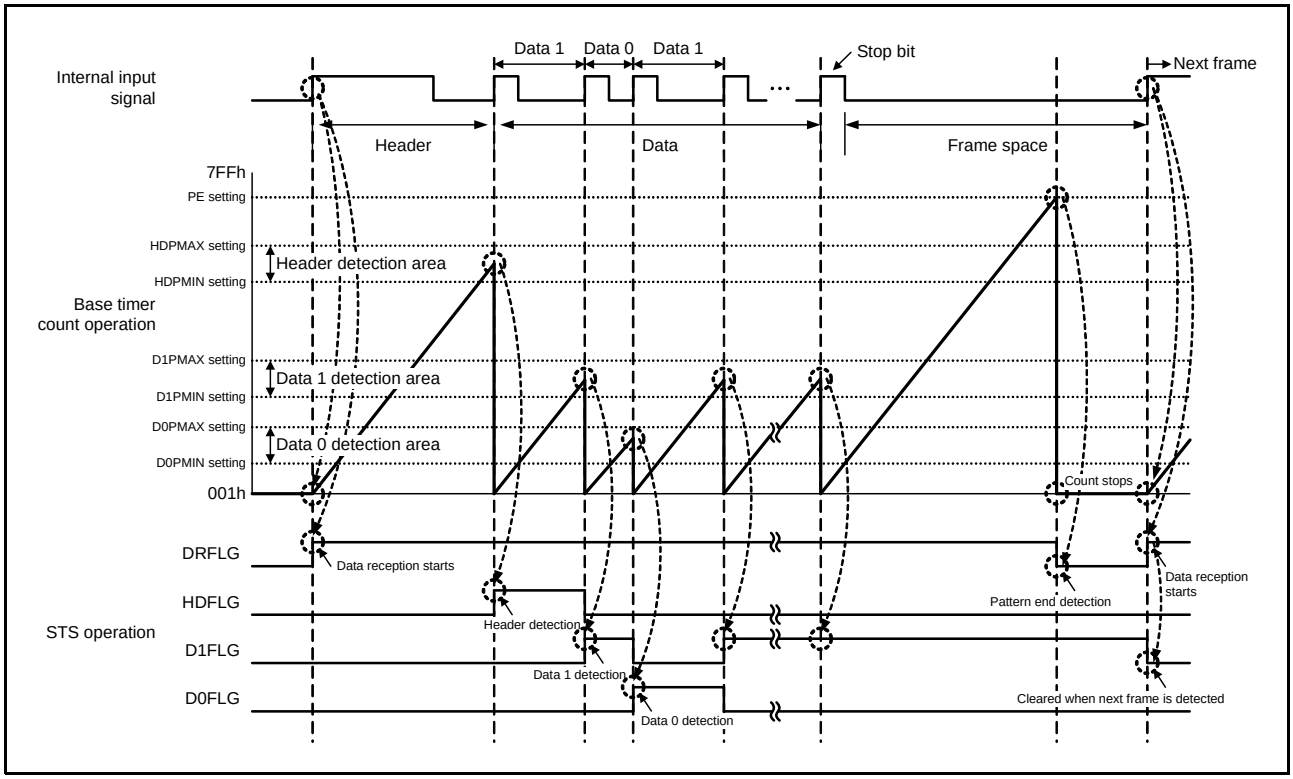


Figure 31.10 Waveform of RCR Operation

31.3.5.1 Header Pattern Detection

The header pattern can be detected by setting the minimum width of the header pattern in the HDPMIN register and the maximum width in the HDPMAX register.
The minimum and maximum widths of the header pattern must be “1 < HDPMIN register value ≤ HDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of header pattern}}{\text{Operating clock cycle}}$$

When not using the header pattern, set the HDPMIN and HDPMAX registers to 000h.
Make sure that the setting value of the header pattern is different from the setting values of data 0, data 1, and special data patterns, and the setting ranges are not overlapped.

When the CON1.TYP[1:0] bits are 00b or 01b, if the data 0, data 1, or special data pattern is detected before the header pattern is detected, the following occur:

- The STS.REFLG flag becomes 1 (an error has occurred).
- The STS.D0FLG, STS.D1FLG, and STS.SDFLG flags remain unchanged.
- The DAT0 to DAT7 registers remain unchanged.

When the CON1.TYP[1:0] bits are 01b, the number of detecting the header pattern is one while DRFLG is 1.

31.3.5.2 Data 0 Pattern Detection

The data 0 pattern can be detected by setting the minimum width of the data 0 pattern in the D0PMIN register and the maximum width in the D0PMAX register.
The minimum and maximum widths of the data 0 pattern must be “1 < D0PMIN register value ≤ D0PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 0 pattern}}{\text{Operating clock cycle}}$$

When not using the data 0 pattern, set the D0PMIN and D0PMAX registers to 00h.
Make sure that the setting value of the data 0 pattern is different from the setting values of the header, data 1, and special data patterns, and the setting ranges are not overlapped.

When the CON1.TYP[1:0] bits are 00b or 01b, if the data 0 pattern or data 1 pattern is detected before the head pattern is detected, the following occur:

- The STS.REFLG flag becomes 1 (an error has occurred).
- The STS.D0FLG, STS.D1FLG, and STS.SDFLG flags remain unchanged.
- The DAT0 to DAT7 registers remain unchanged.

31.3.5.3 Data 1 Pattern Detection

The data 1 pattern can be detected by setting the minimum width of the data 1 pattern in the D1PMIN register and the maximum width in the D1PMAX register.
The minimum and maximum widths of the data 1 pattern must be “1 < D1PMIN register value ≤ D1PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 1 pattern}}{\text{Operating clock cycle}}$$

When not using the data 1 pattern, set the D1PMIN and D1PMAX registers to 00h.
Make sure that the setting value of the data 1 pattern is different from the setting values of the header, data 0, and special data patterns, and the setting ranges are not overlapped.

When the CON1.TYP[1:0] bits are 00b or 01b, if the data 0 pattern or data 1 pattern is detected before the head pattern is detected, the following occur:

- The STS.REFLG flag becomes 1 (an error has occurred).
- The STS.D0FLG, STS.D1FLG, and STS.SDFLG flags remain unchanged.
- The DAT0 to DAT7 registers remain unchanged.

31.3.5.4 Special Data Pattern Detection

The special data pattern can be detected by setting the minimum width of the special data pattern in the SDPMIN register and the maximum width in the SDPMAX register.
The minimum and maximum widths of the special data pattern must be “1 < SDPMIN register value ≤ SDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of special data pattern}}{\text{Operating clock cycle}}$$

When not using the special data pattern, set the SDPMIN and SDPMAX registers to 00h.
Make sure that the setting value of the special data pattern is different from the setting values of the header, data 0, and data 1 patterns, and the setting ranges are not overlapped.

When the CON1.TYP[1:0] bits are 00b or 01b, if the special data pattern is detected before the head pattern is detected, the following occur:

- The STS.REFLG flag becomes 1 (an error has occurred).
- The STS.SDFLG flag remain unchanged.
- The DAT0 to DAT7 registers remain unchanged.

31.3.5.5 Examples of Setting Pattern Setting Registers

For the header, data 0, data 1, and special data setting registers, make sure that the minimum to maximum values of each pattern are different, and the setting ranges do not overlap as shown in Figure 31.11.

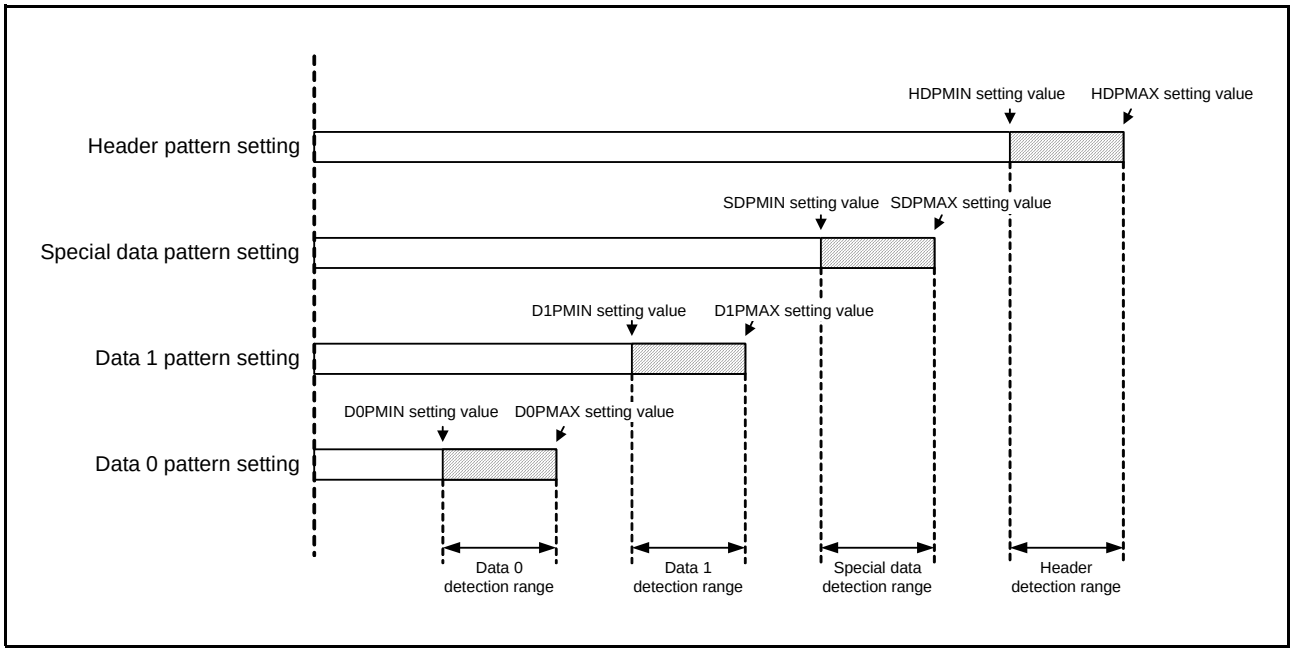


Figure 31.11 Examples of Setting Pattern Setting Registers

31.3.5.6 Updating Status Flags upon Pattern Detection

The detected patterns can be confirmed by reading the following flags: header pattern match flag (STS.HDFLG), data 0 pattern match flag (STS.D0FLG), data 1 pattern match flag (STS.D1FLG), and special data pattern match flag (STS.SDFLG). These flags are negated when a different pattern is detected. If a pattern other than the above patterns is detected, it is detected as an error pattern. This can be confirmed by reading the receive error flag (STS.REFLG). This flag is negated when the next frame is received. Figure 31.12 shows pattern detection and an example of flag operation.

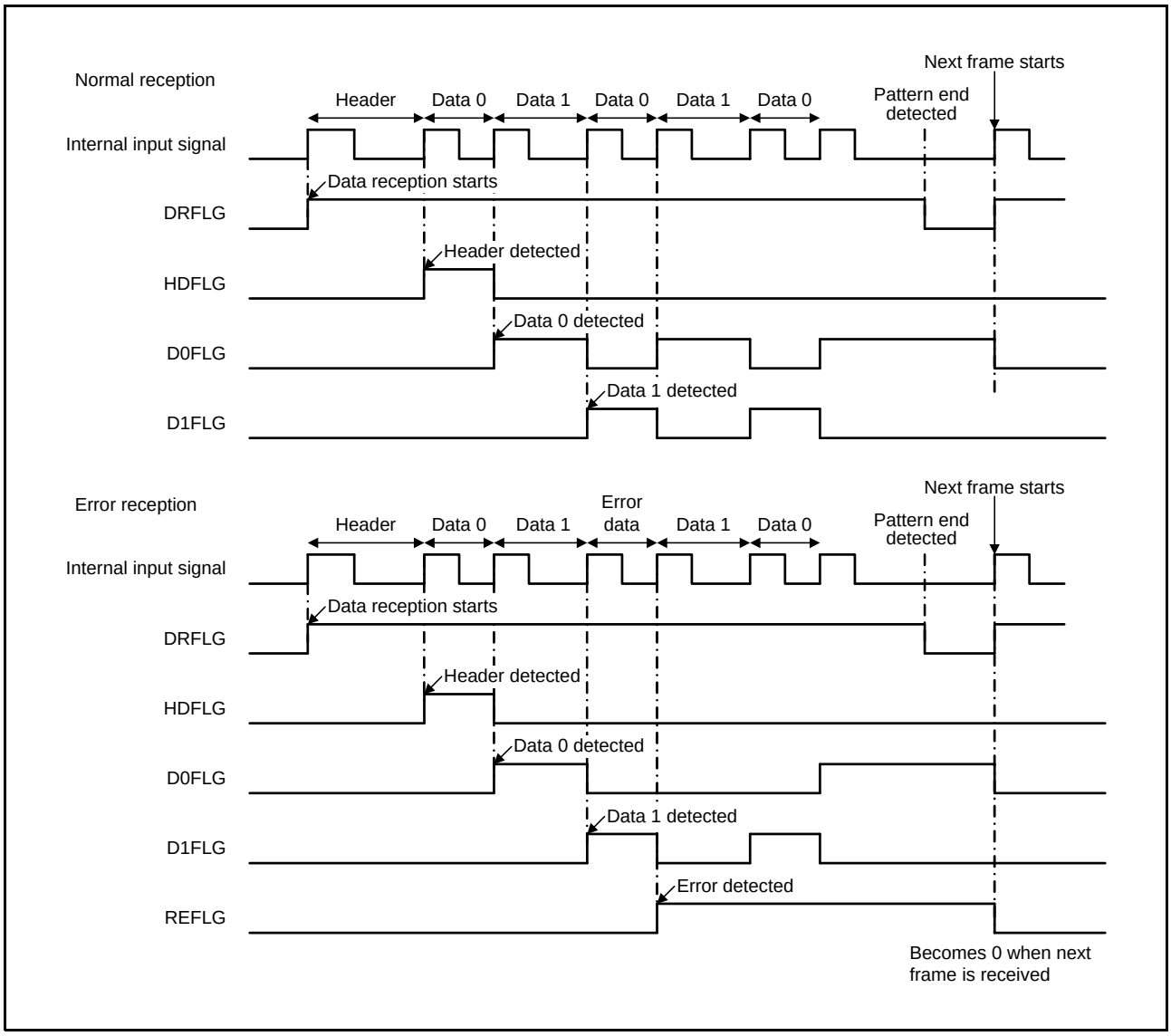


Figure 31.12 Example of Flag Operation

31.3.6 Pattern End

The timing when the STS.DRFLG flag becomes 0 can be set.

When setting the PE register, be sure to set that the PE value > HDPMAX, D0PMAX, D1PMAX, or SDPMAX value.

When the PE value ≤ HDPMAX, D0PMAX, D1PMAX, or SDPMAX value, the PE register cannot be used to set the timing when the STS.DRFLG flag becomes 0. In this case, data reception is completed according to the largest value from among the setting values of the HDPMAX, D0PMAX, D1PMAX, and SDPMAX registers.
Figure 31.13 shows operation of the data reception complete flag for each pattern end setting.

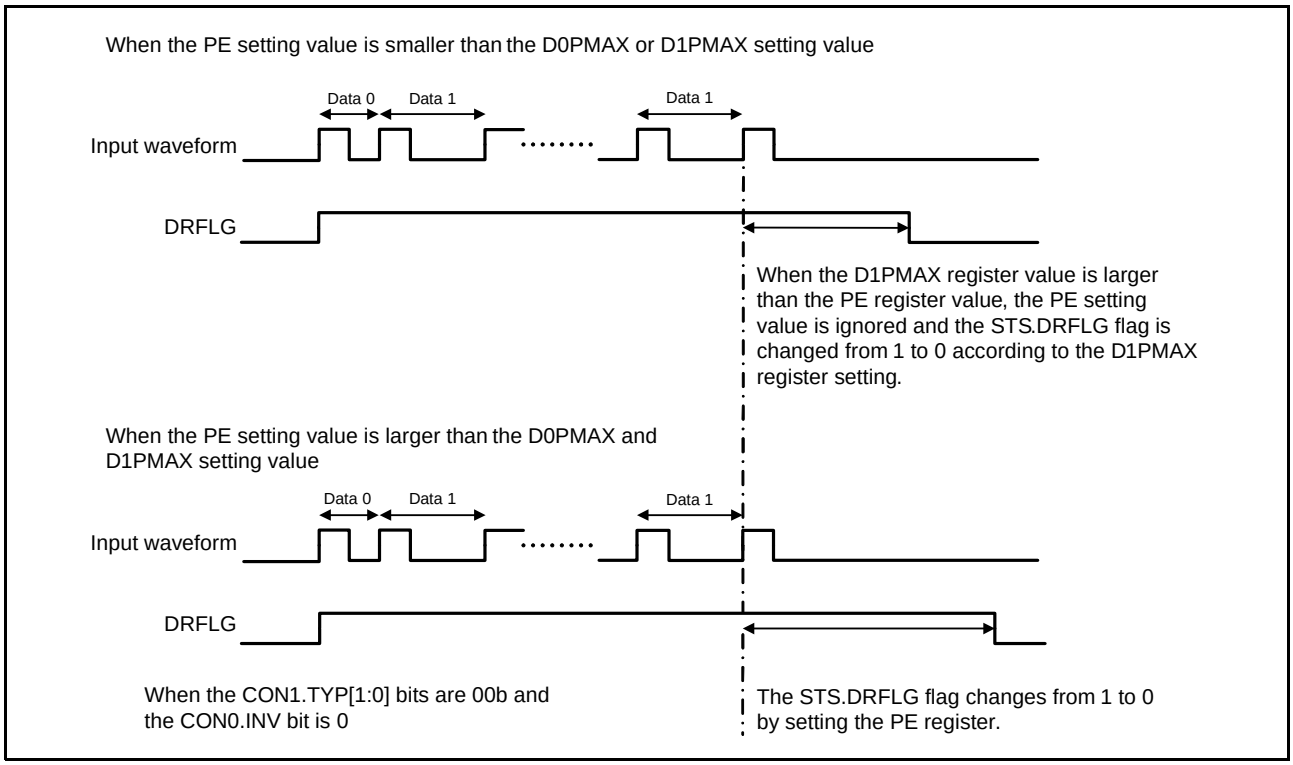


Figure 31.13 Operation of Data Reception Complete Flag for Each Pattern End Setting

31.3.7 Receive Data Buffer

The receive data j register (DATj) (j = 0 to 7) is an 8-byte (64-bit) buffer for storing received data. When data 0 pattern or data 1 pattern is detected, the detection result is sequentially stored starting from the DAT0.DAT00 bit as shown in Figure 31.14. RBIT is counted up at the same time, so the number of the current received bits can be checked by reading RBIT. See Table 31.5 for the relationship between the number of received bits and the location where data is stored. The values of DATj and RBIT do not change even when the header pattern or special pattern is received. If DATj or RBIT is read while the data is being updated, the value read may be undefined.

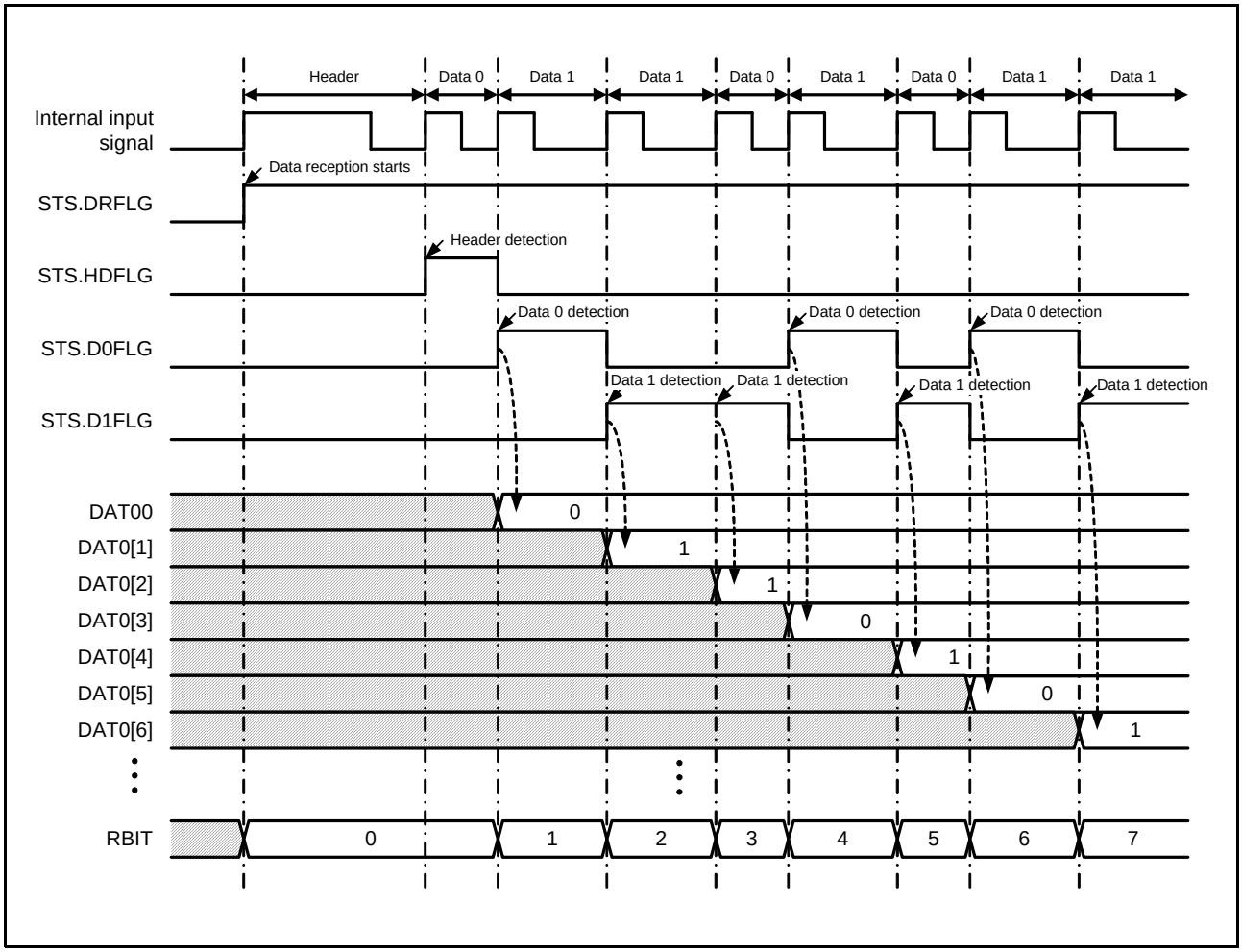


Figure 31.14 Operation of Receive Data Buffer

Table 31.5 Relationship between Number of Received Bits and Location Where Data is Stored

Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
1	DAT0	DAT00
2		DAT0[1]
3		DAT0[2]
4		DAT0[3]
5		DAT0[4]
6		DAT0[5]
7		DAT0[6]
8		DAT0[7]
9	DAT1	DAT1[0]
10		DAT1[1]
11		DAT1[2]
12		DAT1[3]
13		DAT1[4]
14		DAT1[5]
15		DAT1[6]
16		DAT1[7]
17	DAT2	DAT2[0]
18		DAT2[1]
19		DAT2[2]
20		DAT2[3]
21		DAT2[4]
22		DAT2[5]
23		DAT2[6]
24		DAT2[7]
25	DAT3	DAT3[0]
26		DAT3[1]
27		DAT3[2]
28		DAT3[3]
29		DAT3[4]
30		DAT3[5]
31		DAT3[6]
32		DAT3[7]

Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
33	DAT4	DAT4[0]
34		DAT4[1]
35		DAT4[2]
36		DAT4[3]
37		DAT4[4]
38		DAT4[5]
39		DAT4[6]
40		DAT4[7]
41	DAT5	DAT5[0]
42		DAT5[1]
43		DAT5[2]
44		DAT5[3]
45		DAT5[4]
46		DAT5[5]
47		DAT5[6]
48		DAT5[7]
49	DAT6	DAT6[0]
50		DAT6[1]
51		DAT6[2]
52		DAT6[3]
53		DAT6[4]
54		DAT6[5]
55		DAT6[6]
56		DAT6[7]
57	DAT7	DAT7[0]
58		DAT7[1]
59		DAT7[2]
60		DAT7[3]
61		DAT7[4]
62		DAT7[5]
63		DAT7[6]
64		DAT7[7]

Note: When the data exceeds 64 bits, the DATj register is sequentially overwritten from the first bit.

When 0 is written to the DAT0.DAT00 bit, the values of the DAT0 to DAT7 registers become 00h after one to two cycles of the operating clock. Figure 31.15 shows DAT/RBIT register operation when 00h is written to the DAT0 register.

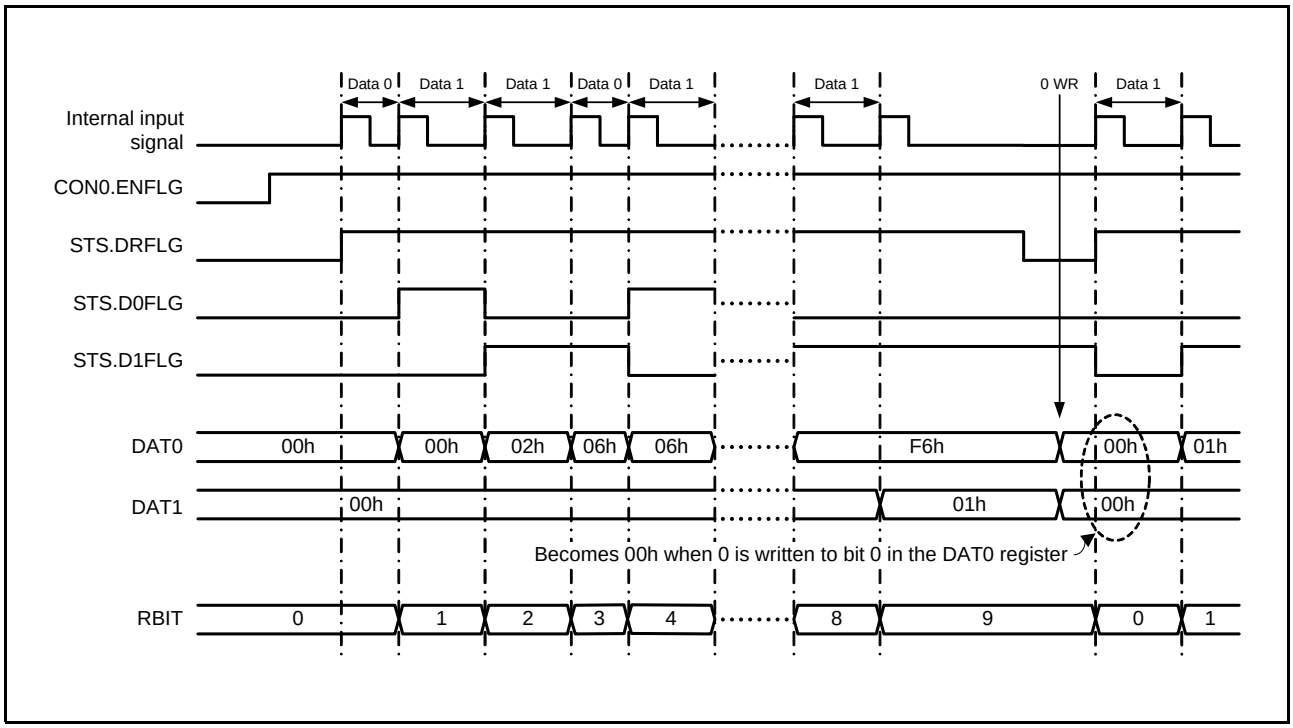


Figure 31.15 DAT/RBIT Operation (00h is Written to DAT0 Register)

When 0 is written to the RBIT.RBIT0 bit, the value of RBIT becomes 00h after one to two cycles of the operating clock. When the CON1.TYP[1:0] bits are 00b or 01b, if the header pattern is detected during data reception, the value of the RBIT is initialized to 00h and the received data is sequentially overwritten from the DAT0.DAT00 bit. Figure 31.16 shows operation of header pattern detection during data reception.

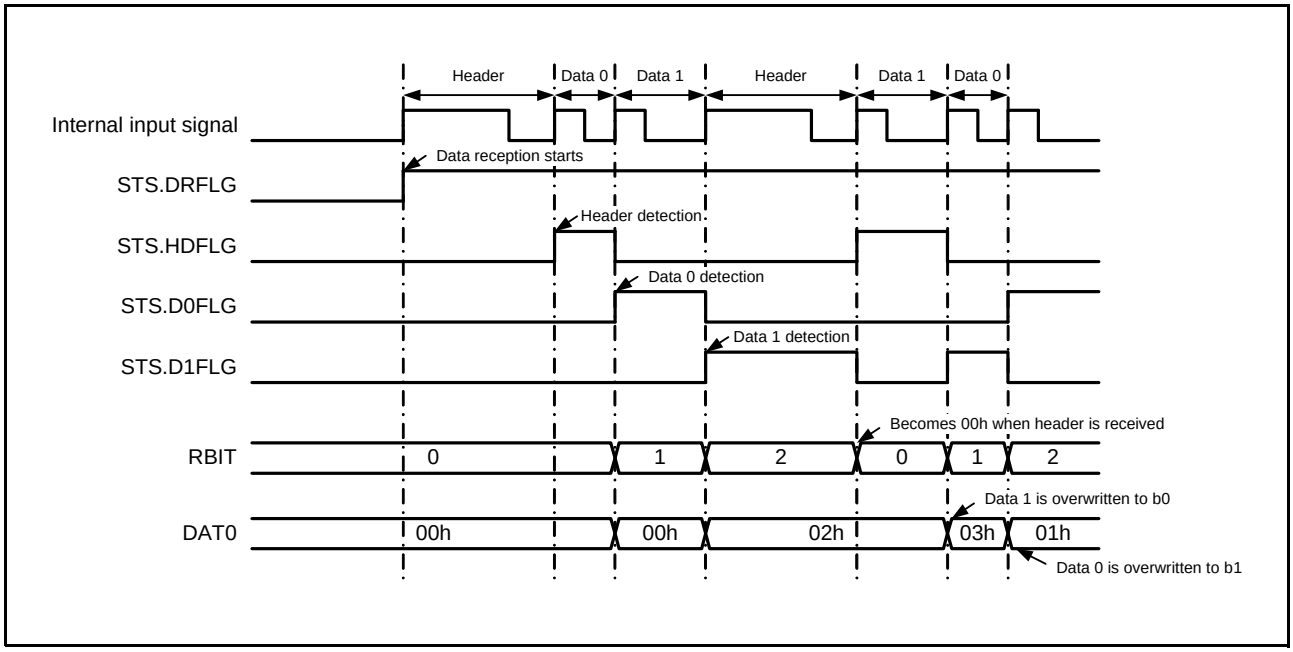


Figure 31.16 Operation of Header Pattern Detection during Data Reception

When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. Figure 31.17 shows RBIT operation when the STS.BFULFLG flag becomes 1.

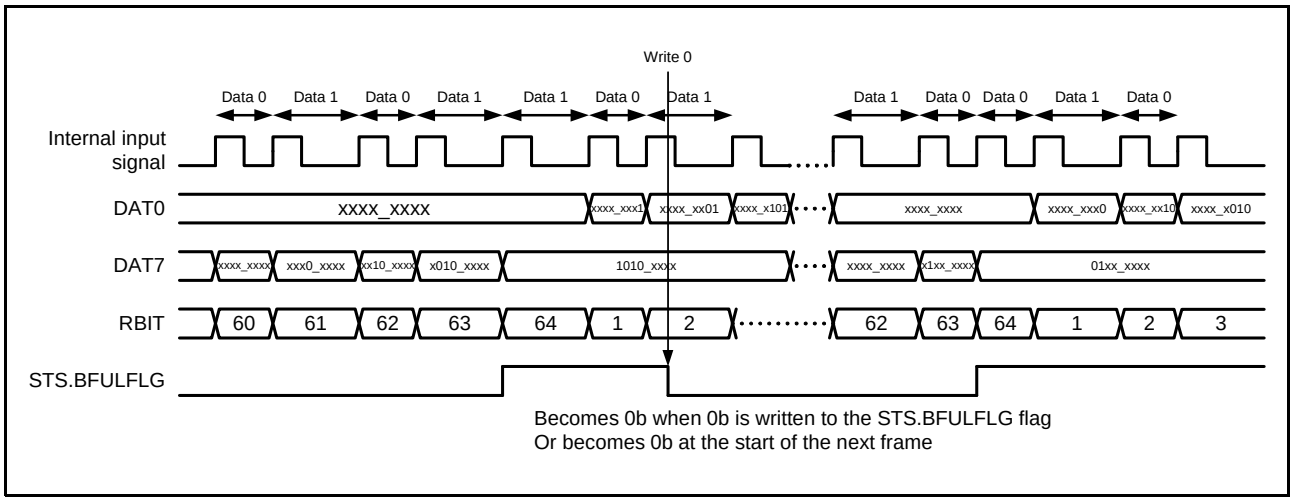


Figure 31.17 RBIT Operation (STS.BFULFLG Flag = 1)

31.3.8 Compare Function

The RCR has a function to compare the value of the CPD register with the value of the DAT0 register. As a result of comparison, it can be detected that the first 1 to 8 bits of the remote control signal are the specific values. Figure 31.18 shows the operation timing of the receive buffer and the compare function.

When using the compare function, set the following:

- Select bits to be compared by setting the CPC.CPN[2:0] bits (when the setting value is n, bits n to 0 are compared. n: 0 to 7).
- Set the compare data in the CPD register.
  When the value of the RBIT register becomes the bit count specified by the CPC.CPN[2:0] bits, if the stored comparison result between the CPD and DAT0 registers matches, the STS.CPFLG flag becomes 1 (compare match).

When the value of the RBIT register matches the bit count specified by the CPC.CPN[2:0] bits during reception of 64 bits or more, even if the comparison result between CPD and DAT0 matches, the STS.CPFLG flag does not become 1 (compare match).

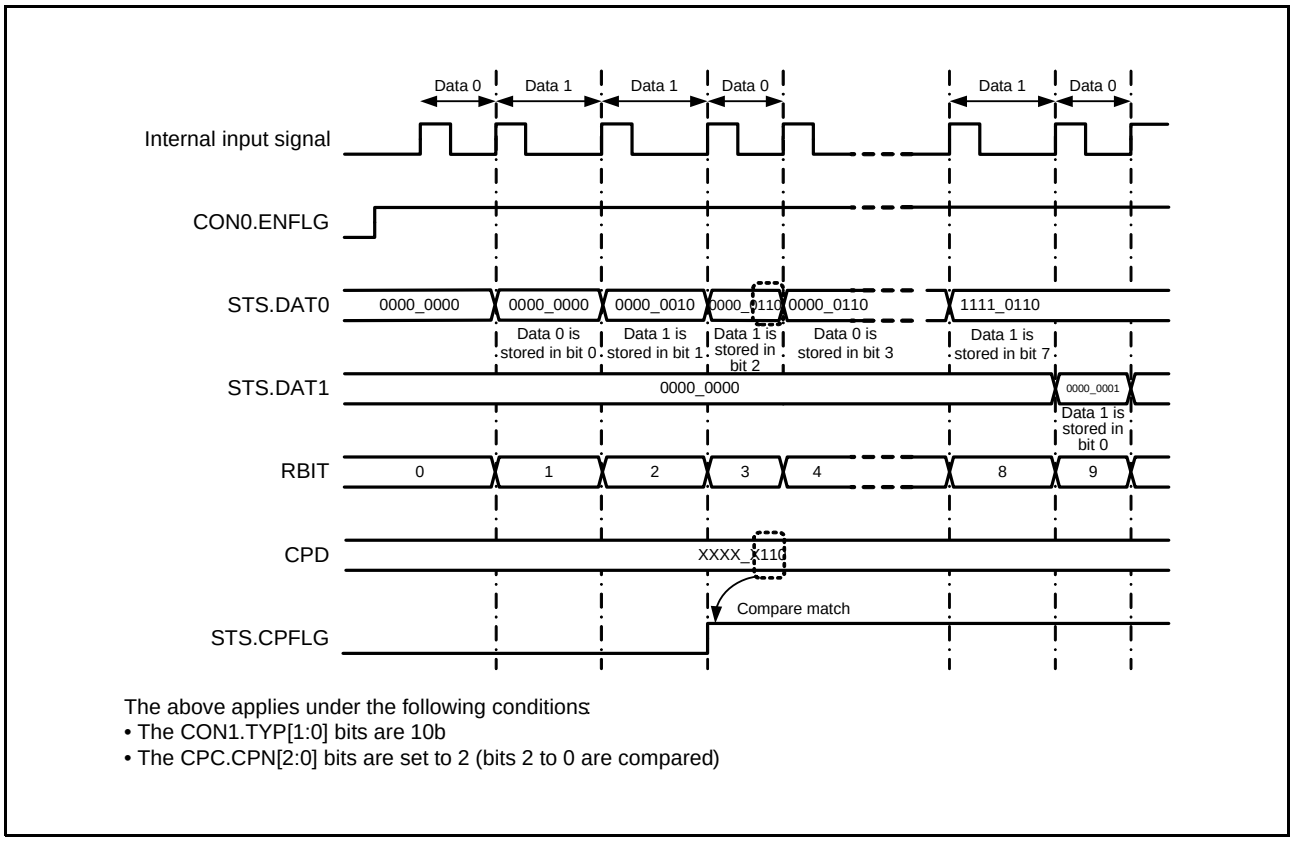


Figure 31.18 Receive Buffer and Compare Function

31.3.9 Error Pattern Reception

When the error pattern is detected during data reception, subsequent operation differs depending on the setting of the CON0.EC bit.

Figure 31.19 shows operation of the DAT and RBIT registers when the CON0.EC bits are set to 0.

If an error is detected while the CON0.EC bit is 0, the data when the error is detected is not captured, but the data is captured when the data 0 pattern or data 1 pattern is detected later.

Figure 31.20 shows operation of the DAT and RBIT registers when the CON0.EC bits are set to 1.

If an error is detected while the CON0.EC bit is 1, the values of the DAT0 to DAT0 registers are not updated even when the data 0 pattern or data 1 pattern is detected later. Once STS.DRFLG is cleared and after data reception is completed, if data reception starts again, STS.REFLG is cleared and the data is captured.

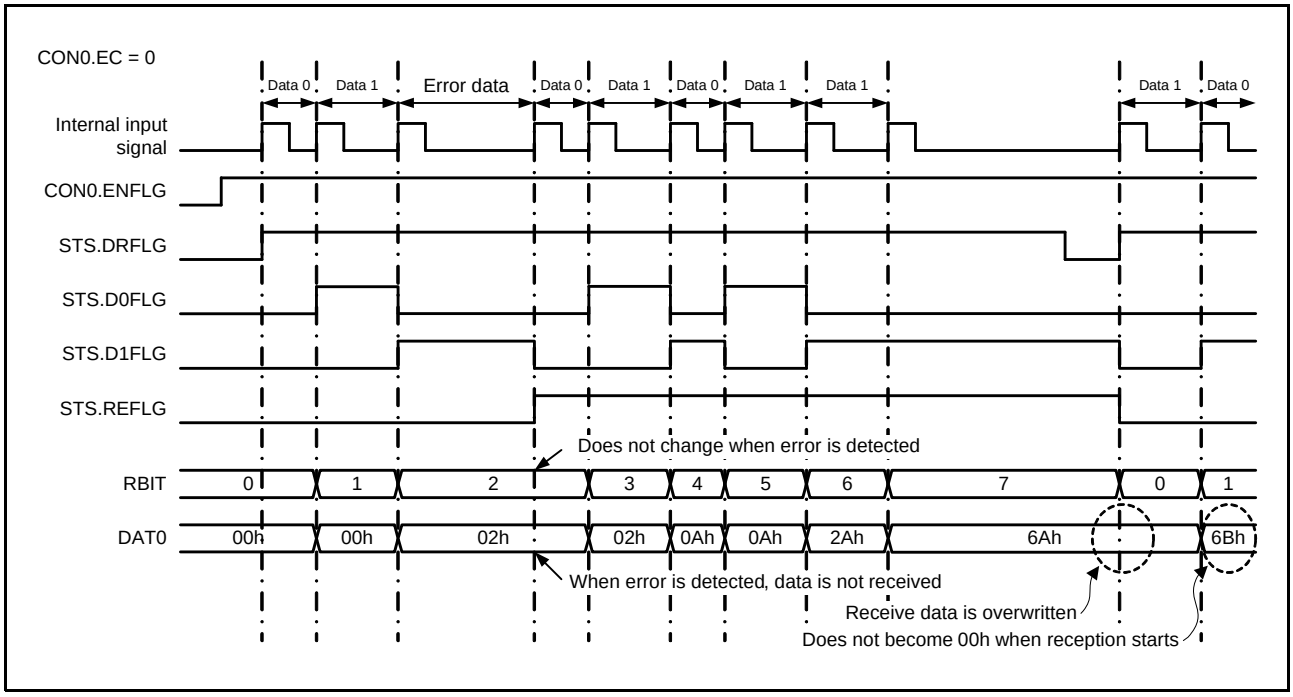


Figure 31.19 DAT and RBIT Operation upon Error Detection (CON0.EC Bit = 0)

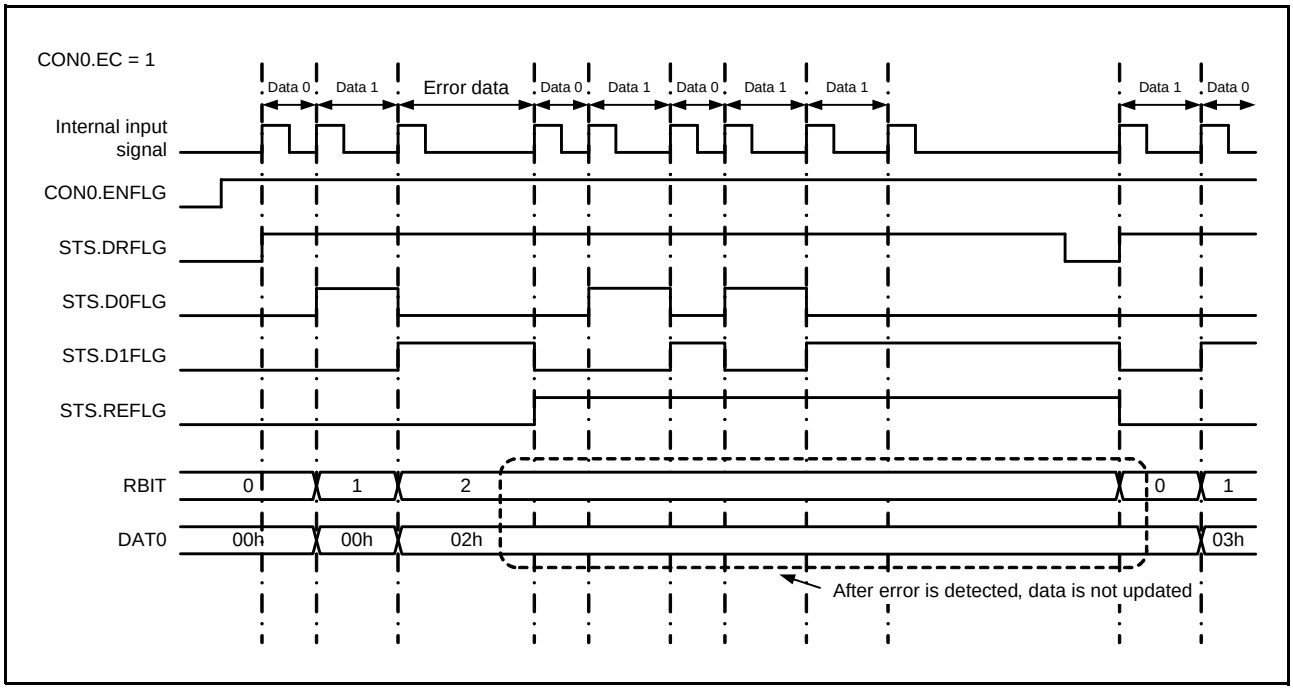


Figure 31.20 DAT and RBIT Operation upon Error Detection (CON0.EC Bit = 1)

31.3.10 Storing Base Timer Value When Event Occurs

The measurement result register (TIM) stores the base timer value when one of the events below occurs. Figure 31.21 shows an operation example of the measurement function.

- Header pattern detection
- Data 0 pattern detection
- Data 1 pattern detection
- Special data pattern detection
- Data pattern detection other than the above (receive error)

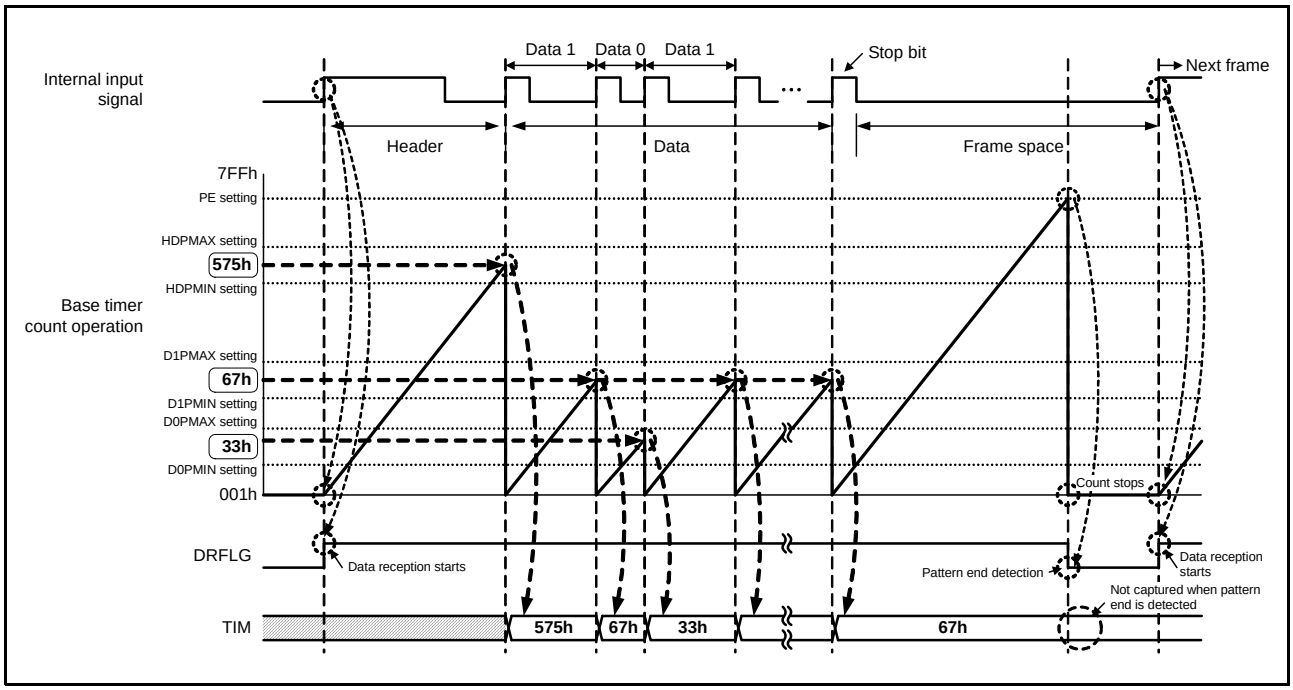


Figure 31.21 Operation Example of Measurement Function

31.3.11 Interrupts

The RCR generates the following interrupt requests: compare match, receive error, data reception complete, receive buffer full, header pattern match, data 0 pattern or data 1 pattern match, and special data pattern match interrupts. Each of these interrupt requests is assigned to a single vector address for each channel, and generated each time the condition is met. If an interrupt enable bit in the INT register is 1, an interrupt request is output when the corresponding generation condition is met.

Table 31.6 lists the interrupt source of the RCR. Refer to section 14, Interrupt Controller (ICUb) for details on interrupt control.

Table 31.6 RCR Interrupt Sources

Interrupt Source	Interrupt Request Generation Condition	Interrupt Status Flag	Interrupt Enable Bit
Compare match	When STS.CPFLG changes from 0 to 1	STS.CPFLG	INT.CPINT
Receive error	When STS.REFLG changes from 0 to 1 (When a receive error is detected)	STS.REFLG	INT.REINT
Completion of data reception	When STS.DRFLG changes from 1 to 0	STS.DRFLG	INT.DRINT
Receive buffer full	When STS.BFULFLG changes from 0 to 1	STS.BFULFLG	INT.BFULINT
Header pattern match	When STS.HDFLG changes from 0 to 1 (When the header pattern is detected)	STS.HDFLG	INT.HDINT
Data 0 pattern or data 1 pattern match	• When STS.D0FLG changes from 0 to 1 (When the data 0 pattern is detected) • When STS.D1FLG changes from 0 to 1 (When the data 1 pattern is detected)	STS.D0FLG, STS.D1FLG	INT.DINT
Special data pattern match	When STS.SDFLG changes from 0 to 1 (When the special data pattern is detected)	STS.SDFLG	INT.SDINT

31.3.12 Data Reception in Low Power Consumption State

In this MCU, data can be received in a low power consumption state (sleep mode, all-module clock stop mode, or software standby mode).

To receive data in a low power consumption state, RCR communications should be set before transitioning to the state. In this case, set the CON1.CSRC[1:0] bits to 00b or 10b, and select the divided RCRILCLK or RCRMCLK clock as the operating clock.

31.3.12.1 Using RCR Interrupt Request to Return from Low Power Consumption State

Power consumption while waiting for data reception can be reduced by using the RCR interrupt request to be output during data reception as the source for returning from the low power consumption state (see Figure 31.22). Pattern detection and compare function enable returning from the low power consumption state only when specified data is received. When using the RCR interrupt to return from the low power consumption state, the RCR operating clock needs to be supplied constantly. Set the IWDTCSTPR.SLCSTP bit when the divided RCRILCLK clock is used as the RCR operating clock, and set the MOFCR.MOFXIN bit when the divided RCRMCLK is used.

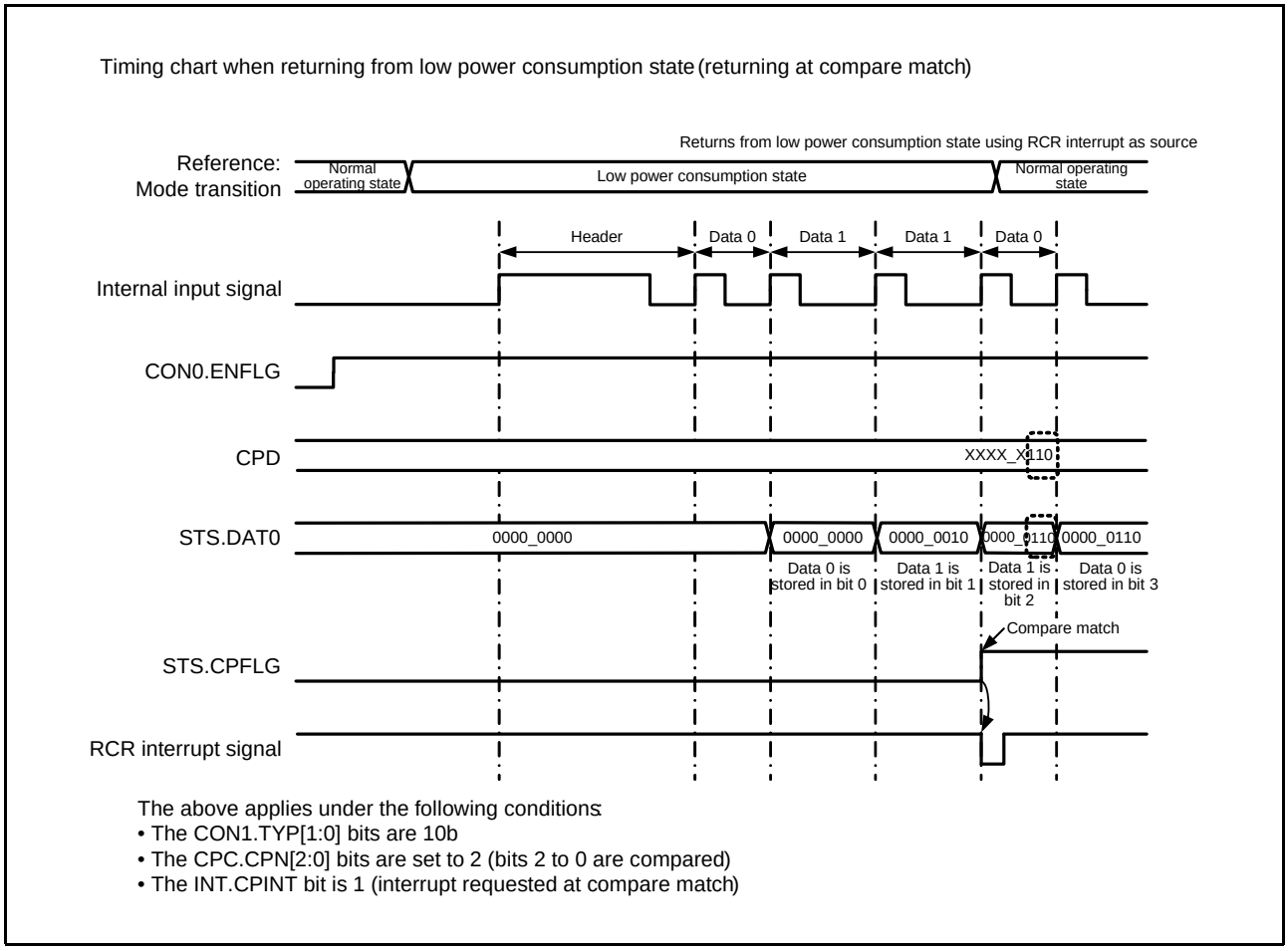


Figure 31.22 Using RCR Interrupt Request to Return from Low Power Consumption State

31.4 Usage Notes

31.4.1 Module Stop Function Setting

RCR operation can be disabled or enabled by setting the module stop control register. The RCR is stopped with the initial setting. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

31.4.2 Clock Division Ratio Setting

Set the peripheral module clock (PCLKB) frequency to higher than the RCR operating clock frequency.

31.4.3 Restriction on Using IWDT Function

Do not use the IWDT function when using the divided RCRILCLK clock as the RCR operating clock and when using the digital filter.

31.4.4 Starting/Stopping Operation of Remote Control Signal Receiver

The CON1.EN bit controls starting/stopping of operation of the remote control signal receiver.
The CON0.ENFLG flag indicates that the operation is enabled or disabled. After the CON1.EN bit is set to 1 (operation enabled), it takes up to zero to one cycle of the count source before the RCR circuit starts operating and the CON0.ENFLG flag becomes 1. During this period, do not access the RCR related registers (listed in section 31.2.1 to section 31.2.19) except for the CON0.ENFLG flag.

31.4.5 Accessing Registers

Change the following registers only when the CON1.EN bit and CON0.ENFLG flag are both 0 (RCR is stopped)

- CON0 register
- CON1 register (except for bits 0 to 2)
- INT register (except for bits 2 and 5)
- CPC register
- CPD register
- Pattern width setting registers for header, data 0, data 1, and special data patterns
- Pattern end setting register

When rewriting the CON1.TYP[1:0] bits while the CON1.EN bit or CON0.ENFLG flag is 1 (RCR is operating), change the values of these bits one bit at a time. If the CON1.TYP[1:0] bits are rewritten when the CON0.INFLG flag changes, the signal captured into the remote control signal receiver may be undefined.
After 0 is written to bit 0 in the DAT0 or RBIT register or the STS.BFULFLG flag, do not write 0 to the same bit again for two cycles of the operating clock. If 0 is written when the CON0.INFLG flag changes, the values of the DATj and RBIT registers and the STS.BFULFLG flag may be undefined.

31.4.6 PMCn Input Control

If the CON0.FILSEL, FIL, and CON0.INV bits are rewritten, the signal captured into the remote control signal receiver is undefined for three cycles of the digital filter sampling clock.

31.4.7 Operating Clock

When the CON1.CSRC[2:0] bits are rewritten, set the following registers again: CON0, CON1, INT, CPC, CPD, PE, and header, data 0, data 1, and special data pattern width setting registers.

31.4.8 Reading Registers

When the following registers are read while data changes, an undefined value may be read.
Flags in CON0 and STS (except for STS.DRFLG), TIM, DAT0 to DAT7, and RBIT

Follow the procedures below to avoid reading an undefined value.

- Using an interrupt
  Set the INT.DRINT bit to 1 (data reception complete interrupt enabled) and read the registers within the RCR interrupt routine.
- Monitoring by a program 1
  Set the INT.DRINT bit to 1 (data reception complete interrupt enabled) and monitor the ICU.IRn.IR flag by a program. Read the registers when the IF bit becomes 1 (interrupt request generated).
- Monitoring by a program 2
  (1) Monitor the STS.DRFLG flag.
  (2) When the STS.DRFLG flag becomes 1, monitor this flag until it becomes 0.
  (3) Read the necessary content of the registers when the STS.DRFLG flag becomes 0.

31.4.9 Notes on When Using RCRMCLK as Operating Clock

When the RCRMCLK is selected as an operating clock by setting the CON1.CSRC[2:0] bits, the ranges of the frequency and the pulse width that can be input as the main clock differ depending on whether the noise filter is enabled or disabled by the MONFCR register. For details, see section 41, Electrical Characteristics.

32. CEC Transmission/Reception Circuit (CEC)

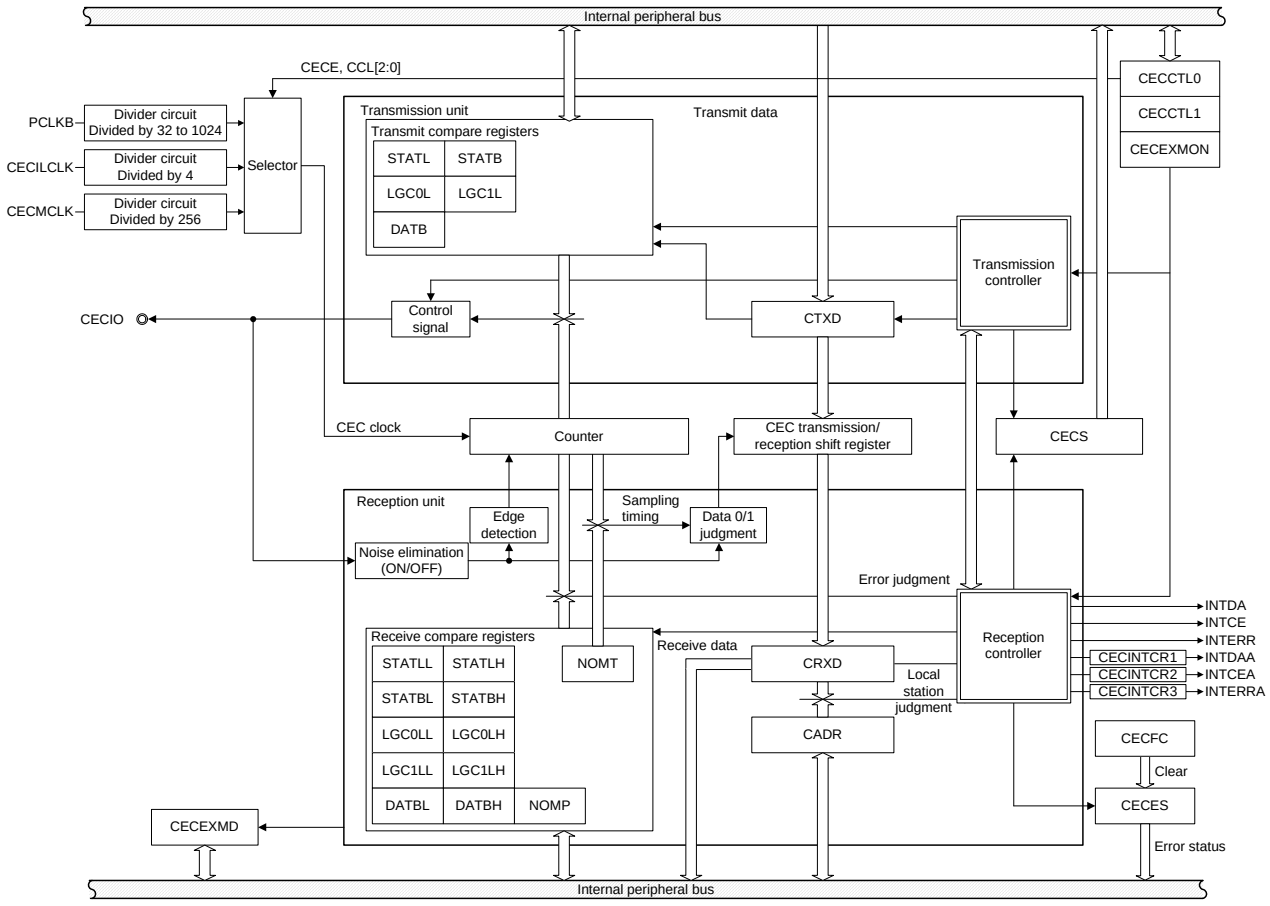
32.1 Overview

The CEC transmission/reception circuit can generate and receive CEC signals conforming to the Consumer Electronics Control (CEC) standard, and automatically detect communication statuses by the CEC. CEC transmission/reception can be easily controlled by using these functions.
Table 32.1 lists the CEC specifications.

Table 32.1 CEC Specifications

Item	Description
Communication method	Serial communication can be performed conforming to the Consumer Electronics Control (CEC) standard in the High-Definition Multimedia Interface (HDMI) Ver. 1.4b.
CEC clock	Selectable from among PCLKB/32, PCLKB/64, PCLKB/128, PCLKB/256, PCLKB/512, PCLKB/1024, CECILCLK*1/4, and CECMCLK*2/256
Interrupt sources	Three sources, two interrupts*3 • Data interrupt (INTDA, INTDAA) • Communication complete interrupt (INTCE, INTCEA) • Error interrupt (INTERR, INTERRA) (transmit, ACK, arbitration, timing, underrun, overrun, and bus lock errors)
Other functions	• Communication bit width adjustment function Can be used to set the low-level width and bit width of the start bit and data bit that configure the CEC data frame during transmission. • Count function during signal-free time Can be used to count during the signal-free time (transmission disable period) specified in the CEC standard and set the count period. • Error handling function Can be used to output an error handling pulse by detecting a timing error of the data bit. • Function that restarts reception by detecting the start bit during reception • Data can be received in a low power consumption state, and returning from the state is possible when the local address and reception address match.

Note 1. CECILCLK is an operating clock supplied from the IWDT-dedicated on-chip oscillator.
Note 2. CECMCLK is an operating clock supplied from the main clock oscillator.
Note 3. Each of three interrupt sources can be used to output the same two types of interrupt requests. Use the two types of interrupt requests exclusively depending on the application of the interrupt request. In this section, only one of the two types is described.



CECEXMD:	CEC extension mode register
STATL:	CEC transmission start bit low width setting register
LGC0L:	CEC transmission logical 0 low width setting register
DATB:	CEC transmission data bit width setting register
STATB:	CEC transmission start bit width setting register
LGC1L:	CEC transmission logical 1 low width setting register
STATLL:	CEC reception start bit minimum low width setting register
STATBL:	CEC reception start bit minimum bit width setting register
LGC0LL:	CEC reception logical 0 minimum low width setting register
LGC1LL:	CEC reception logical 1 minimum low width setting register
DATBL:	CEC reception data bit minimum bit width setting register
STATLH:	CEC reception start bit maximum low width setting register
STATBH:	CEC reception start bit maximum bit width setting register
LGC0LH:	CEC reception logical 0 maximum low width setting register
LGC1LH:	CEC reception logical 1 maximum low width setting register
DATBH:	CEC reception data bit maximum bit width setting register
NOMT:	CEC reception data sampling time setting register
NOMP:	CEC data bit reference width setting register
CTXD:	CEC transmission buffer register
CRXD:	CEC reception buffer register
CADR:	CEC local address setting register
CECCTL0:	CEC control register 0
CECCTL1:	CEC control register 1
CECEXMON:	CEC extension monitor register
CECS:	CEC communication status register
CECFC:	CEC communication error flag clear trigger register
CECES:	CEC communication error status register
CECINTCR1:	CEC interrupt control register 1
CECINTCR2:	CEC interrupt control register 2
CECINTCR3:	CEC interrupt control register 3

Figure 32.1 CEC Block Diagram

Figure 32.2 shows an example of I/O pin connection to the external circuit. Table 32.2 lists an I/O pin used for the CEC.

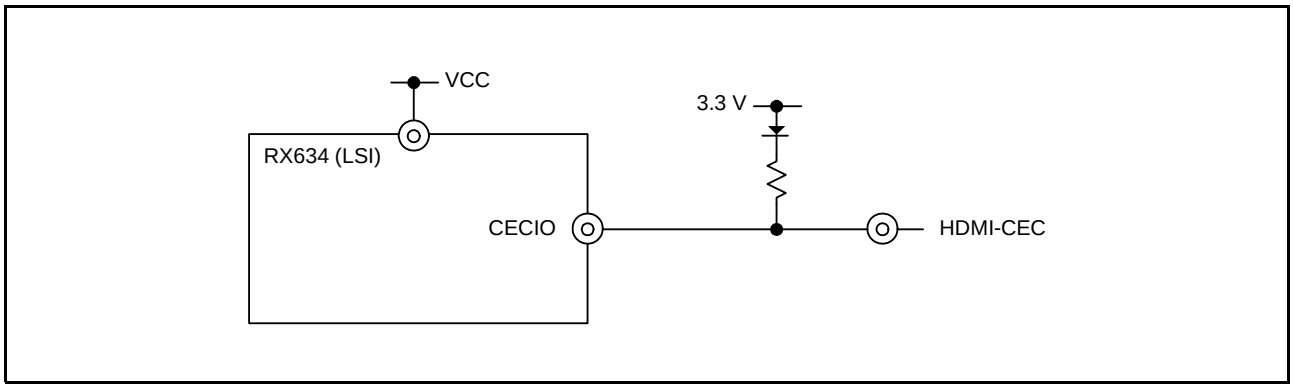


Figure 32.2 Example of I/O Pin Connection to External Circuit

Table 32.2 CEC Pin Configuration

Pin Name	I/O	Function
CECIO	I/O	CEC data communication

32.1.1 Term Description

- Initiator: Device that transmits or is transmitting CEC messages
- Follower: Device that receives or is receiving CEC messages
- Message: All data from the start bit to the operand
- Initiator address: Source address
- Destination address: Destination address
- Direct address communication (direct address message): Communication with one follower
- Broadcast communication (broadcast message): Communication with multiple followers
- Arbitration: Prioritizing the devices that output a low level to the CEC line when multiple initiators exist
- Arbitration loss: State in which competing devices are prioritized. At this time, the local station stops transmitting
- Bus free: State in which no communication is performed and transmission can be performed
- Bus busy: During communication
- Error handling: Outputting an error handling pulse (low level with a width of the bit width × 1.5) and transitioning to the communication standby state when a received bit width is shorter than the setting bit width of the data
- ACK/NACK: The logic levels received at the ACK bit timing are as follows
  ACK: Outputs logical 0
  NACK: Outputs logical 1

  (Example) If the initiator outputs logical 1 and the follower outputs logical 0 during an ACK bit period
  Initiator: Transmits a NACK
  Follower: Transmits an ACK

Figure 32.3 shows an example of outputting the ACK bit.

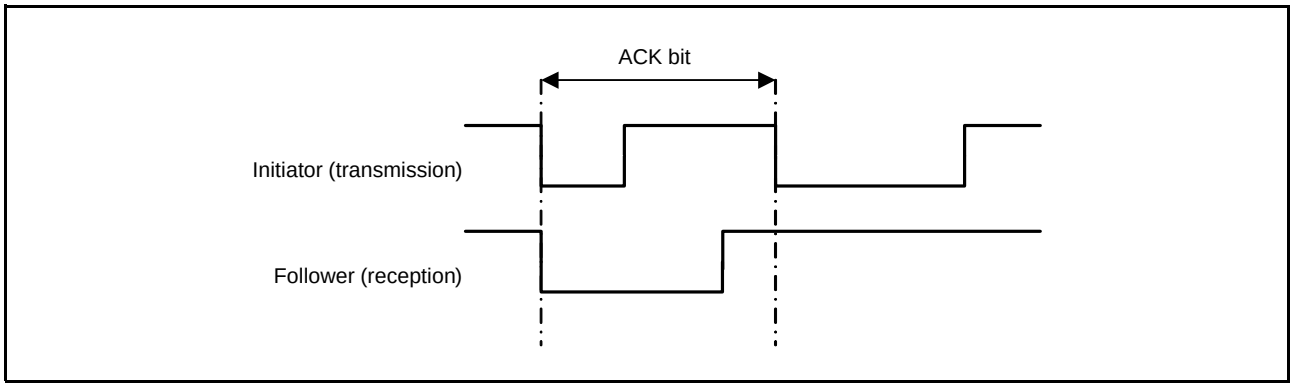


Figure 32.3 Example of Outputting ACK Bit from Initiator/Follower

32.2 Register Descriptions

32.2.1 CEC Local Address Setting Register (CADR)

Address(es) 000A 0A00h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADR14	ADR13	ADR12	ADR11	ADR10	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02	ADR01	ADR00
Value after 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	ADR00	Local Address at Address 0 (TV)	0: Does not set as local address. 1: Sets as local address.	R/W
b1	ADR01	Local Address Setting at Address 1 (Recording Device 1)	0: Does not set as local address. 1: Sets as local address.	R/W
b2	ADR02	Local Address Setting at Address 2 (Recording Device 2)	0: Does not set as local address. 1: Sets as local address.	R/W
b3	ADR03	Local Address Setting at Address 3 (Tuner 1)	0: Does not set as local address. 1: Sets as local address.	R/W
b4	ADR04	Local Address Setting at Address 4 (Playback Device 1)	0: Does not set as local address. 1: Sets as local address.	R/W
b5	ADR05	Local Address Setting at Address 5 (Audio System)	0: Does not set as local address. 1: Sets as local address.	R/W
b6	ADR06	Local Address Setting at Address 6 (Tuner 2)	0: Does not set as local address. 1: Sets as local address.	R/W
b7	ADR07	Local Address Setting at Address 7 (Tuner 3)	0: Does not set as local address. 1: Sets as local address.	R/W
b8	ADR08	Local Address Setting at Address 8 (Playback Device 2)	0: Does not set as local address. 1: Sets as local address.	R/W
b9	ADR09	Local Address Setting at Address 9 (Recording Device 3)	0: Does not set as local address. 1: Sets as local address.	R/W
b10	ADR10	Local Address Setting at Address 10 (Tuner 4)	0: Does not set as local address. 1: Sets as local address.	R/W
b11	ADR11	Local Address Setting at Address 11 (Playback Device 3)	0: Does not set as local address. 1: Sets as local address.	R/W
b12	ADR12	Local Address Setting at Address 12 (Reserved)	0: Does not set as local address. 1: Sets as local address.	R/W
b13	ADR13	Local Address Setting at Address 13 (Reserved)	0: Does not set as local address. 1: Sets as local address.	R/W
b14	ADR14	Local Address Setting at Address 14 (Specific Use)	0: Does not set as local address. 1: Sets as local address.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: To specify address 15 (unregistered) as the CEC local address, clear the ADR00 to ADR14 bits to 0.
Note: Do not rewrite the set value during a communication (CECS.BUSST = 1).

CADR is a 16-bit register that sets local addresses. It is valid only during a reception, the ADR00 to ADR14 bits correspond to CEC logical addresses 0 to 14, and up to 15 local addresses can be set. To specify address 15 as the CEC local address, clear the ADR00 to ADR14 bits to 0. A broadcast address always operates as a local address.
For example, when using addresses 0 as local addresses, set the ADR00 bit to 1.

32.2.2 CEC Transmission Buffer Register (CTXD)

Address(es) 000A 0A30h

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

CTXD is an 8-bit register that sets transmit data. It sequentially transmits 8 bits of data, starting from bit 7 with MSB first. A transmission/reception interrupt request signal (INTDA) is generated at the start timing of the header block and data block. Successive transmission can be performed by writing the next data to the CTXD register before the transmission ends after INTDA was generated.
If an underrun error occurs (UERR = 1), transmission is not continued. An error interrupt is generated and the transmission wait state is entered.
If transmit data is written to this register after a data interrupt (INTDA) was generated during transmission of the last block, the data will be invalid.

32.2.3 CEC Reception Buffer Register (CRXD)

Address(es) 000A 0A31h

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

CRXD is an 8-bit register that retains receive data.
Receive data can be read by reading this register.
For every byte of data received, new data will be transferred from the CEC reception shift register.
If an overrun error occurs (CECES.OERR = 1), the data of the reception buffer register will be overwritten.

32.2.4 CEC Control Register 0 (CECCTL0)

Address(es) 000A 0A35h

	b7	b6	b5	b4	b3	b2	b1	b0
	CECE	ACKTEN	CCL[2:0]			TXTRG	CECRXEN	EOM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EOM	EOM Setting	0: Continues transmission. 1: Last frame	R/W
b1	CECRXEN	Reception Enable Control	0: Disables reception. 1: Enables reception. Table 32.3 lists the reception status and ACK/NACK timing output.	R/W
b2	TXTRG	Transmission Start Trigger	0: Does not start CEC transmission. 1: Starts CEC transmission.	R/W
b5 to b3	CCL[2:0]	CEC Clock (CECCLK) Select *1	b5 b3 0 0 0: PCLKB/32 0 0 1: PCLKB/64 0 1 0: PCLKB/128 0 1 1: PCLKB/256 1 0 0: PCLKB/512 1 0 1: PCLKB/1024 1 1 0: CECILCLK/4 1 1 1: CECMCLK/256	R/W
b6	ACKTEN	ACK Bit Timing Error (Bit Width) Check Enable *1	0: Does not detect ACK bit timing errors. 1: Detects ACK bit timing errors.	R/W
b7	CECE	CEC Operation Enable Flag	0: Stops CEC operation. 1: Enables CEC operation.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

CECCTL0 selects enabling operation, starting transmission, and the operating clock.

EOM Bit (EOM Setting)

The next frame after 1 is written to this bit will be the last frame.
Set the EOM bit before writing transmit data to the transmission buffer register (CTXD).
Writing to the CECCTL0.EOM bit and the transmission buffer register (CTXD) should be set after a data interrupt (INTDA) is generated and before the next data interrupt (INTDA) is generated (timing at which the EOM bit transmission is completed). However, this setting is not required when the data interrupt (INTDA) for the last block is generated.

CECRXEN Bit (Reception Enable Control)

Setting this bit to 1 enables reception. Set this bit to 1 after determining the local address (setting the CADR register). Do not rewrite the set value during a communication (CECS.BUSST = 1).

Table 32.3 Reception Status and ACK/NACK Timing Output According to Reception Enable Control Bit Status

CECRXEN	Reception Enable Control Bit		
1	Enables continuing reception or reports normal reception.		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	ACK
		Timing error occurrence	NACK
	During broadcast reception	Normal reception	NACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high-impedance)	
0	Stops continuing reception or reports abnormal reception.		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	NACK
		Timing error occurrence	NACK
	During broadcast reception	Normal reception	ACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high-impedance)	

TXTRG Bit (Transmission Start Trigger)

Setting this bit to 1 starts transmission.

This bit is a trigger bit and the read value is 0.

Set CECCTL0.TXTRG to 1 only when CEC operation is enabled (CECCTL0.CECE = 1) and the bus is free (CECS.BUSST = 0). Transmission starts no more than three CECCLK cycles after CECCTL0.TXTRG is set to 1.

CCL[2:0] Bits (CEC Clock (CECCLK) Select)

These bits are used to select the CEC clock. Set the frequency of the CEC clock (CECCLK) in the range of 23.4375 to 78.125 kHz. Examples of CECCLK settings are shown in Table 32.4 to Table 32.6.

Table 32.4 Examples of CEC Clock Settings

	CEC Clock (CECCLK)						
CEC Clock Selected	**When PCLKB = 32 MHz**	**When PCLKB = 20 MHz**	**When PCLKB = 16 MHz**	**When PCLKB = 12 MHz**	**When PCLKB = 10 MHz**	**When PCLKB = 8 MHz**	**When PCLKB = 2 MHz**
PCLKB/32	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	62.5 kHz
PCLKB/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	31.25 kHz
PCLKB/128	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	78.125 kHz	62.5 kHz	Setting prohibited
PCLKB/256	Setting prohibited	78.125 kHz	62.5 kHz	46.875 kHz	39.0625 kHz	31.25 kHz	Setting prohibited
PCLKB/512	62.5 kHz	39.0625 kHz	31.25 kHz	23.4375 kHz	Setting prohibited	Setting prohibited	Setting prohibited
PCLKB/1024	31.25 kHz	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

Table 32.5 Examples of CEC Clock Settings (CECICLK Selected)

CEC Clock Selected	CEC Clock (CECCLK) CECILCLK = 125 kHz (typ)
CECICLK/4	31.25 kHz

Table 32.6 Examples of CEC Clock Settings (CECMCLK Selected)

CEC Clock Selected	CEC Clock (CECCLK) When main clock = 20 MHz	When main clock = 16 MHz	When main clock = 12 MHz	When main clock = 10 MHz	When main clock = 8 MHz
CECMCLK/256	78.125 kHz	62.5 kHz	46.875 kHz	39.0625 kHz	31.25 kHz

ACKTEN Bit (ACK Bit Timing Error (Bit Width) Check Enable)

When this bit is set to 1, timing errors are detected for the bit width (the values specified for DATBL and DATBH), in addition to the low-level width of the ACK bit (the values specified for LGC0LL, LGC0LH, LGC1LL, and LGC1LH). However, the maximum bit width (DATBH) is not checked for the ACK bit of the last frame (EOM = 1) even if ACKTEN is 1.

CECE Bit (CEC Operation Enable Flag)

Setting this bit to 1 enables CEC operation. Setting this bit to 0 resets the internal block but does not reset control registers.

32.2.5 CEC Control Register 1 (CECCTL1)

Address(es) 000A 0A02h

	b7	b6	b5	b4	b3	b2	b1	b0
	CDFC	CINTMK	BLERRD	STERRD	CESEL[1:0]		SFT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SFT[1:0]	Signal-Free Time Data Bit Width Select	b1 b0 0 0: 3-data bit width 0 1: 5-data bit width 1 0: 7-data bit width 1 1: Does not detect signal-free time.	R/W
b3, b2	CESEL[1:0]	Communication Complete Interrupt (INTCE) Generation Timing Select *1	b3 b2 0 0: Generates communication complete interrupt once after ACK transmission (reception) of the last frame (EOM = 1) is completed and another time after signal-free time is detected. 0 1: Generates communication complete interrupt after ACK transmission (reception) of the last frame (EOM = 1) is completed. 1 0: Generates communication complete interrupt after signal-free time is detected. 1 1: Setting prohibited	R/W
b4	STERRD	Start Bit Error Detection Select *1	0: Does not detect timing errors during start bit reception. 1: Detects timing errors during start bit reception.	R/W
b5	BLERRD	Bus Lock Detection Select*1	0: Does not detect sticking of receive data to high or low level. 1: Detects sticking of receive data to high or low level.	R/W
b6	CINTMK	CEC Data Interrupt (INTDA) Generation Select*1	0: Does not generate an interrupt when the addresses do not match. 1: Generates an interrupt when the addresses do not match.	R/W
b7	CDFC	Digital Filter Select*1	0: Does not use a digital filter. 1: Uses a digital filter.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

CECCTL1 selects a digital filter, data interrupt generation, a start bit error interrupt, and whether to generate a communication complete interrupt and when to generate it.

SFT[1:0] Bits (Signal-Free Time Data Bit Width Select)

The bit width of the signal-free time is selected by setting the SFT[1:0] bits.
The bit width of the signal-free time is set in the NOMP register.
Rewrite these bits only after confirming that the signal-free time rewrite disable report flag (CECS.SFTST) is 0.

CESEL[1:0] Bits (Communication Complete Interrupt (INTCE) Generation Timing Select)

The timing at which a communication complete interrupt (INTCE) is generated is selected by setting the CESEL[1:0] bits.

STERRD Bit (Start Bit Error Detection Select)

This bit can be used to detect timing errors during start bit reception according to the set value of the STATLL, STATLH, STATBL, or STATBH register.
Timing errors of the start bit can be detected according to the set value of the STATLL, STATLH, STATBL, or STATBH register by setting CECCTL1.STERRD to 1. If a timing error occurred, the start bit for which the error occurred is determined to be disabled and the communication standby state is entered.

If CECCTL1.STERRD is 0, no timing error is detected. All pulses are determined to be start bits.

BLERRD Bit (Bus Lock Detection Select)

The bus lock status of the CEC line can be detected by setting this bit to 1. If the next falling edge is not input for a period 2.5 times the 1-data bit width set with the NOMP register in a state in which the falling edge of the CEC line is awaited (excluding the communication standby state), an error interrupt (INTERR) is generated and a bus lock error detection flag (BLERR) is set. Afterward, the communication standby state is entered.

CINTMK Bit (CEC Data Interrupt (INTDA) Generation Select)

This bit is used to select whether to generate INTDA of the header block when the destination address and local address do not match during a reception and whether to generate INTCE when communication is completed.

CDFC Bit (Digital Filter Select)

This bit is used to eliminate noise from one CECCLK cycle using a digital filter.

32.2.6 CEC Communication Status Register (CECS)

Address(es) 000A 0A33h

	b7	b6	b5	b4	b3	b2	b1	b0
	SFTST	—	—	ITCEF	EOMF	TXST	BUSST	ADRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADRF	Address Match Detection Flag	0: During a communication between other stations, while communication is stopped, or while the local station is transmitting 1: During local reception	R
b1	BUSST	Bus Busy Detection Flag	0: Bus-free state 1: Bus busy state	R
b2	TXST	Transmission Status Flag	0: During the communication standby state or reception (A follower is operating.) 1: During transmission (The initiator is operating.)	R
b3	EOMF	EOM Flag	0: The EOM bit received immediately before is logically 0. 1: The EOM bit received immediately before is logically 1.	R
b4	ITCEF	INTCE Generation Source Flag	0: Generates a communication complete interrupt (INTCE) if the signal-free time has been counted. 1: Generates INTCE if communication is completed or an error is detected.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SFTST	Signal-Free Time Rewrite Disable Report Flag	0: Enables rewriting CECCTL1.SFT[1:0]. 1: Disables rewriting CECCTL1.SFT[1:0].	R

CECS indicates the CEC communication status.

ADRF Flag (Address Match Detection Flag)

This flag can be used to check whether communication is addressed to the local station.

[Setting conditions]

- When the local address and reception destination address match
- When a broadcast address is received

[Clearing conditions]

- When CEC operation is stopped (CECCTL0.CECE = 0)
- When reception is completed

BUSST Flag (Bus Busy Detection Flag)

This flag can be used to check the CEC bus state.

[Setting conditions]

- When a fall of the CEC line is detected
- When "CEC operation is stopped" is changed to "CEC operation is enabled" (CECCTL0.CECE is changed from 0 to 1)

[Clearing conditions]

- When CEC operation is stopped (CECCTL0.CECE = 0)
- When the signal-free time has elapsed after the communication has ended

TXST Flag (Transmission Status Flag)

This flag can be used to check whether or not transmission is in progress.

EOMF Flag (EOM Flag)

This flag can be used to check the value of the bit received immediately before. The value is updated each time a data interrupt (INTDA) is generated.

ITCEF Flag (INTCE Generation Source Flag)

By checking ITCEF after INTCE is generated, it can be determined which case is the interrupt generation source that the signal-free time has been counted, communication ends, or an error is detected. This setting is enabled only if CECCTL1.CESEL[1:0] are cleared to 0.

SFTST Flag (Signal-Free Time Rewrite Disable Report Flag)

This flag can be used to check whether rewriting CECCTL1.SFT[1:0] is enabled or disabled.

[Setting condition]

- When a write access to CECCTL1 is performed

[Clearing conditions]

- When CEC operation is stopped (CECCTL0.CECE = 0)
- When the CECCTL1.SFT[1:0] rewrite disable period has elapsed

32.2.7 CEC Communication Error Status Register (CECES)

Address(es) 000A 0A32h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BLERR	AERR	TXERR	TERR	ACKERR	UERR	OERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OERR	Overrun Error Detection Flag	0: No overrun error has occurred. 1: An overrun error has occurred.	R
b1	UERR	Underrun Error Detection Flag	0: No underrun error has occurred. 1: An underrun error has occurred.	R
b2	ACKERR	ACK Error Detection Flag	0: No ACK error has occurred. 1: An ACK error has occurred.	R
b3	TERR	Timing Error Detection Flag	0: No timing error has occurred. 1: A timing error has occurred.	R
b4	TXERR	Transmission Error Detection Flag*1	0: No transmission error has occurred. 1: A transmission error has occurred.	R
b5	AERR	Arbitration Loss Detection Flag	0: No arbitration loss has occurred. 1: An arbitration loss has occurred	R
b6	BLERR	Bus Lock Error Detection Flag	0: No bus lock error has occurred. 1: A bus lock error has occurred.	R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Transmission errors are not detected while transmitting the start bit and ACK bit.

CECES indicates whether a bus lock error, arbitration loss, transmission error, timing error, ACK error, underrun error, or overrun error is detected.

OERR Flag (Overrun Error Detection Flag)

The flag can be used to check whether an overrun has occurred.

[Setting condition]

- When the next receive operation is completed before the receive data stored in the reception buffer register (CRXD) is read

[Clearing condition]

- When 1 is written to CECFC.OCTRG

UERR Flag (Underrun Error Detection Flag)

The flag can be used to check whether an underrun has occurred.

[Setting condition]

- When transmit data is not written to the transmission buffer register (CTXD) after a data interrupt (INTDA) is generated and before the next interrupt (INTDA) is generated

[Clearing condition]

- When 1 is written to CECFC.UCTRG

ACKERR Flag (ACK Error Detection Flag)

The flag can be used to check whether an ACK error has occurred.

[Setting conditions]

- When logical 1 is received at the ACK bit timing during a direct address communication

- When logical 0 is received at the ACK bit timing during a broadcast communication
- When logical 1 is received at the ACK bit timing during a logical address allocation transmission

[Clearing condition]

- When 1 is written to CECFC.ACKCTRG

TERR Flag (Timing Error Detection Flag)

The flag can be used to check whether a timing error has occurred.

[Setting condition]

- When a violation is detected in the timing check of the received data

[Clearing condition]

- When 1 is written to CECFC.TCTRG

TXERR Flag (Transmission Error Detection Flag)

The flag can be used to check whether a transmission error has occurred.

[Setting condition]

- When the logic of the transmit data and receive data are compared and do not match when the initiator is operating

[Clearing condition]

- When 1 is written to CECFC.TXCTRG

AERR Flag (Arbitration Loss Detection Flag)

The flag can be used to check whether arbitration is lost.

[Setting condition]

- When arbitration loss occurs between start bit transmission and initiator address transmission

[Clearing condition]

- When 1 is written to CECFC.ACTRG

BLERR Flag (Bus Lock Error Detection Flag)

The flag can be used to check whether a bus lock error has occurred.

[Setting condition]

- When the next falling edge is not input for a time 2.5 times the 1-data bit width set by the NOMP register after the falling edge of the CEC reception signal while the CEC reception signal is fixed to low or high level midway through a frame

[Clearing condition]

- When 1 is written to CECFC.BLCTRG

32.2.8 CEC Communication Error Flag Clear Trigger Register (CECFC)

Address(es) 000A 0A34h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BLCTRG	ACTRG	TXCTRG	TCTRG	ACKCTRG	UCTRG	OCTRG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OCTRG	Overrun Error Detection Flag Clear Trigger*1	0: Does not clear overrun error detection flag. 1: Clears overrun error detection flag.	W
b1	UCTRG	Underrun Error Detection Flag Clear Trigger*1	0: Does not clear underrun error detection flag. 1: Clears underrun error detection flag.	W
b2	ACKCTRG	ACK Error Detection Flag Clear Trigger*1	0: Does not clear ACK error detection flag. 1: Clears ACK error detection flag.	W
b3	TCTRG	Timing Error Detection Flag Clear Trigger*1	0: Does not clear timing error detection flag. 1: Clears timing error detection flag.	W
b4	TXCTRG	Transmission Error Detection Flag Clear Trigger*1	0: Does not clear transmission error detection flag. 1: Clears transmission error detection flag.	W
b5	ACTRG	Arbitration Loss Detection Flag Clear Trigger*1	0: Does not clear arbitration loss detection flag. 1: Clears arbitration loss detection flag.	W
b6	BLCTRG	Bus Lock Error Detection Flag Clear Trigger*1	0: Does not clear bus lock error detection flag. 1: Clears bus lock error detection flag.	W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The read value is 0.

CECFC clears error detection flags written to the communication error status register (CECES). Only the set bits can be cleared by setting 1 to each flag.

32.2.9 CEC Transmission Start Bit Low Width Setting Register (STATL)

Address(es) 000A 0A06h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATL[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	STATL[8:0]	CEC Transmission Start Bit Low Width Setting*1	Low-level width of the start bit during a transmission Low-level width = (set value of STATL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.10 CEC Transmission Start Bit Width Setting Register (STATB)

Address(es) 000A 0A04h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATB[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	STATB[8:0]	CEC Transmission Start Bit Width Setting*1	Bit width of the start bit during a transmission Bit width = (set value of STATB[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.11 CEC Transmission Logical 0 Low Width Setting Register (LGC0L)

Address(es) 000A 0A08h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC0L[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	LGC0L[8:0]	CEC Transmission Logical 0 Low Width Setting*1	Low-level width of logical 0 during a transmission Low-level width = (set value of LGC0L[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.12 CEC Transmission Logical 1 Low Width Setting Register (LGC1L)

Address(es) 000A 0A0Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC1L[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	LGC1L[8:0]	CEC Transmission Logical 1 Low Width Setting*1	Low-level width of logical 1 during a transmission Low-level width = (set value of LGC1L[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.13 CEC Transmission Data Bit Width Setting Register (DATB)

Address(es) 000A 0A0Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DATB[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	DATB[8:0]	CEC Transmission Data Bit Width Setting*1	Bit width of the data bit during a transmission 1-data bit width = (set value of DATB[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.14 CEC Reception Start Bit Minimum Low Width Setting Register (STATLL)

Address(es) 000A 0A10h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATLL[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	STATLL[8:0]	CEC Reception Start Bit Minimum Low Width Setting*1	Minimum value of the low-level width of the start bit during a reception Low-level width = (set value of STATLL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The value of this register is not used when the timing of the start bit is not checked (CECCTL1.STERRD = 0) and restarting reception upon start detection is not enabled (CECEXMOD.RERCVEN = 0). The set value will be invalid.

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.15 CEC Reception Start Bit Maximum Low Width Setting Register (STATLH)

Address(es) 000A 0A12h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATLH[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	STATLH[8:0]	CEC Reception Start Bit Maximum Bit Width Setting*1	Maximum value of the low-level width of the start bit during a reception Low-level width = (set value of STATLH[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The value of this register is not used when the timing of the start bit is not checked (CECCTL1.STERRD = 0) and restarting reception upon start detection is not enabled (CECEXMOD.RERCVEN = 0). The set value will be invalid.
Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.16 CEC Reception Start Bit Minimum Bit Width Setting Register (STATBL)

Address(es) 000A 0A14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATBL[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	STATBL[8:0]	CEC Reception Start Bit Minimum Bit Width Setting*1	Minimum value of the bit width of the start bit during a reception Bit width = (set value of STATBL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The value of this register is not used when the timing of the start bit is not checked (CECCTL1.STERRD = 0) and restarting reception upon start detection is not enabled (CECEXMOD.RERCVEN = 0). The set value will be invalid.
Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.17 CEC Reception Start Bit Maximum Bit Width Setting Register (STATBH)

Address(es) 000A 0A16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STATBH[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	STATBH[8:0]	CEC Reception Start Bit Maximum Bit Width Setting*1	Maximum value of the bit width of the start bit during a reception Bit width = (set value of STATBH[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The value of this register is not used when the timing of the start bit is not checked (CECCTL1.STERRD = 0) and restarting reception upon start detection is not enabled (CECEXMOD.RERCVEN = 0). The set value will be invalid.

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.18 CEC Reception Logical 0 Minimum Low Width Setting Register (LGC0LL)

Address(es) 000A 0A18h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC0LL[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	LGC0LL[8:0]	CEC Reception Logical 0 Minimum Low Width Setting*1	Minimum value of the low-level width of logical 0 during a reception Low-level width = (set value of LGC0LL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.19 CEC Reception Logical 0 Maximum Low Width Setting Register (LGC0LH)

Address(es) 000A 0A1Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC0LH[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	LGC0LH[8:0]	CEC Reception Logical 0 Minimum Low Width Setting*1	Maximum value of the low-level width of logical 0 during a reception Low-level width = (set value of LGC0LH[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.20 CEC Reception Logical 1 Minimum Low Width Setting Register (LGC1LL)

Address(es) 000A 0A1Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC1LL[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	LGC1LL[8:0]	CEC Reception Logical 1 Minimum Low Width Setting*1	Minimum value of the low-level width of logical 1 during a reception Low-level width = (set value of LGC1LL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.21 CEC Reception Logical 1 Maximum Low Width Setting Register (LGC1LH)

Address(es) 000A 0A1Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LGC1LH[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	LGC1LH[8:0]	CEC Reception Logical 1 Maximum Low Width Setting*1	Maximum value of the low-level width of logical 1 during a reception Low-level width = (set value of LGC1LH[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.22 CEC Reception Data Bit Minimum Bit Width Setting Register (DATBL)

Address(es) 000A 0A20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DATBL[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	DATBL[8:0]	CEC Reception Data Bit Minimum Bit Width Setting*1	Minimum value of the bit width of the data bit during a reception Bit width = (set value of DATBL[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.23 CEC Reception Data Bit Maximum Bit Width Setting Register (DATBH)

Address(es) 000A 0A22h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DATBH[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	DATBH[8:0]	CEC Reception Data Bit Maximum Bit Width Setting*1	Maximum value of the bit width of the data bit during a reception Bit width = (set value of DATBH[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

32.2.24 CEC Reception Data Sampling Time Setting Register (NOMT)

Address(es) 000A 0A0Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NOMT[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	NOMT[8:0]	CEC Reception Data Sampling Time Setting*1, *2	Sampling time of received data Sampling time = (set value of NOMT[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).
Note 2. Set this register within a period of LGC1LH < NOMT < LGC0LL.

32.2.25 CEC Data Bit Reference Width Setting Register (NOMP)

Address(es) 000A 0A24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NOMP[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	NOMP[8:0]	CEC Data Bit Reference Width Setting*1	1-data bit width Bit width = (set value of NOMP[8:0] bits + 1) × clock cycle of CECCLK	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

This 1-data bit width is used when counting the number of bits for errors handling, the signal-free time, and bus locking detection.

32.2.26 CEC Extension Mode Register (CECEXMD)

Address(es) 000A 0A28h

	b7	b6	b5	b4	b3	b2	b1	b0
	RCVINTDSEL	—	RERCVEN	LERPLEN	—	—	—	—
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	LERPLEN	Pulse Output Function Enable by Long Bit Width Error*1	0: Detects only a long bit width error. 1: Detects a long bit width error and outputs an error handling pulse.	R/W
b5	RERCVEN	Start Detection Reception Restart Enable*1	0: Does not restart reception when the start bit is detected during reception. 1: Restarts reception when the start bit is detected during reception.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	RCVINTDSEL	INTDA Reception Interrupt Timing Change*1	0: EOM timing (9th bit of data) 1: ACK timing (10th bit of data)	R/W

Note 1. Rewritable only when CEC operation is stopped (CECCTL0.CECE = 0).

This register is used to control enabling of error handling when a long bit error is detected, restarting of reception by detecting a start bit, and selecting of the timing for generating reception interrupts.

LERPLEN Bit (Pulse Output Function Enable by Long Bit Width Error)

This bit is used to control outputting of an error handling pulse when a long bit width error is detected.

RERCVEN Bit (Start Detection Reception Restart Enable)

This bit is used to control restarting of reception when the start bit is detected.

RCVINTDSEL Bit (INTDA Reception Interrupt Timing Change)

This bit is used to change the timing for generating reception interrupts.

32.2.27 CEC Extension Monitor Register (CECEXMON)

Address(es) 000A 0A2Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ACKF	CECLNMON
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Bit Name	Description	R/W
b0	CECLNMON	CEC Line Monitor*1	The CEC line can be read. 0: Low level 1: High level	R
b1	ACKF	ACK Flag*2	The value of the received ACK bit can be read. The value is updated at the timing of ACK reception regardless of transmission/reception or address match/mismatch during reception. However, the value is not updated if an error is detected before an ACK is received. The value is updated when an ACK is received at the next communication.	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using the value of the CECEXMON.CECLNMON bit, determine the value after matching two or three times.
Note 2. When using the CECEXMON.ACKF bit, the relationship between the read timing and the read value changes depending on the value of the CECEXMD.RCVINTDSEL bit.

This register can be used to read the CEC line and the ACK flag.

CECLNMON Bit (CEC Line Monitor)

The CEC pin state can be read by reading this bit. When using the value of this bit, determine the value after matching two or three times.

ACKF Flag (ACK Flag)

The value of the received ACK bit can be read by reading this bit. When using this bit, the relationship between the read timing and the read value changes depending on the value of the CECEXMD.RCVINTDSEL bit.
Figure 32.4 shows the ACKF update timing and reception interrupt generation timing.

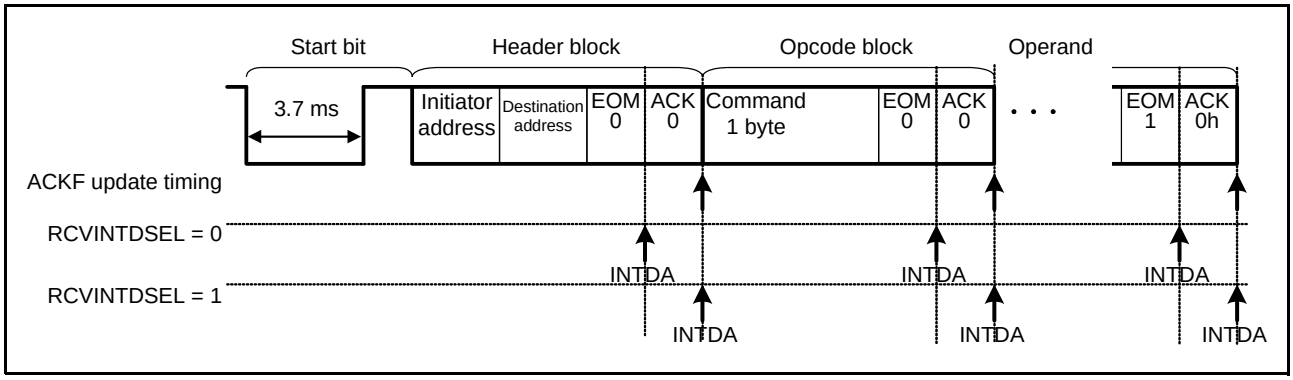


Figure 32.4 ACKF Update Timing and Reception Interrupt Generation Timing

When reading the ACKF bit while RCVINTDSEL = 0, the received ACK state can be read by reading this bit when a 1-bit wait time has elapsed after a reception interrupt is generated. (If this bit is read after a reception interrupt is generated, the ACK of the previously received data is read.)
When reading the ACKF bit while RCVINTDSEL = 1, read this bit after a reception interrupt is generated. The ACK of the newest received data can be read.

32.2.28 CEC Interrupt Control Register i (CECINTCRi) (i = 1 to 3)

Address(es) 0008 750Dh to 0008 750Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CECINTMD[1:0]		—	—
Value after reset:	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
b3, b2	CECINTMD[1:0]	CEC Interrupt Detection Setting	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Both edges	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set how to detect INTDAA, INTCEA, and INTERRA. CECINTCR1 corresponds to INTDAA, CECINTCR2 corresponds to INTCEA, and CECINTCR3 corresponds to INTERRA. Only change the settings of this register while the interrupt request is disabled by the corresponding interrupt request enable bit (IERm.IENj bit is 0). After changing the settings, clear the IR flag before setting the interrupt enable bit.

CECINTMD[1:0] Bits (CEC Interrupt Detection Setting)

Since the interrupt request signal issued from the CEC is active-low, set these bits to a falling edge.

32.3 Operation

32.3.1 Operation of CEC Transmission/Reception Circuit

32.3.1.1 CEC Transmission/Reception Data Format

Figure 32.5 shows the basic CEC communication format. A CEC data frame consists of a start bit, a header block, data block 1 (opcode), and data block 2 (operand). The three blocks other than the start bit are configured of 10 bits.

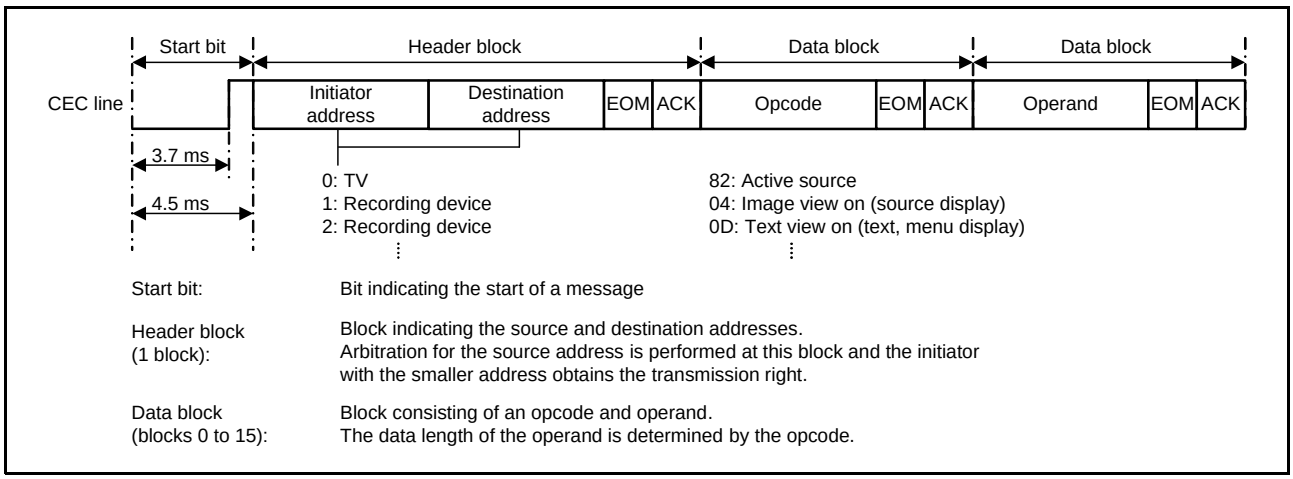


Figure 32.5 CEC Communication Format

32.3.1.2 Communication Types

CEC transmission/reception takes place in the state of a direct address message or broadcast message. In a CEC communication, the transmitting side transmits a start bit and message (data) and the receiving side receives the message and returns a desired acknowledge signal to the transmitting side. CEC transmission/reception is configured of a start bit and a data bit and performs all transmissions/receptions of CEC.

32.3.1.3 Bit Timing

Figure 32.6 shows an example of the pulse format of a start bit. Whether the start bit is valid is judged at the low-level period (a) and bit period (b).

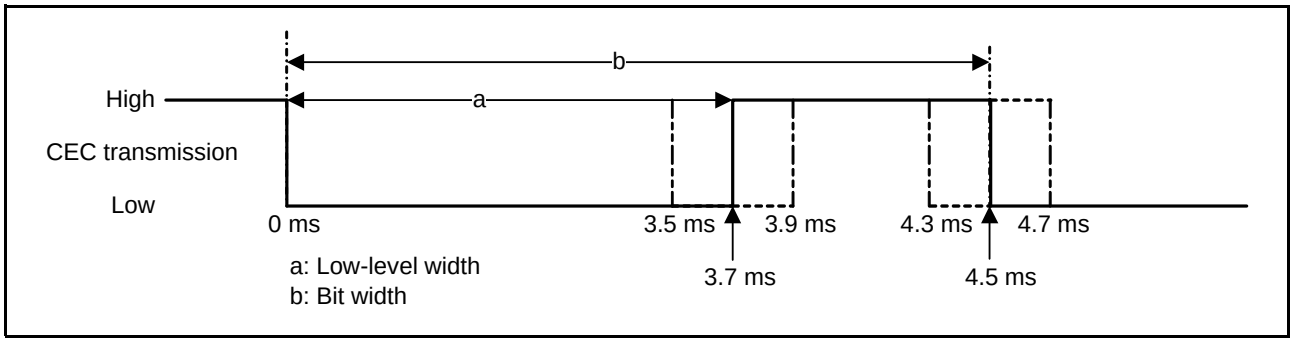
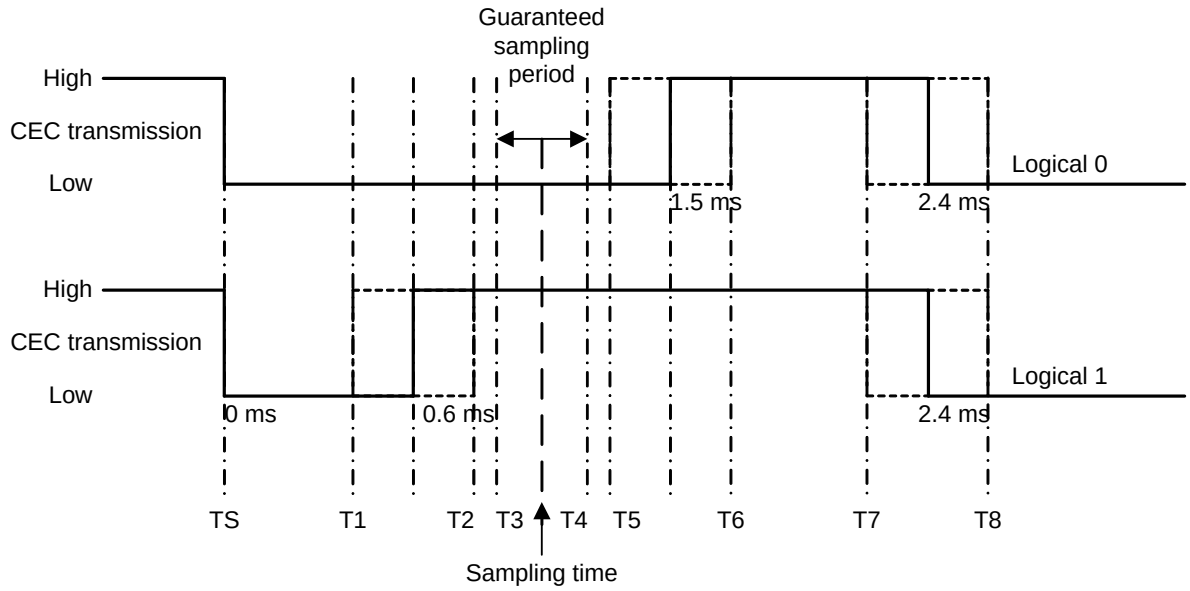


Figure 32.6 Start Bit Format Example

Figure 32.7 shows an example of the pulse format of a data bit timing. A data bit is sampled at a sampling timing set with the CEC reception data sampling time setting register (NOMT). If the result of sampling is low level, the pulse format is logical 0. If the result of sampling is high level, the pulse format is logical 1. The last change from high to low level of a data bit is the start of the next data bit. Consequently, the last data bit remains at high level.



Ts	0 ms	The bit start event
T1	0.4 ms	The earliest time for a low-high transition when indicating a logical 1
T2	0.8 ms	The latest time for a low-high transition when indicating a logical 1
T3	0.85 ms	The earliest time for safely sampling the signal line to determine its state.
T4	1.25 ms	The latest time for safely sampling the signal line to determine its state
T5	1.3 ms	The earliest time at which a device is permitted to return to a high impedance state (logical 0)
T6	1.7 ms	The latest time at which a device is permitted to return to a high impedance state (logical 0)
T7	2.05 ms	The earliest time for the start of a following bit
	2.4 ms	The nominal data bit period
T8	2.75 ms	The latest time for the start of a following bit

Figure 32.7 Data Bit Format Example

32.3.1.4 Header Block/Data Block

All data blocks are configured of 10 bits and have the same structure. Figure 32.8 shows the format of header and data blocks. An information bit has a different meaning for a header block and data block, and indicates the data, opcode, and address. EOM (End of Message) and ACK (Acknowledge) are control bits and have the same meanings for a header block and data block.

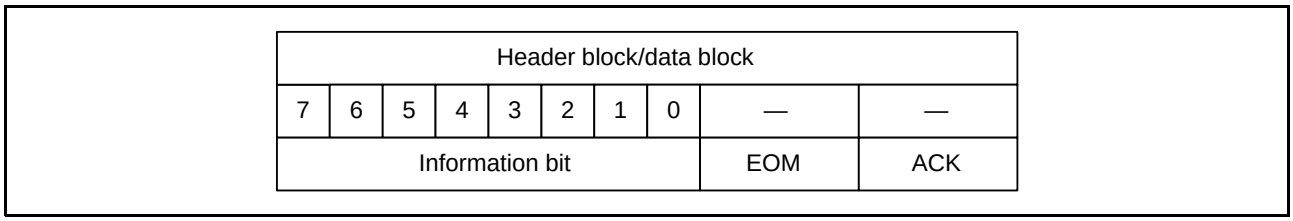


Figure 32.8 Header Block and Data Block Format

A header block consists of an initiator logical address, destination logical address, EOM (End of Message), and ACK (Acknowledge). Information bits 7 to 4 indicate initiator logical addresses and bits 3 to 0 indicate destination logical addresses. The EOM of a header block is used for “ping” with another device (checking whether the power of another device is turned on). “ping” can be checked by setting EOM = 1 and transmitting only the header block (transmitting a message without data blocks). In the case of direct address transmission, the power of the device to which the header block is transmitted is turned on if an ACK is returned.

32.3.1.5 EOM (End of Message)

An EOM indicates whether the transmitted block is the last block of a message. It is added to an information bit and output.
EOM bit = 0: When one block or multiple blocks follow
EOM bit = 1: When the transmitted block is the last block

32.3.1.6 ACK (Acknowledge)

The meaning of an ACK depends on whether the receiving party of a transmission is a direct address message or broadcast message. The result of comparing the received data and CEC line data is transmitted to the transmitting side as an ACK or NACK.
The initiator outputs logical 1 at the ACK bit timing. Consequently, a follower determines the logic level of the ACK bit.

- An ACK (ACK = logical 0) is the normal value for a direct address message.

(1) If no error exists in the header block and the local address is used, the ACK bit is logical 0.
(2) If no error exists in the data blocks, the ACK bit is logical 0.
(3) If an error exists in the header block or another address is used, the ACK bit is logical 1.
(4) If an error exists in the data blocks, the ACK bit is logical 1.

- An NACK (ACK = logical 1) is the normal value for a broadcast message.

(1) If one or more followers have abandoned the message, the ACK bit is logical 0.
(2) If no followers have abandoned the message, the ACK bit is logical 1.

32.3.2 Operating Clocks

The CEC can use one of the following clocks as its operating clock: the divided clock of the peripheral module clock (PCLKB), the divided clock of the CECILCLK supplied from the IWDT-dedicated on-chip oscillator, or the divided clock of the CECMCLK supplied from the main clock oscillator. When using the CECILCLK or CECMCLK as the CEC operating clock, take note of the respective procedures for supplying these clocks. The following describes how to supply these clocks.

32.3.2.1 When Using CECILCLK as CEC Operating Clock

This section describes the flow for using the divided clock of the CECILCLK supplied from the IWDT-dedicated on-chip oscillator.

When the ILOCOCR.ILCSTP bit is set to 0, the IWDT-dedicated on-chip oscillator starts operating. After oscillation starts, the operating clock is supplied to the CEC when the oscillation stabilization wait time has elapsed. When continuing operation of the IWDT-dedicated on-chip oscillator in software standby mode, set the IWDTCSTPR.SLCSTP bit to 0. For details on the ILOCOCR register, see section 9.2.8, IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR). For details on the IWDTCSTPR register, see section 29.2.5, IWDT Count Stop Control Register (IWDTCSTPR).

Note that the operating clock is also supplied to the IWDT while the IWDT-dedicated on-chip oscillator is operating. When using the divided clock of the CECILCLK as the CEC operating clock, do not use the IWDT function in order to prevent an unexpected reset or an interrupt from being generated.

Figure 32.9 shows an example of the flowchart for starting the CECILCLK supply to the CEC.

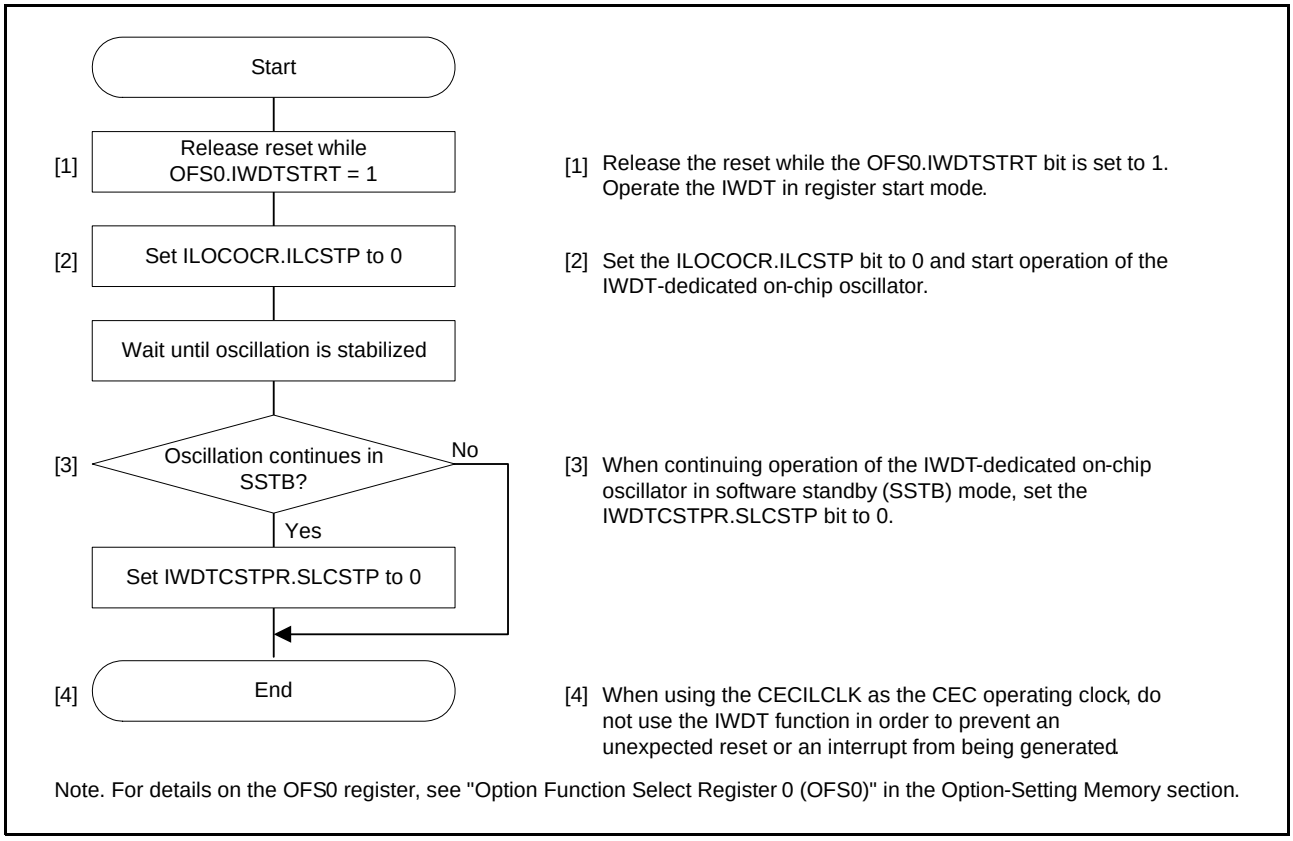


Figure 32.9 Example of Flowchart for Starting CECILCLK Supply to CEC

32.3.2.2 When Using CECMCLK as CEC Operating Clock

This section describes the flow for using the divided clock of the CECMCLK supplied from the main clock oscillator as the CEC operating clock.

When the MOSCCR.MOSTP bit is set to 0, the main clock oscillator starts operating.

Although oscillation can also be started by setting the MOFCR.MOFXIN bit to 1, if this bit is used to start oscillation, the main clock cannot be supplied to the system clock. Therefore, make sure to use the MOSCCR.MOSTP bit to start oscillation. The MOFCR.MOFXIN bit should be set only to continue oscillation in software standby mode. For details on the MOSCCR register, see section 9.2.6, Main Clock Oscillator Control Register (MOSCCR). For details on the MOFCR register, see section 9.2.11, Main Clock Oscillator Forced Oscillation Control Register (MOFCR).

After the main clock oscillator starts operating, the main clock supply to the CEC is enabled when the main clock is counted for the cycles set by MOSCWTCR.MSTS[4:0]. Also, the main clock supply to the system is enabled after the main clock is counted for an additional 16,384 cycles. For details on the MOSCWTCR register, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR). This waiting operation for oscillation stabilization is the same for the returning operation when oscillation is stopped in software standby mode.

After the main clock oscillation stabilization wait time has elapsed and the main clock supply to the CEC is enabled, set the MOSCR.MOSE bit to 1 to start the main clock supply. For details on the MOSCR register, see section 9.2.12, Main Clock Supply Control Register (MOSCR).

A noise filter is included in the path for supplying the main clock to the CEC. This noise filter can be enabled or disabled by the MONFCR register. For details on the MONFCR register, see section 9.2.13, Main Clock Noise Filter Control Register (MONFCR). Rewrite the MONFCR register while the MOSCR.MOSE bit is set to 0 (main clock supply to CEC is stopped).

To continue operation of the main clock oscillator in software standby mode, set the MOFCR.MOFXIN bit to 1 when the main clock is oscillating while the MOSCCR.MOSTP bit is set to 0 with the MOSCR.MOSE bit set to 0 (the main clock supply to the CEC is stopped).

Figure 32.10 shows an example of the flowchart for starting the CECMCLK supply to the CEC, and Figure 32.11 shows the path for supplying the main clock to the CEC.

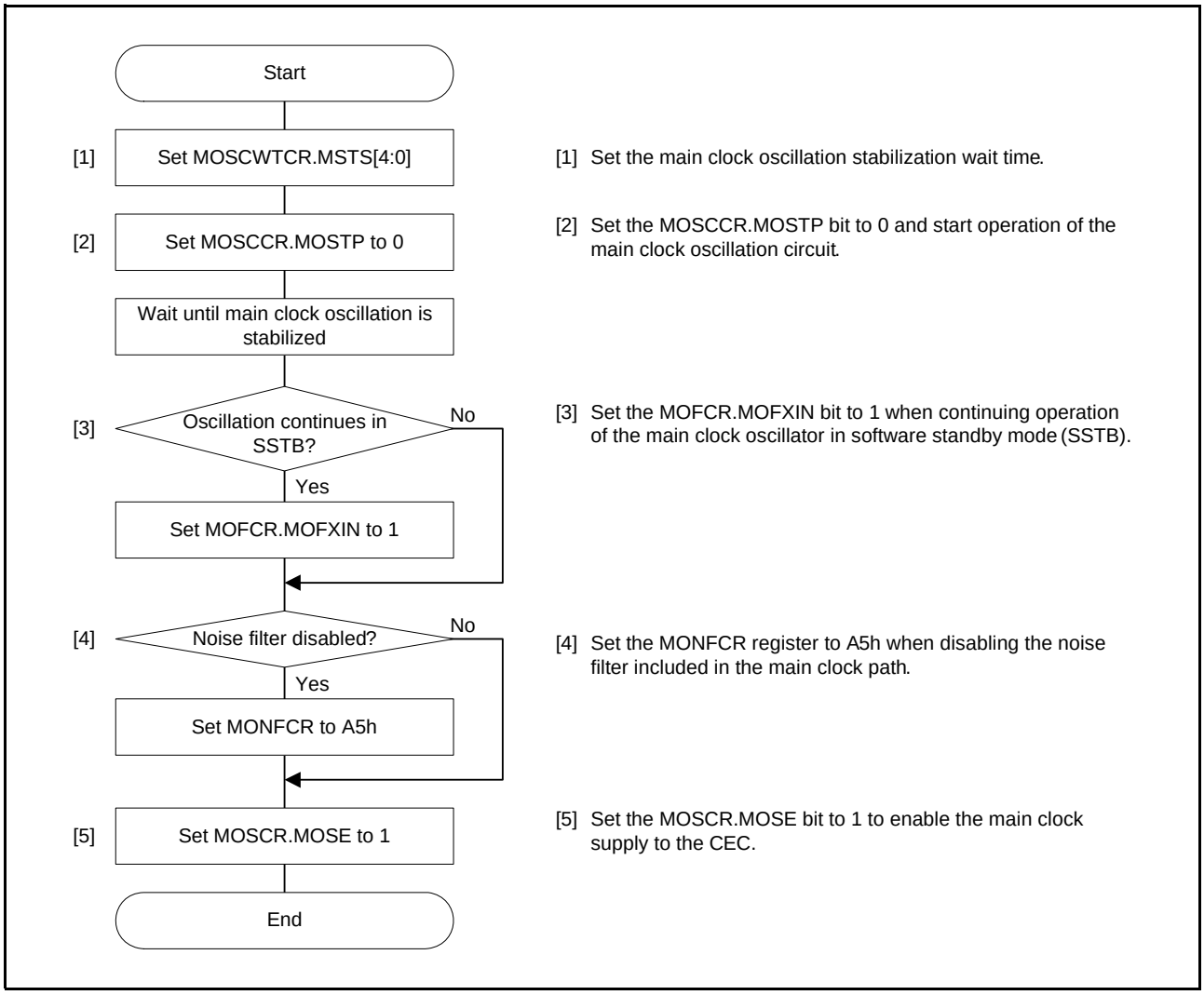


Figure 32.10 Example of Flowchart for Starting CECMCLK Supply to CEC

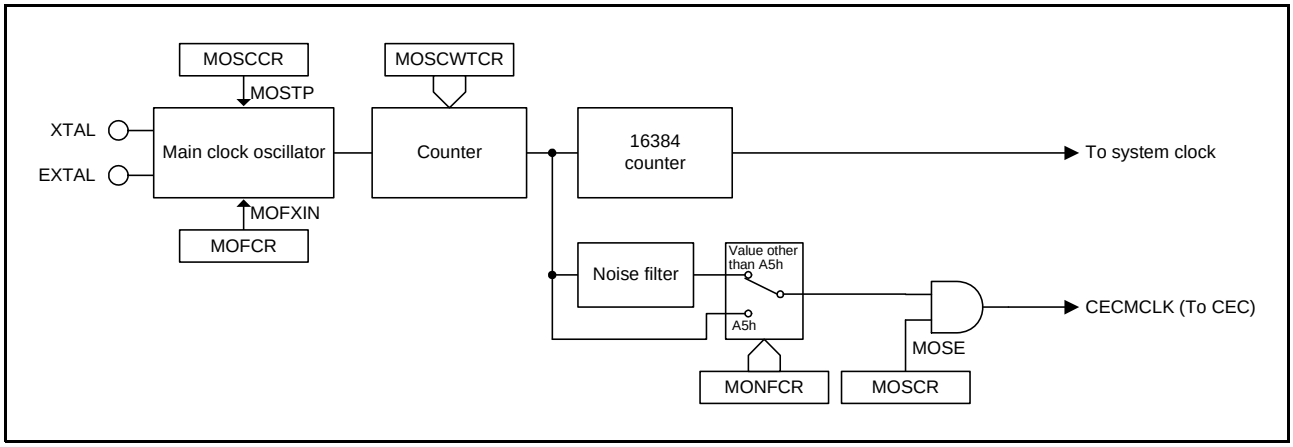


Figure 32.11 Path for Main Clock Supply to CEC

32.3.3 CEC Communication Functions

32.3.3.1 Communication Bit Width Adjustment Function

This function can be used to set the low-level width and bit width of the start bit and data bit during a transmission. The values of some registers must be set according to the specified relationships. Set the following registers to ensure the relationships are kept as shown below.

- STATL < STATB
- LGC1L < LGC0L < DATB

The relationships between the various width setting registers (see section 32.2.9 to section 32.2.13) and the bit timing are shown in (1) to (3).

(1) Start bit

The STATL register is used to set the low-level width and the STATB register is used to set the bit width of the start bit. Figure 32.12 shows the output waveform of the start bit.

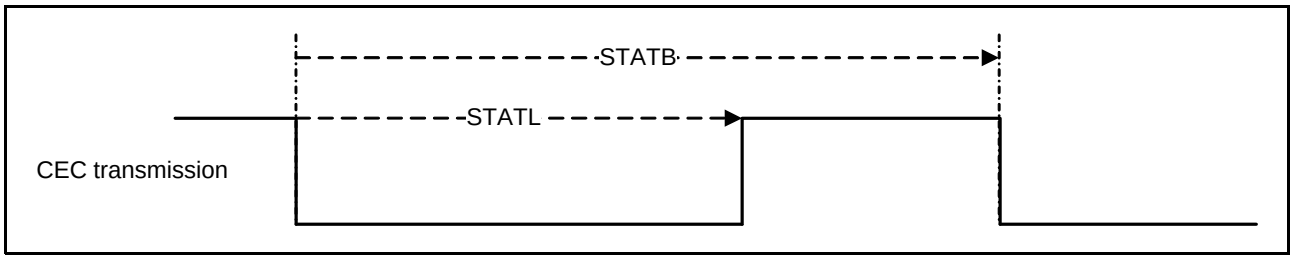


Figure 32.12 Start Bit Output Waveform

(2) Data bit (logical 0)

The LGC0L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 0. Figure 32.13 shows the output waveform of the data bit (logical 0).

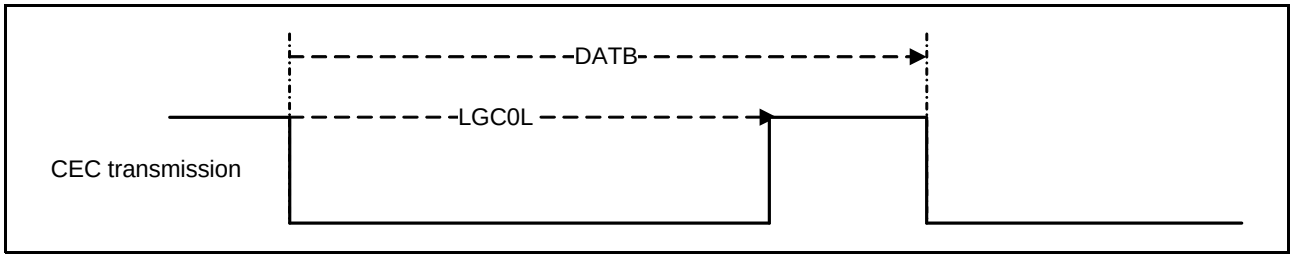


Figure 32.13 Data Bit (Logical 0) Output Waveform

(3) Data bit (logical 1)

The LGC1L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 1. Figure 32.14 shows the output waveform of the data bit (logical 1).

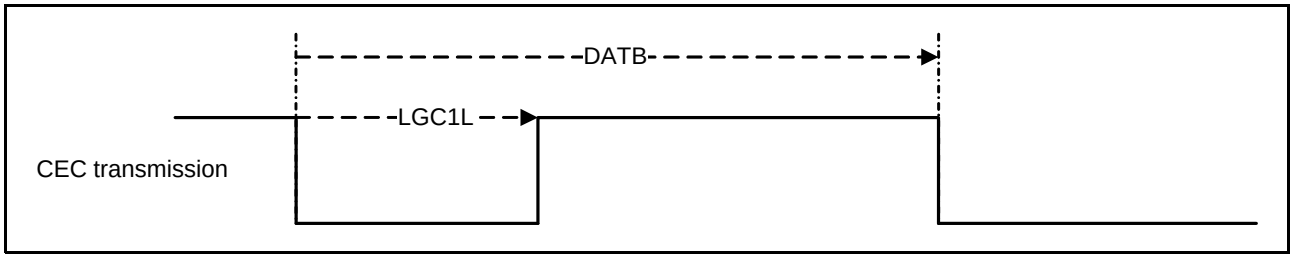


Figure 32.14 Data Bit (Logical 1) Output Waveform

32.3.3.2 Receive Bit Timing Check Function

The CEC transmission/reception circuit has a timing check function that judges whether the low-level width and bit width of the start bit and data bit during a reception are within the set range. The timing check time can be set using the various timing judgment registers (see section 32.2.14 to section 32.2.23).
The values of some registers must be set according to the specified relationships. Set the following registers to ensure the relationships are kept as shown below.

- STATLL < STATLH
- STATBL < STATBH
- LGC0LL < LGC0LH
- LGC1LL < LGC1LH
- DATBL < DATBH

The relationships between the timing judgment registers and bit timing are shown in (1) to (3).

(1) Start bit

The STATLL register is used to set the minimum low-level width and the STATLH register is used to set the maximum low-level width of the start bit. The STATBL register is used to set the minimum value and the STATBH register is used to set the maximum value of the start bit width. Figure 32.15 shows the timing at which the start bit is received.

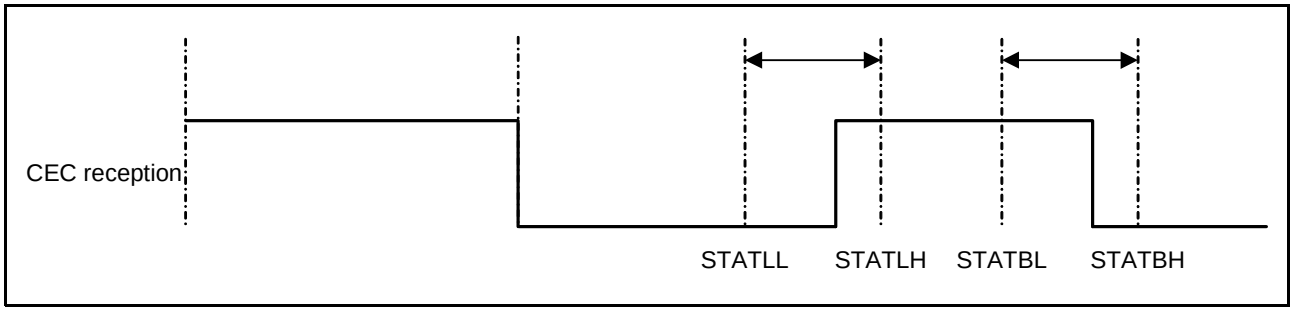


Figure 32.15 Start Bit Reception Timing

(2) Data bit (logical 0)

The LGC0LL register is used to set the minimum low-level width and the LGC0LH register is used to set the maximum low-level width of the data bit (logical 0). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width. Figure 32.16 shows the timing at which the data bit (logical 0) is received.

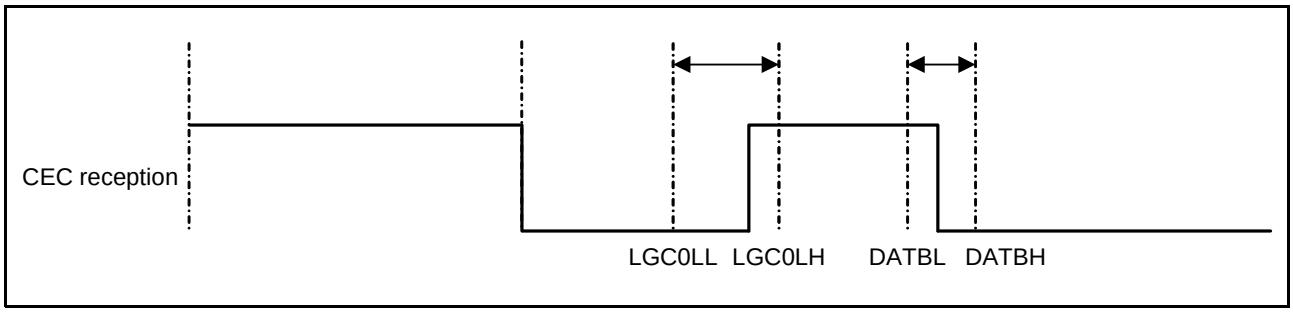


Figure 32.16 Data Bit (Logical 0) Reception Timing

(3) Data bit (logical 1)

The LGC1LL register is used to set the minimum low-level width and the LGC1LH register is used to set the maximum low-level width of the data bit (logical 1). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width. Figure 32.17 shows the timing at which the data bit (logical 1) is received.

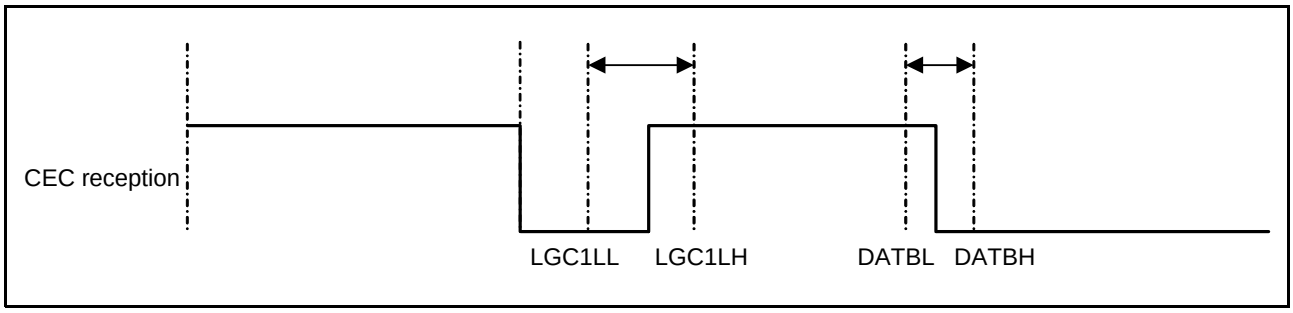


Figure 32.17 Data Bit (Logical 1) Reception Timing

32.3.3.3 Initial CEC Communication Settings

The initial CEC communication setting flow is explained below. The logical address acquisition flow is executed by setting the various control registers and using direct address transmission after a reset. In a logical address acquisition transmission, EOM = 1 is set because the same address is set for the source and destination addresses and only the header block is transmitted. Furthermore, to prevent a false address match from occurring before the local address is determined, CECRXEN = 0 must be set until the CADR setting. Figure 32.18 shows the logical allocation timing diagram and Table 32.7 lists the operation procedure and an explanation of the operation.

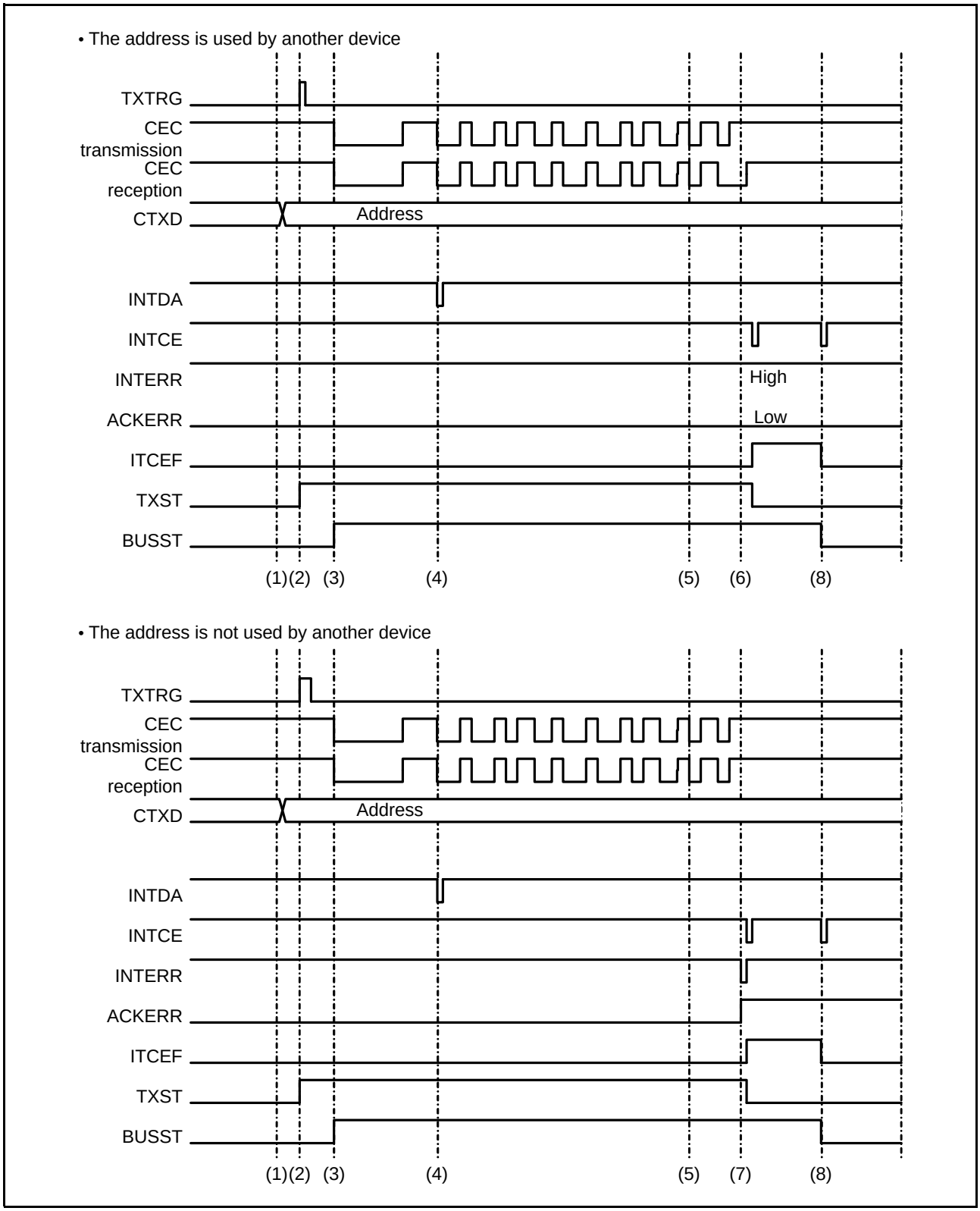


Figure 32.18 Logical Allocation (CECCTL1.CESEL[1:0] = 00b)

Table 32.7 Initial CEC Communication Setting Procedure (1/2)

	Software Manipulation	CEC State
Initial CEC setting	[CEC clock (CECCLK) setting] Set the CCL[2:0] bits. [Reception rejection control setting] Set CECRXEN to 0. [Setting for reporting address mismatch] Set CINTMK. [Noise elimination selection] Set CDFC. (Specify whether to use the noise filter.) [Start bit low-level/bit width setting] Set the low/bit width to STATL/STATB. [Logical 0/1 low-level/bit width setting] Set the low/bit width to LGC0L/LGC1L/DATB. [Sampling time setting] Specify the time to sample the received data for NOMT. [Bit width setting] Specify the bit width for NOMP. [Register settings for timing check] Set the low-level width timing check time of the start bit to STATLL/STATLH. Set the bit width timing check time of the start bit to STATBL/STATBH. Set the low-level width timing check time of the data bit (logical 0) to LGC0LL/LGC0LH. Set the low-level width timing check time of the data bit (logical 1) to LGC1LL/LGC1LH. Set the bit width timing check time of the data bit to DATBL/DATBH. [Bus lock detection setting] Set up BLERRD. (Select whether to detect bus locking.) [Start bit timing error detection setting] Set STERRD. (Select whether to detect timing errors of the start bit.) [Communication complete interrupt setting] Set the CESEL[1:0]. [Signal-free time setting] Set the SFT[1:0] bits. (Set the signal-free time detection time.) [CEC clock (CECCLK) supply] Set CECE to 1. →	The CEC clock (CECCLK) is stopped. The CEC clock (CECCLK) is supplied. Transmission can be performed. The signal-free time is started. BUSST becomes 1. BUSST becomes 0 and the communication standby state is entered after counting up to the set values of the SFT[1:0] bits.

Table 32.7 Initial CEC Communication Setting Procedure (2/2)

	Software Manipulation	CEC State
Logical address allocation	[EOM setting] Set EOM to 1. [Transmit data setting] (1) Set the transmit data (logical address) to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1. →	→ Transmission is started. The start bit is output (3).
	Do not write the next data, because only the header block is transmitted. ←	← The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
	[Local address setting]	1 is output from the EOM bit (5).
	• ACK Change the transmitted address and then retransmit the address, because it is used by another station. ←	The ACK bit is received. ← When logical 0 is received, INTERR is not output and the ACKERR flag is not set (6).
	• NACK Use the transmitted address as the local address, because the transmitted address is not used by another station (CADR setting). ←	← When logical 1 is received, INTERR is output and the ACKERR flag is set (7).
		INTCE is output according to the CESEL[1:0] and SFT[1:0] bit settings (8).
	[Reception rejection control setting] Set CECRXEN to 1. →	→ The communication standby state is entered.

32.3.3.4 CEC Transmission

A CEC transmission performs a receive operation even during transmission and performs an arbitration check, a data check, and a timing check.

The value of the reception buffer register (CRXD) during a transmit operation, however, is not guaranteed.

A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error. Communication is not restarted. Figure 32.19 shows the basic timing of transmission, and Table 32.8 shows the procedure for manipulating CEC transmission.

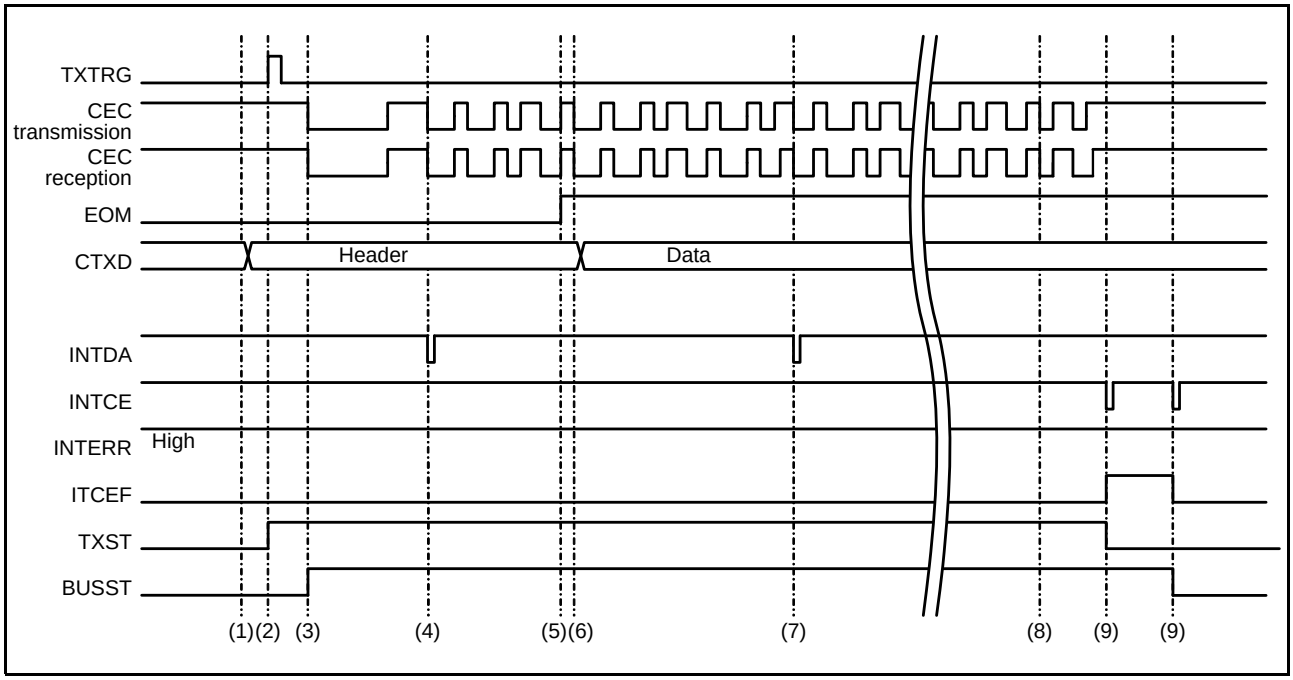


Figure 32.19 Basic Transmission Timing (Direct Address Transmission) (CECCTL1.CESEL[1:0] = 00b)

(1) CEC transmission manipulation procedure

Table 32.8 CEC transmission manipulation procedure

	Software Manipulation	CEC State
Initial CEC setting	See Table 32.7.	
CEC transmit operation	[Signal-free time setting] Set the SFT[1:0] bits. (Set the signal-free time detection time.) [EOM setting] (1) Set EOM (EOM = 0). [Transmit data setting] (1) Set the transmit data to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1. →	Transmission is started. The start bit is output (3).
	[EOM setting] (5) Set the EOM of the next frame (EOM = 1) before the next frame starts (7).	← The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
	[Transmit data setting] (6) Set the transmit data to CTXD.	Outputting the data of the second frame is started (7). 1 is output at the EOM bit position because the last frame is reached (8). INTCE is output according to the CESEL[1:0] and SFT[1:0] bit settings (9). The communication standby state is entered.

(2) Broadcast transmission

When Fh is set to the destination address of the header block transmit data (CTXD), the CEC recognizes the current transmission as a broadcast transmission and operates. Normally, the communication is judged as being successfully performed when logical 0 is received at the ACK bit timing; however, in broadcast communication, the communication is judged as being successfully performed when logical 1 is received at the ACK bit timing.

The CEC judges whether the communication is a direct communication or broadcast communication by looking at the transmit data of the header block, and it automatically determines whether the reception of logical 0 or logical 1 has been successfully performed.

(3) CEC transmission interrupt

The CEC has three interrupt functions, namely a data interrupt (INTDA), a communication complete interrupt (INTCE), and an error interrupt (INTERR). Figure 32.20 shows the timing for generating interrupts during transmission.

A data interrupt (INTDA) occurs at the start of each block.

A communication complete interrupt (INTCE) can be generated if ACK reception for a data block for which EOM is set to 1 ends, if the signal-free time specified using the CECCTL1.SFT[1:0] bits elapses, or if both conditions occur, depending on the settings of the CECCTL1.CESEL[1:0] bits.

An error interrupt (INTERR) is generated if a timing error, ACK error, underrun error, or transmission error is detected during any period of communication.

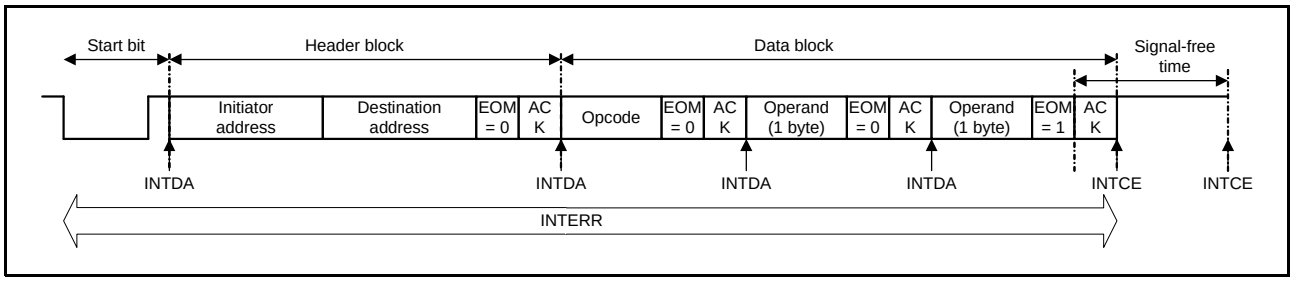


Figure 32.20 Interrupt Generation Timing

If the falling edge of the CEC line is detected when receiving the ACK bit by setting EOM to 1 (before receiving the ACK bit ends), an irregular operation is performed as shown in Table 32.9 according to that timing.

Table 32.9 Operation If Falling Edge of CEC Line Is Detected before Receiving ACK Bit Ends

CEC Line Falling Timing	CECCTL1.CESEL[1:0] Bit Setting	INTCE Generation	Handling of ACK Bit	Operation After CEC Line Falls
After the minimum data bit value (DATBL ≤ counter)	CESEL[1:0] = 00b or CESEL[1:0] = 01b	INTCE is generated once when the CEC line falls.	Handling the ACK bit is enabled because it has the correct width. (ACK or NACK is correctly determined.)	The start of the next communication is recognized and then determining whether to receive the start bit starts.
	CESEL[1:0] = 10b	INTCE is not generated.		
Before the minimum data bit value (counter < DATBL)	CESEL[1:0] = 00b or CESEL[1:0] = 01b	INTCE is generated once when the CEC line falls.	ACK cannot be correctly determined because it has the incorrect width. (If ACKTEN is set to 1, a timing error occurs.)	
	CESEL[1:0] = 10b	INTCE is not generated.		

(4) Receiving error handling pulse

During initiator operation, if the received data is at low level when the maximum low-level width of logical 0 is reached, an error handling pulse is judged to be received as shown in Figure 32.21, a timing error occurs, transmission stops, and then the communication standby state is entered.

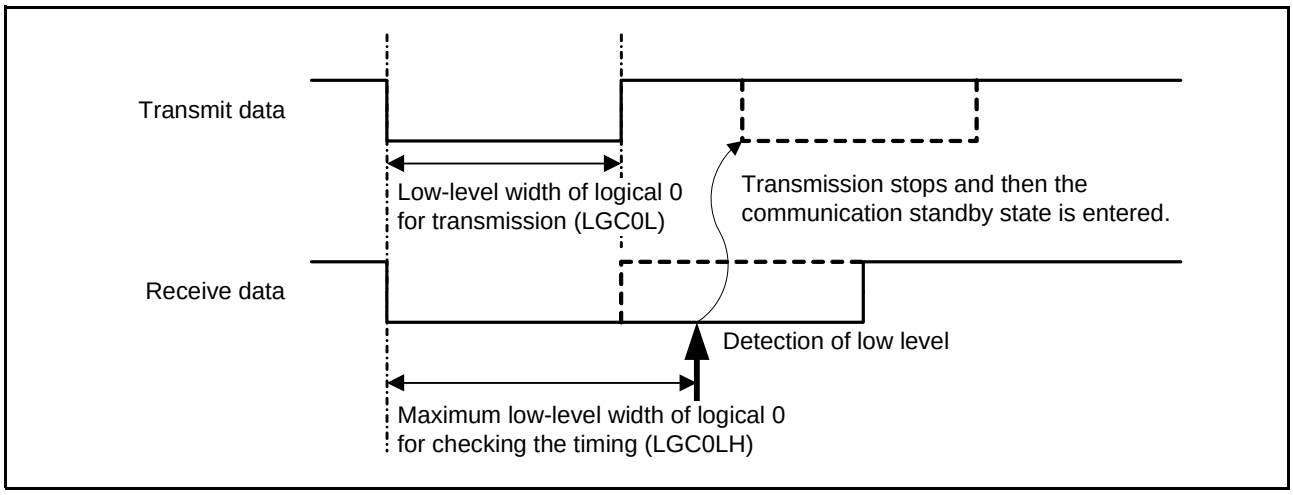


Figure 32.21 Operation of Receiving Error Handling Pulse

32.3.3.5 CEC Reception

During reception, the data is received at the sampling timing set by the CEC reception data sampling time setting register (NOMT) and stored in the reception buffer register (CRXD).
The receive operation differs depending on the CECCTL0.CECRXEN bit setting value, CECCTL1.CINTMK bit setting value, communication type (direct address communication or broadcast communication), and whether the reception address and local address match.
The correspondences between various conditions and the operations are shown in Table 32.10.

A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error. However, when the restart reception function is used (CECEXMOD.RERCVEN = 1), a timing error is detected and judged as the start of new reception, so reception operation is performed again.

Table 32.10 Operation during CEC Reception

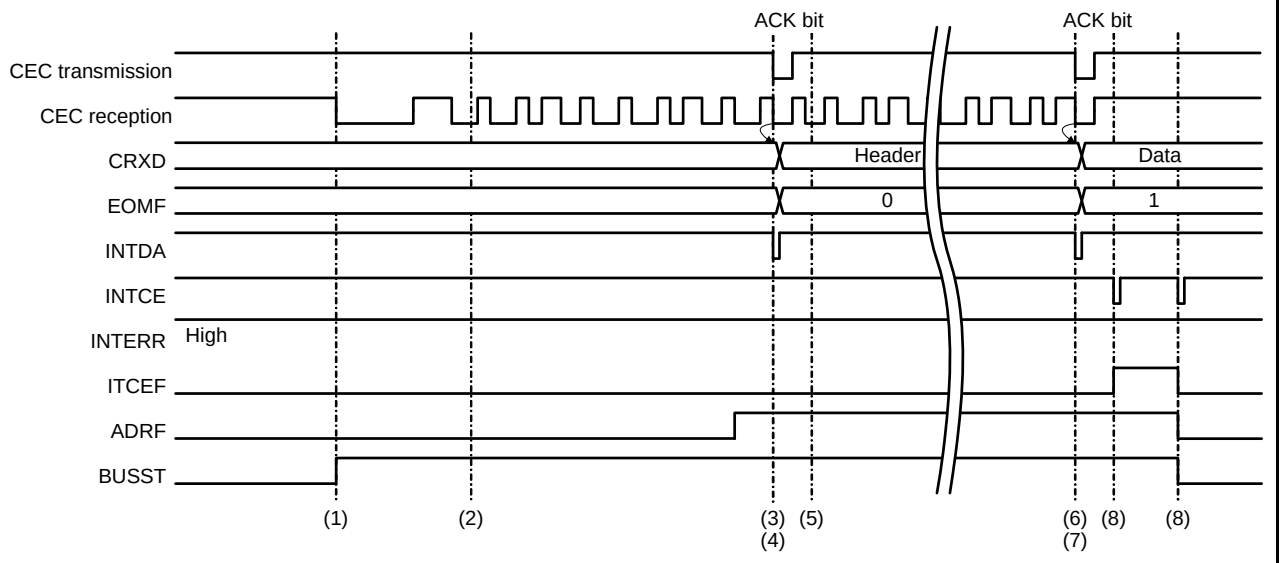
CEC RXEN	**0**	**1**							
Communication type	—	Start bit	Header			Direct (data)			Broadcast (data)
Address match/ mismatch	—	—	Mismatch		Match	Mismatch		Match	—
CINTMK bit	—	—	0	1	—	0	1	—	—
BUSST operation	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
INTDA output	Not supported	Not supported	Not supported	Supported	Supported	Not supported	Supported	Supported	Supported
INTCE output	Not supported	Supported*3	Supported*3	Supported	Supported	Not supported	Supported	Supported	Supported
INTERR output	Not supported	Not supported	Supported	Supported	Supported	Not supported	Supported	Supported	Supported
Error flag operation	Not supported	Not supported	Supported	Supported	Supported	Not supported	Supported	Supported	Supported
Error detection (other than short bit width detection)	Not supported	Supported*4	Supported	Supported	Supported	Not supported	Supported	Supported	Supported
Error detection (short bit width detection)	Not supported	Supported*4	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Error handling output	Not supported	Not supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Bus lock detection*1	Supported*2	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ACK/NACK output	Not supported	Not supported	Supported	Supported	Supported	Not supported	Not supported	Supported	Supported
Signal-free time count	Not supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported

—: Don't care

Note 1. Bus lock errors are detected by setting CECCTL1.BLERRD.
Note 2. A bus lock error is detected but a flag is not set.
Note 3. Only if an error is detected.
Note 4. This is supported only if detecting timing errors for the start bit (CECCTL1.STERRD = 1). An error is detected but a flag is not set.

(1) CEC reception manipulation procedure

For operation of receiving a direct address message, Figure 32.22 and Table 32.11 show operation when the addresses match, and Figure 32.23 and Table 32.12 show operation when the addresses do not match.

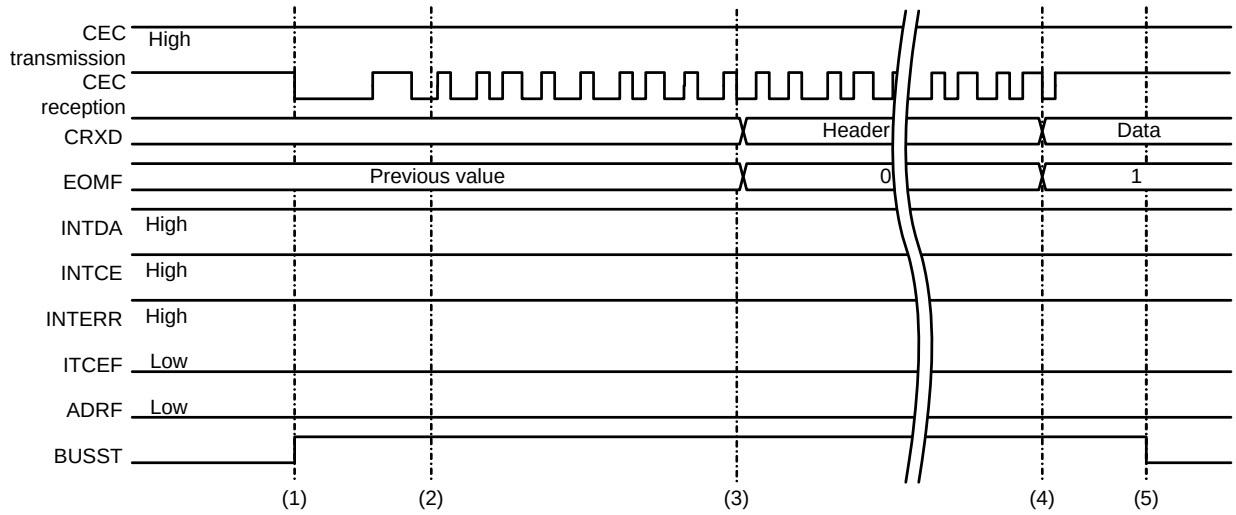


Note: (1) to (8) in Figure 32.22 correspond to (1) to (8) in Table 32.11.

Figure 32.22 Basic Reception Timing (1) (Direct Address Reception, CECCTL1.CESEL[1:0] = 00b)

Table 32.11 CEC Reception Manipulation Procedure

	Software Manipulation	CEC State
CEC initial setting	See Table 32.8.	
CEC receive operation	Prepare for receiving data by returning from low power consumption mode or the like in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame. ←	[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is generated because the address received at the header block has matched with the local address (3). [ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (4). [Continuing reception] The data of the second frame is continuously received (5).
	Read the receive data from CRXD in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame. ←	[Receive data interrupt] When receiving 8-bit data is completed, the data is transferred to CRXD and INTDA is generated (6). [ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (7). [Reception completion] The reception is judged to be completed because EOM = 1 is received and INTCE is output according to the CECCTL1.CESEL[1:0] and CECCTL1.SFT[1:0] bit settings (8).



Note: (1) to (5) in Figure 32.23 correspond to (1) to (5) in Table 32.12.

Figure 32.23 Basic Reception Timing (2) (CECCTL0.CECRXEN = 1, Direct Address, Address Mismatch, CECCTL1.CINTMK = 0)

Table 32.12 CEC Reception Manipulation Procedure (2)

	Software Manipulation	CEC State
CEC receive operation		[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started. Set BUSST flag (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is not generated and neither ACK nor NACK is returned because the address received at the header block does not match the local address and CINTMK is 0 (3). However, monitoring the CEC line continues in order to check the bit length and detect completion of communication. [ACK bit transmission] Neither ACK nor NACK is returned because communication is performed between other stations (4). [Reception completion] Communication between other stations is judged to be completed because EOM = 1 is received, the signal-free time is counted according to the SFT[1:0] bit settings, and then BUSST is cleared to 0 (5).

(2) Broadcast reception

The reception flow and timing check period are the same as those of direct address reception. If the destination address transmitted by the initiator is Fh, the communication operates as a broadcast reception.
The differences from direct address reception are as follows.

- Logical 1 is transmitted at the ACK bit timing in a normal operation.
- If reception has failed or CECRXEN = 0 has been set, logical 0 is transmitted at the ACK bit timing.

(3) CEC reception interrupt

Three interrupt functions, namely a data interrupt (INTDA), communication complete interrupt (INTCE), and error interrupt (INTERR) are provided. Figure 32.24 shows the timing for generating interrupts during CEC reception.

A data interrupt (INTDA) is output at the following timings during reception (follower).

- When the address received at the header block of a direct address communication has matched the local address
- When address reception has been completed at the header block of a direct address communication when CECCTL1.CINTMK = 1 is set
- When address reception of a broadcast communication has been completed at the header block
- When data reception has been completed at the data block and the receive data has been stored in the CRXD register

Communication complete interrupt INTCE is output at the following timings during reception (follower).

- CECCTL1.CESEL[1:0] = 00b
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends, if the signal-free time has been counted, or if the falling edge of the CEC line is detected in the high-level period of the ACK bit of the last frame or while the signal-free time is counted.
- CECCTL1.CESEL[1:0] = 01b
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends or if the falling edge of the CEC line is detected in the high-level period of the ACK bit of the last frame or while the signal-free time is counted.
- CECCTL1.CESEL[1:0] = 10b
  INTCE is output if the signal-free time has been counted.

An error interrupt (INTERR) is output at the following timings during reception (follower).

- When a timing error is detected
- When an overrun error is detected
- When a bus lock error is detected when CECCTL1.BLERRD = 1 is set

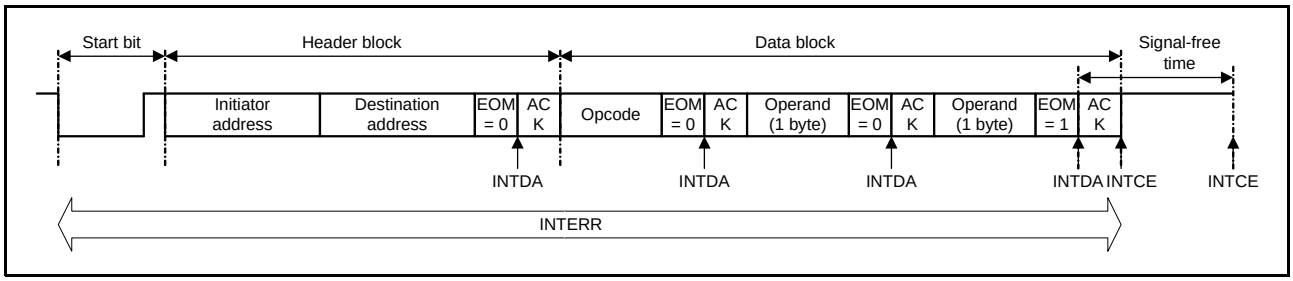


Figure 32.24 Basic Reception Interrupt Timing

32.3.3.6 Status Flag Functions

Table 32.13 lists status flags.

Table 32.13 Status Flags

No.	Status Flag	Register.Bit Symbol
1	Address match detection flag	CECS.ADRF
2	Bus busy detection flag	CECS.BUSST
3	Transmission status flag	CECS.TXST
4	EOM flag	CECS.EOMF
5	INTCE generation source flag	CECS.ITCEF
6	Signal-free time rewrite disable report flag	CECS.SFTST
7	CEC line monitor	CECEXMON.CECLNMON
8	ACK flag	CECEXMON.ACKF

(1) Address match detection flag

As shown in Figure 32.25, during follower operation, if the destination address of the header block received during direct address communication matches the address set by the CEC local address setting register (CADR) or during broadcast communication, the address match flag (CECS.ADRF) is set to 1 at the same time when a data interrupt (INTDA) of the header block is generated.

The address match flag is cleared at the timing of the communication completion interrupt (INTCE) that is generated upon completion of the signal-free time counting after the last frame (EOM = 1) is received.

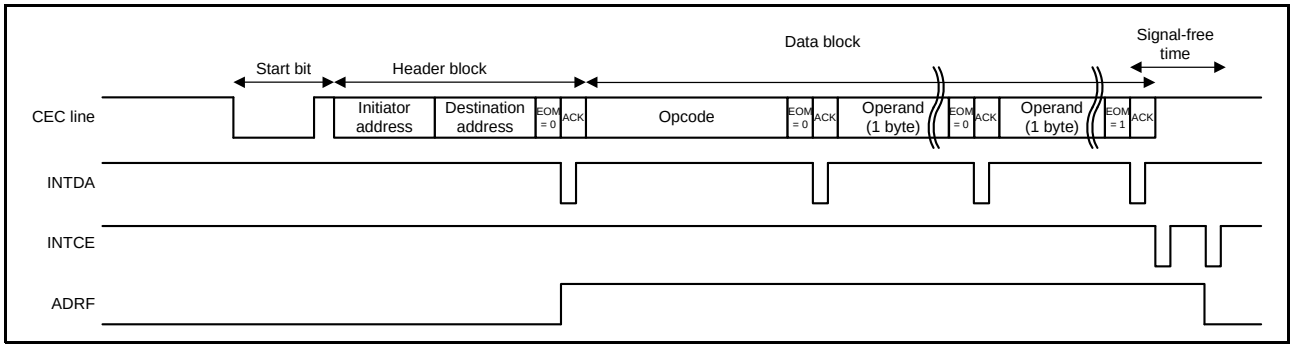
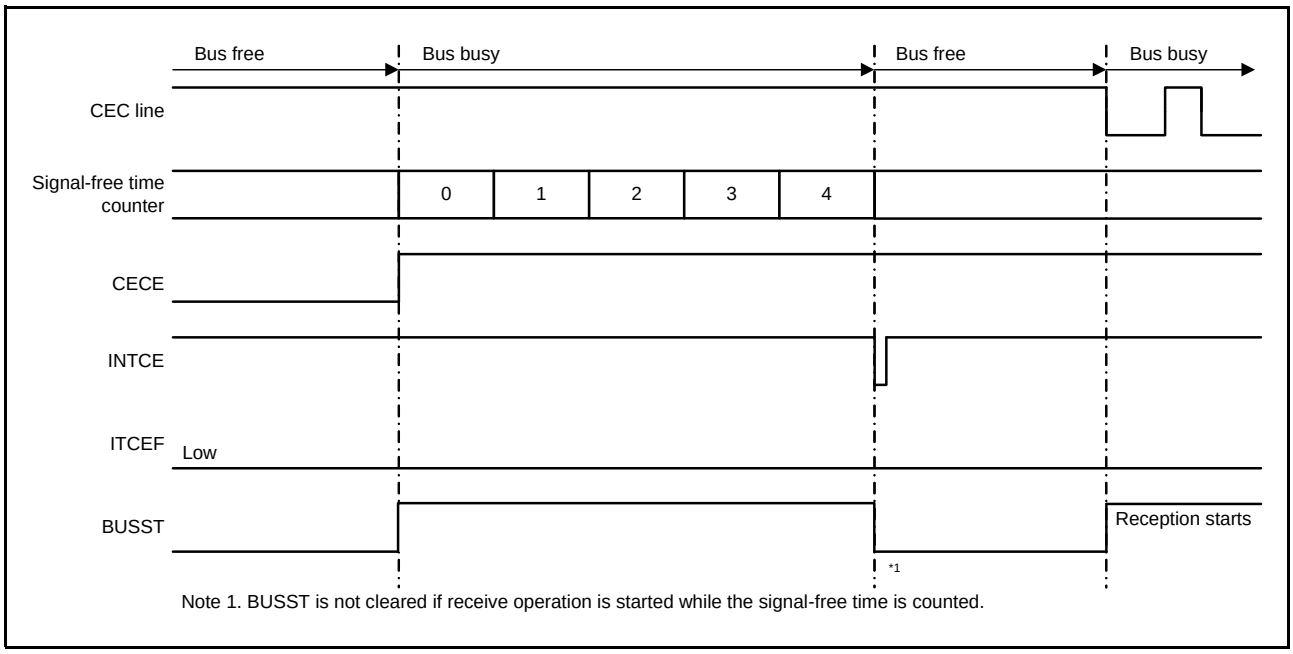


Figure 32.25 ADRF Bit Operation Timing

(2) Bus busy detection flag

Figure 32.26 to Figure 32.28 show the timing of operation of the bus busy flag (CECS.BUSST). When CEC operation is enabled (CECCTL0.CECE = 0→1) or operation of the CEC line is detected, the bus busy flag (CECS.BUSST) is set. After communication is completed and then the signal-free time has elapsed, the bus busy flag is cleared.



Note 1. BUSST is not cleared if receive operation is started while the signal-free time is counted.

Figure 32.26 Timing When CECE is Set to 1 at Start of Reception

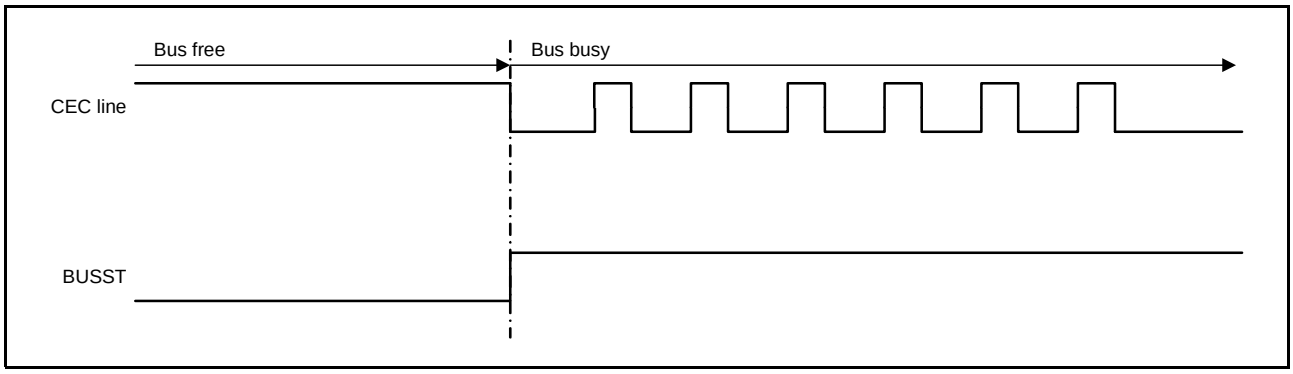


Figure 32.27 CEC Line Fall Detection Timing

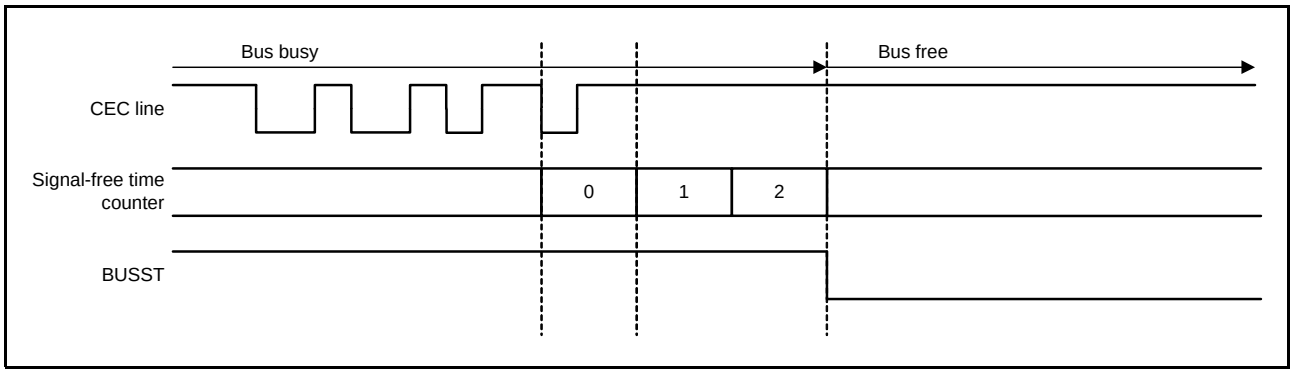


Figure 32.28 Timing When Signal-Free Time Set by CECCTL1.SFT[1:0] Bits Has Elapsed After Completion of Communication

(3) Transmission status flag

As shown in Figure 32.29, when 1 is written to the transmission start trigger bit (CECCTL0.TXTRG) during initiator operation, the transmission status flag (CECS.TXST) is set.
The transmission status flag is cleared when the communication complete interrupt (INTCE) that is generated upon completion of ACK reception for the data block with EOM = 1. However, as shown in Figure 32.30, if arbitration is lost, the transmission status flag is cleared at the same time when an error interrupt (INTERR) is generated and the arbitration loss detection flag (CECES.AERR) is set.
Likewise, if an underrun error occurs, the transmission status flag is cleared as the same time when an error interrupt is generated.

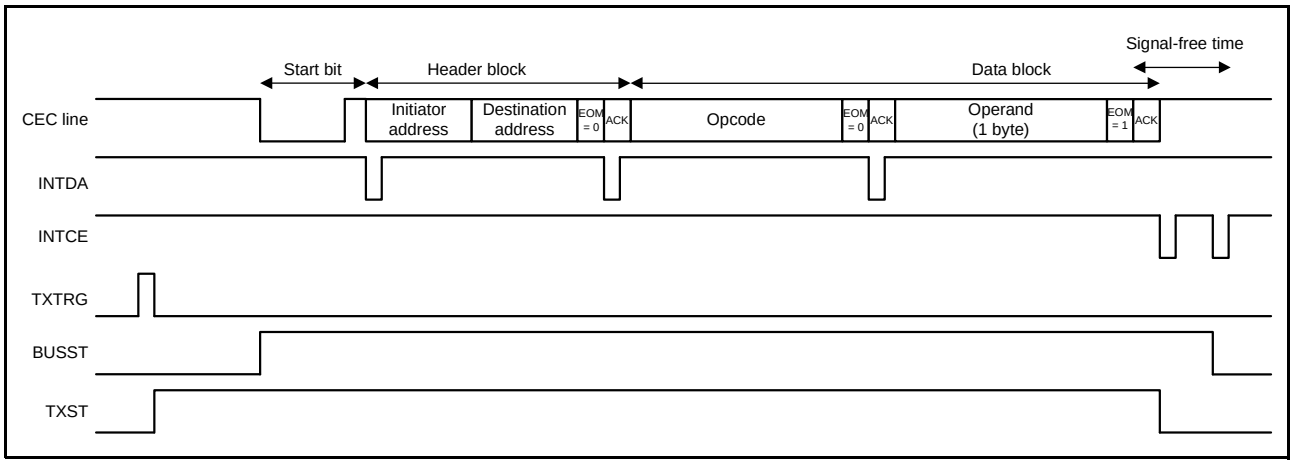


Figure 32.29 Transmission Status Flag Timing during Normal Transmit Operation

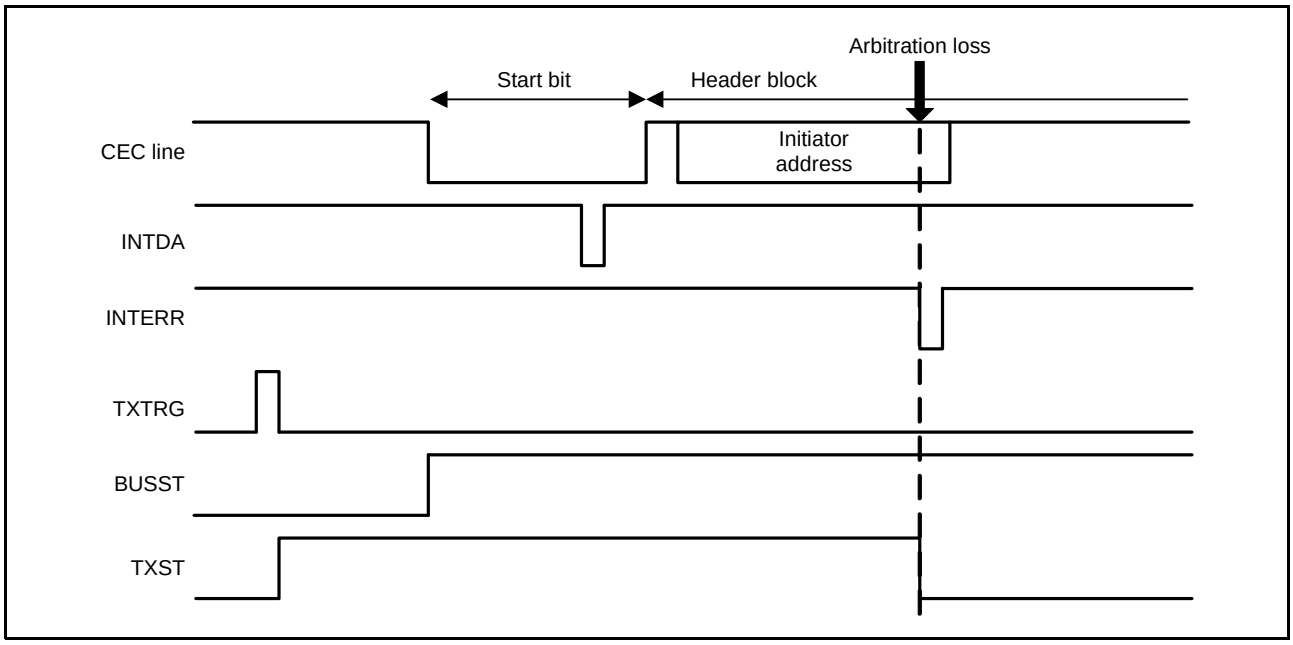


Figure 32.30 Transmission Status Flag Timing When Arbitration is Lost

(4) EOM flag

As shown in Figure 32.31, during follower operation, the EOM flag (CECS.EOMF) is updated at the same time when a data interrupt (INTDA) is generated.

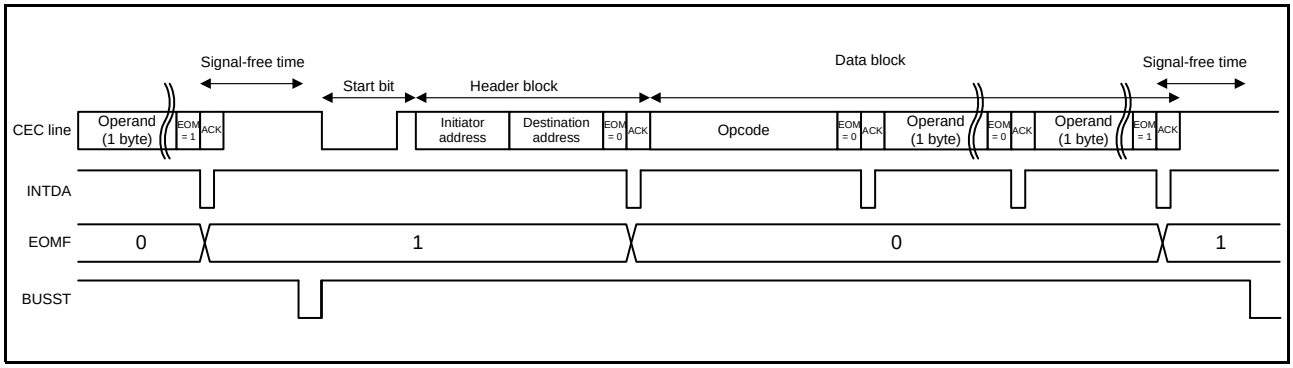


Figure 32.31 EOMF Bit Operation Timing

(5) INTCE generation source flag

This flag indicates which source was the generation source when a communication complete interrupt (INTCE) is generated. The INTCE generation source flag (CECS.ITCEF) is set at the same time as the communication complete interrupt (INTCE) at the timing of ACK reception of the last block or when an error occurs.
After communication is completed, the INTCE generation source flag is cleared when the signal-free time has elapsed. However, if receive operation is started during the signal-free time, the INTCE generation source flag is not cleared and remains 1. Figure 32.32 shows an example of using the INTCE generation source flag (CECS.ITCEF) to check the INTCE generation source.

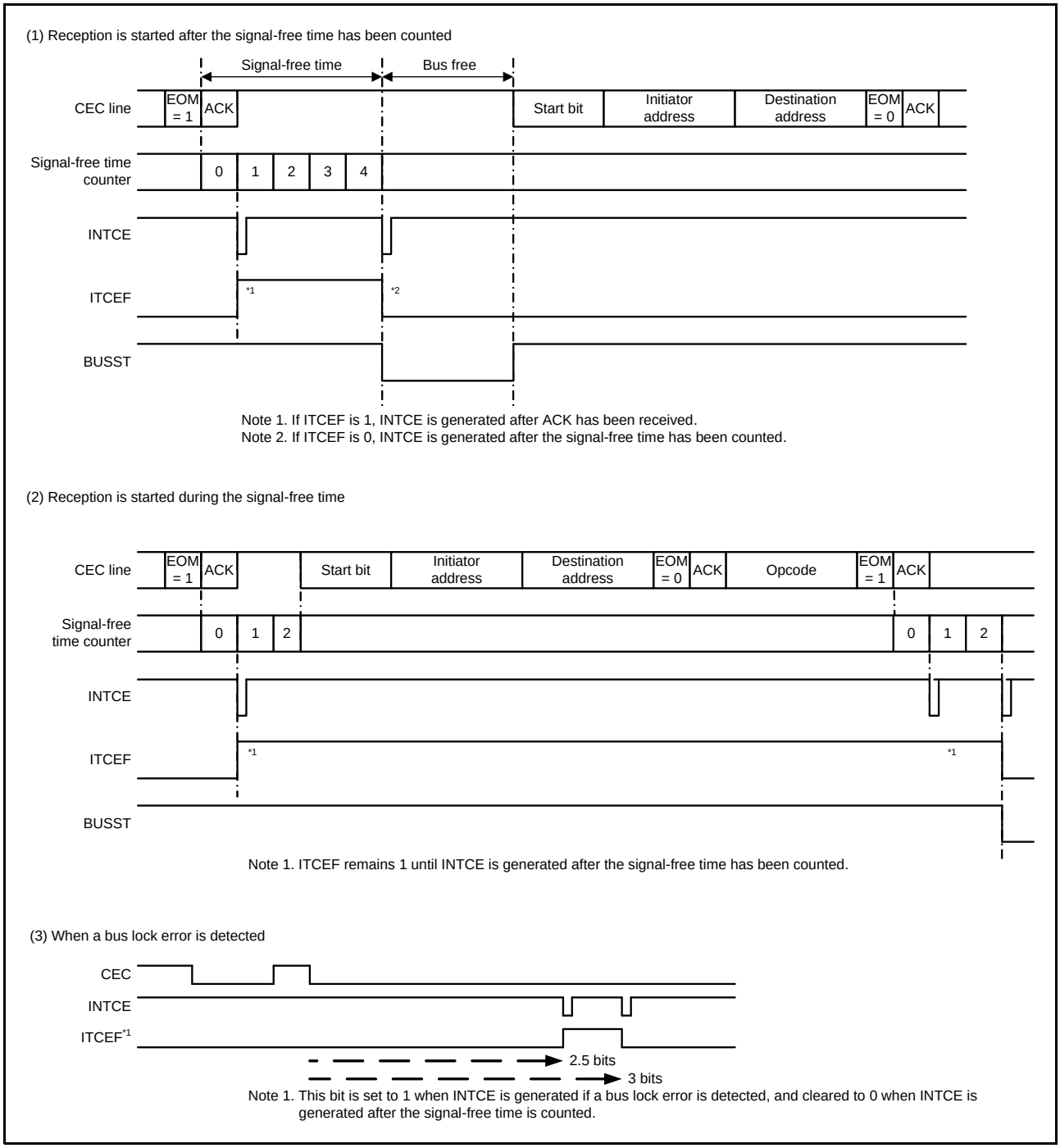


Figure 32.32 Using ITCEF to Check INTCE Generation Source When CECCTL1.CESEL[1:0] = 00b

(6) Signal-free time rewrite disable report flag

This flag indicates the rewrite disable period of the signal-free time data width select bits (CECCTL1.SFT[1:0]). As shown in Figure 32.33, when the CECCTL1.SFT[1:0] bits are accessed, the signal-free time rewrite disable report flag (CECS.SFTST) is set.
This flag is cleared after the CECCTL1.SFT[1:0] bit setting is applied to the CEC internal control circuit.

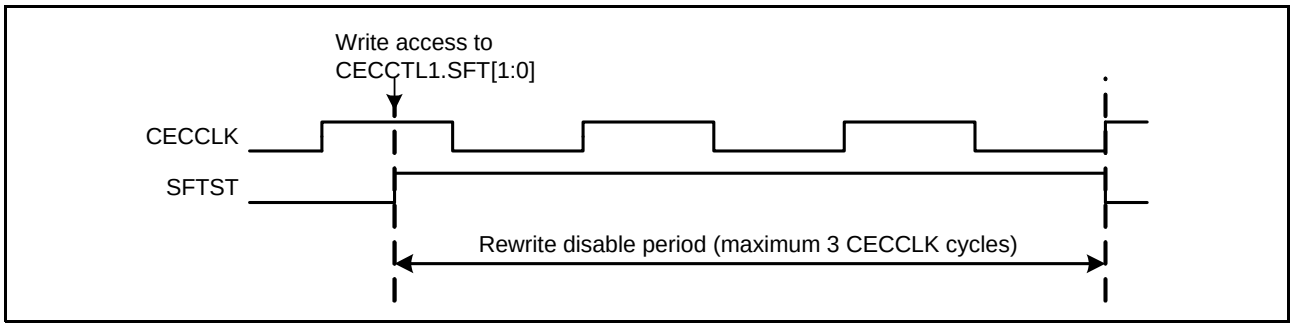


Figure 32.33 SFTST Bit Operation Timing

(7) CEC line monitor

Figure 32.34 shows the timing of operation of the CECEXMON.CECLNMON bit.
The state of the CEC pin can be read by reading the CECEXMON.CECLNMON bit.

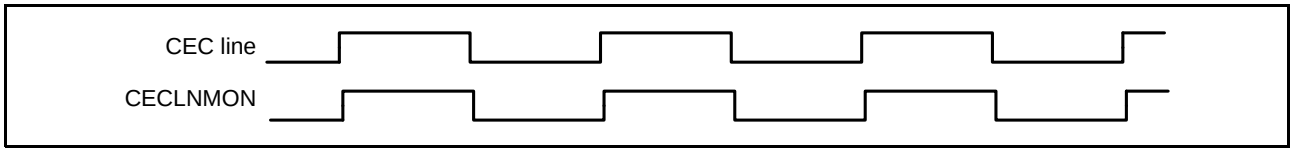


Figure 32.34 CECLNMON Bit Operation Timing

(8) ACK flag

During follower operation, the ACK flag (CECEXMON.ACKF) is updated at the timing of ACK bit reception.

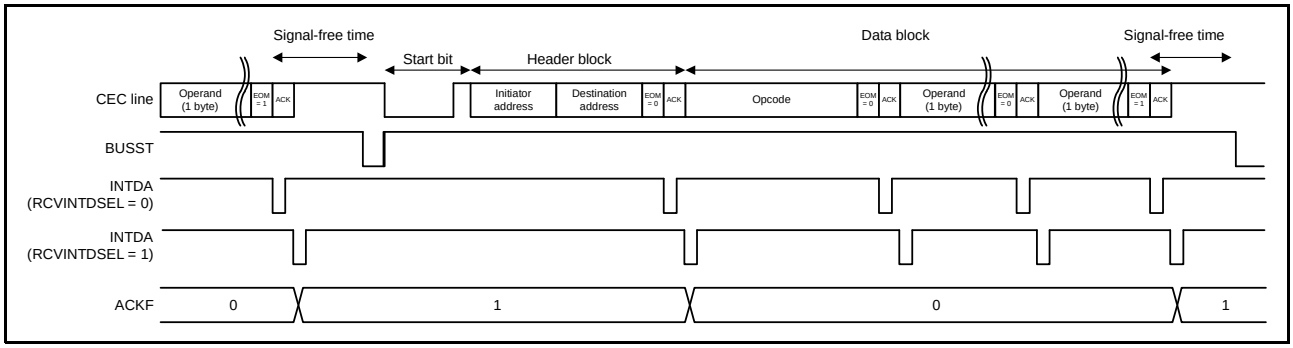


Figure 32.35 ACKF Bit Operation Timing

As shown in Figure 32.35, when reading the ACKF bit while CECEXMD.RCVINTDSEL = 0, the received ACK state can be read by reading this bit when a 1-bit wait time has elapsed after a data interrupt (INTDA) is generated. (If this bit is read after a data interrupt is generated, the ACK of the previously received data is read).
When reading the ACKF bit while CECEXMD.RCVINTDSEL = 1, read this bit after a data interrupt (INTDA) is generated. The ACK of the newest received data can be read.

32.3.3.7 CEC Interrupts

The CEC transmission/reception circuit generates three interrupt requests.

- Data interrupt (INTDA, INTDAA)
  During transmission, this interrupt is generated at the timing when transmission of each block is started.
  During reception, this interrupt is generated at the timing of each completion of EOM bit reception if CECEXMD.RCVINTDSEL is 0, and each completion of ACK bit transmission if CECEXMD.RCVINTDSEL = 1, depending on the value of the reception interrupt timing change bit.
- Communication complete interrupt (INTCE, INTCEA)
  During both transmission and reception, this interrupt is generated upon completion of the message or signal-free time is completed. It is also possible to select only one of the two (completion of the message or signal-free time) by setting the communication complete interrupt generation timing change bits (CECCTL1.CESEL[1:0]).
- Error interrupt (INTERR, INTERRA)
  This interrupt is generated when an error is generated.

Figure 32.36 shows the timing for generating each interrupt.

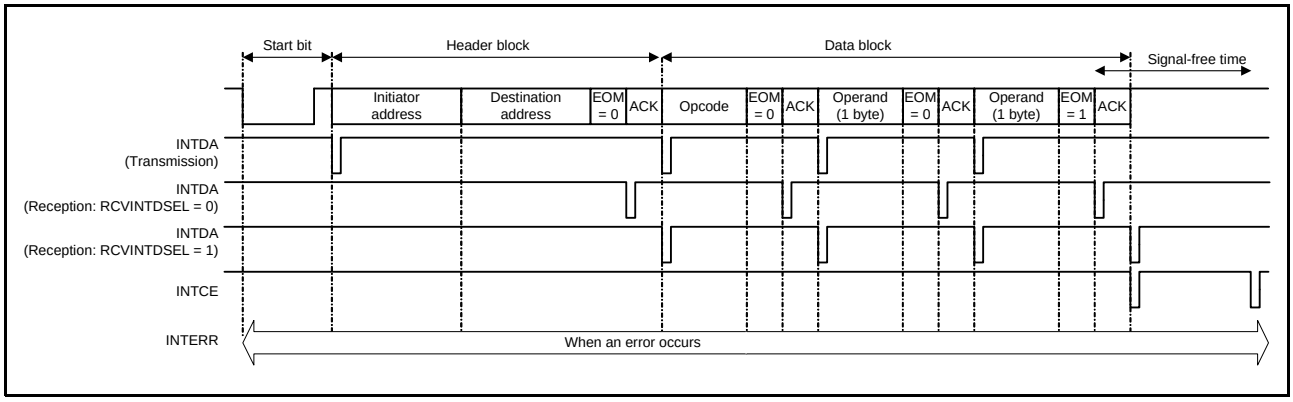


Figure 32.36 Interrupt Generation Timing

Each of three interrupt requests can be used to output two types of interrupt requests. Use the two types exclusively depending on the application of the interrupt request. See Table 32.14 for details on the application corresponding to each CEC interrupt request.

Table 32.14 CEC Interrupt Sources

	Corresponding Source				
Name	**CPU Interrupt**	**DTC Activation**	**DMAC Activation**	**Return from Software Standby Mode**	**Return from All-Module Clock Stop Mode**
INTDAA	Possible	Possible	Not possible	Possible	Possible
INTCEA	Possible	Possible	Not possible	Possible	Possible
INTERRA	Possible	Possible	Not possible	Possible	Possible
INTDA	Possible	Possible	Possible	Not possible	Not possible
INTCE	Possible	Possible	Possible	Not possible	Not possible
INTERR	Possible	Not possible	Not possible	Not possible	Not possible

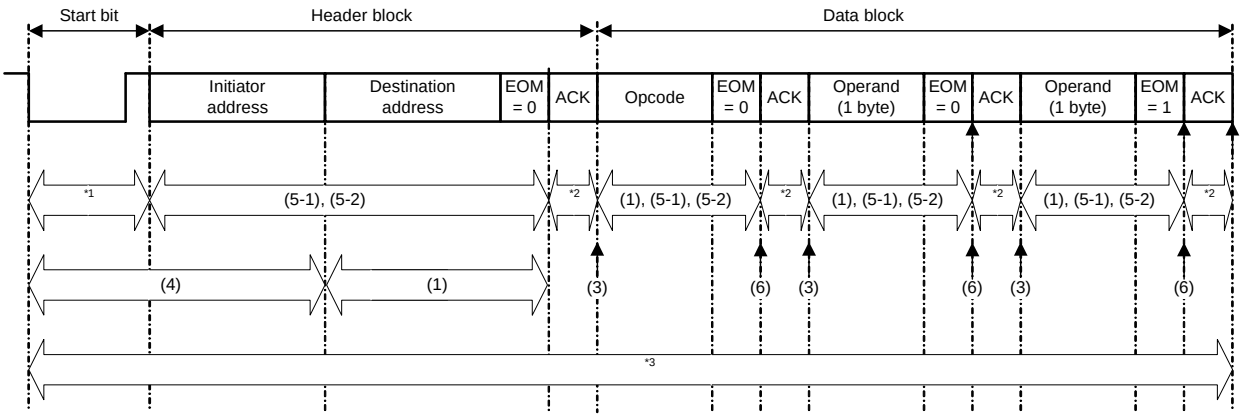
32.3.3.7.1 Error Interrupt Sources

The CEC detects the following seven errors.

Table 32.15 lists the errors that can be detected for the initiator and follower, and Figure 32.37 shows the error detection period.

Table 32.15 Errors That Can Be Detected for Initiator and Follower

Error	Initiator	Follower
(1) Transmission error	Detected	Not detected
(2) ACK error	Detected	Not detected
(3) Underrun error	Detected	Not detected
(4) Arbitration error	Detected	Not detected
(5-1) Timing error (low-level width)	Detected	Detected
(5-2) Timing error (bit width)	Detected	Detected
(6) Overrun error	Not detected	Detected
(7) Bus lock error	Not detected	Detected



Note 1. (5-1) and (5-2) if detecting timing errors for the start bit (CECCTL1.STERRD = 1).
Note 2. (5-2) if detecting timing errors for the ACK bit (CECCTL0.ACKTEN = 1), in addition to (2) and (5-1).
Note 3. (7) if detecting bus lock errors (CECCTL1.BLERRD = 1).

Figure 32.37 Error Detection Period

The details of each error are explained below.

(1) Transmission error

As shown in Figure 32.38, during initiator operation, the logic of the data the initiator has transmitted is compared with that of the CEC line receive data and a transmission error occurs when they differ. Errors are judged at the timing specified by the value set to the CEC reception data sampling time setting register (NOMT). Errors are judged during the data bit period of the frame that includes the EOM bit. An error interrupt (INTERR) is generated after error detection, the transmission error detection flag (CECES.TXERR) is set, and transmission is stopped according to the value of the flag. A communication complete interrupt INTCE is generated at the end of the bit at which transmission stopped and after the signal-free time has been counted, according to the values specified for CECCTL1.CESEL[1:0].

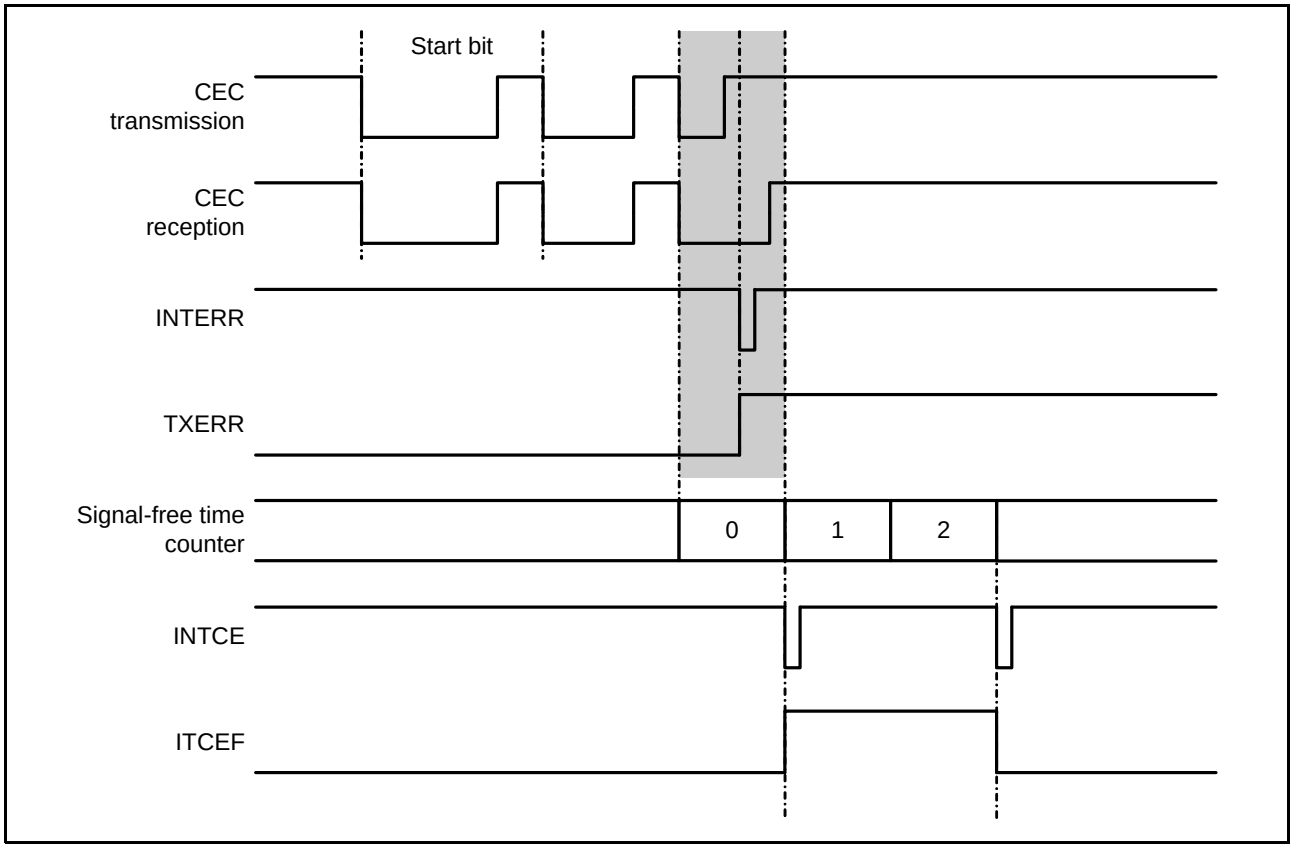


Figure 32.38 Transmission Error Detection Waveform (When 3 Bits are Set as Signal-Free Time)

When a transmission error is detected, the transmit operation is stopped at the bit where the error was detected, regardless of the set value of the CECCTL0.EOM bit.

If EOM = 0 is received even though the initiator is transmitting EOM = 1, the transmission is judged as a transmission error and stopped. The follower judges that the transmission be continued, because EOM = 0. So waiting for data reception. If CECCTL1.BLERRD is set to 1, whether the received data is staying at high or low level can be detected.

(2) ACK error

During direct address transmission, an ACK error occurs when the initiator receives logical 1 at the ACK bit timing. During broadcast transmission, an ACK error occurs when the initiator receives logical 0 at the ACK bit timing. Figure 32.39 shows the timing of ACK error detection.
Errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the ACK error detection flag (CECES.ACKERR) is set. After the end of the ACK bit, the communication standby state is entered and the signal-free time is counted. A communication complete interrupt (INTCE) occurs once or twice depending on the set values of CECCTL1.CESEL[1:0].

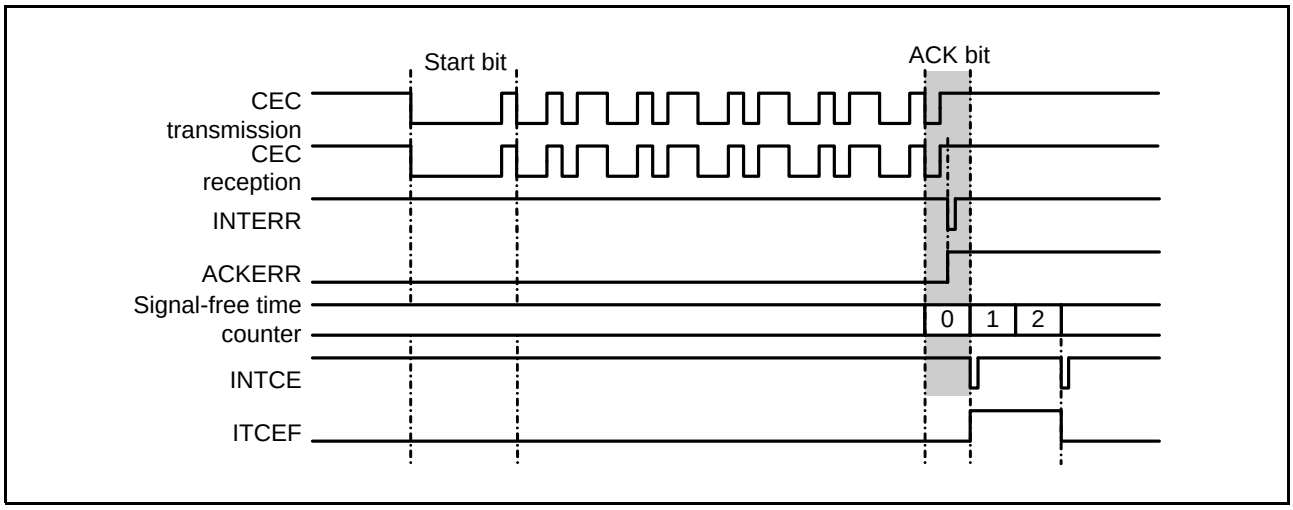


Figure 32.39 ACK Error during Direct Address Communication (When 3 Bits are Set as Signal-Free Time)

(3) Underrun error

An underrun error occurs when no data is set to the transmission buffer when transmitting the next data is started. When an underrun is detected as shown in Figure 32.40, an error interrupt (INTERR) is generated, the underrun error detection flag (CECES.UERR) is set, the transmission is aborted, and the communication standby state is entered. A communication complete interrupt (INTCE) occurs once or twice depending on the set values of CECCTL1.CESEL[1:0].

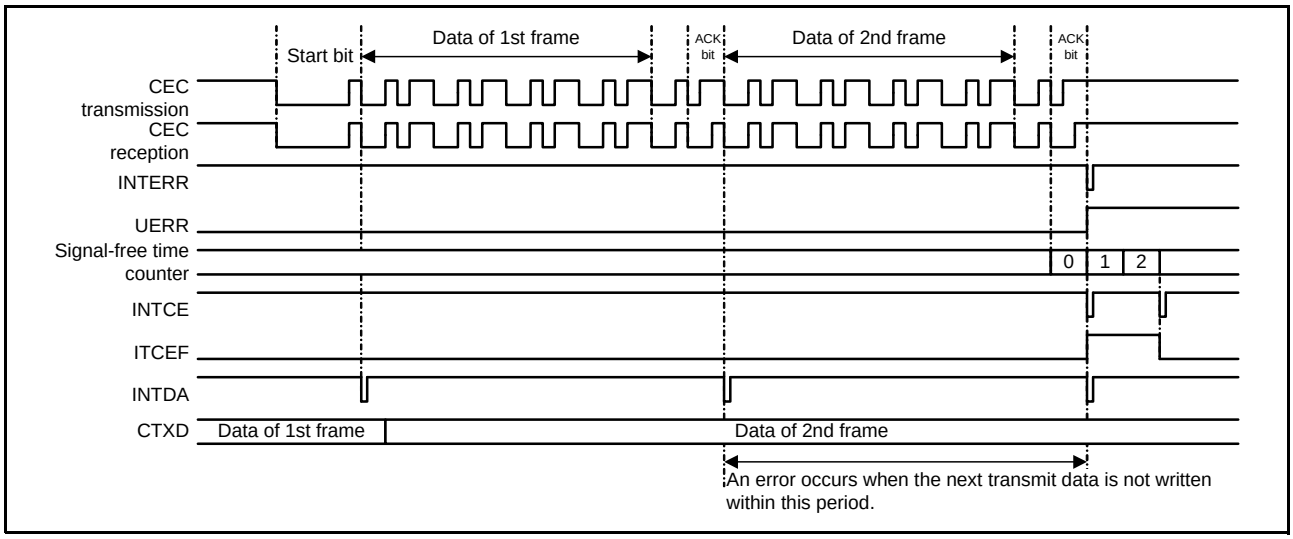


Figure 32.40 Underrun Error Timing

(4) Arbitration error

As shown in Figure 32.41, if logical 0 is received in response to logical 1 transmission during the period from the transmission start trigger (CECCTL0.TXTRG) being set to the source address being transmitted, an arbitration error occurs. Errors between setting the transmission start trigger and outputting the start bit are judged when low level is output to the CEC transmission signal. While the source address is being transmitted, errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the arbitration loss detection flag (CECES.AERR) is set. At this time, the transmission is aborted, but the receive operation is continued. Multiple error flags may be detected, as shown in Figure 32.42, until the source address detection period is entered.

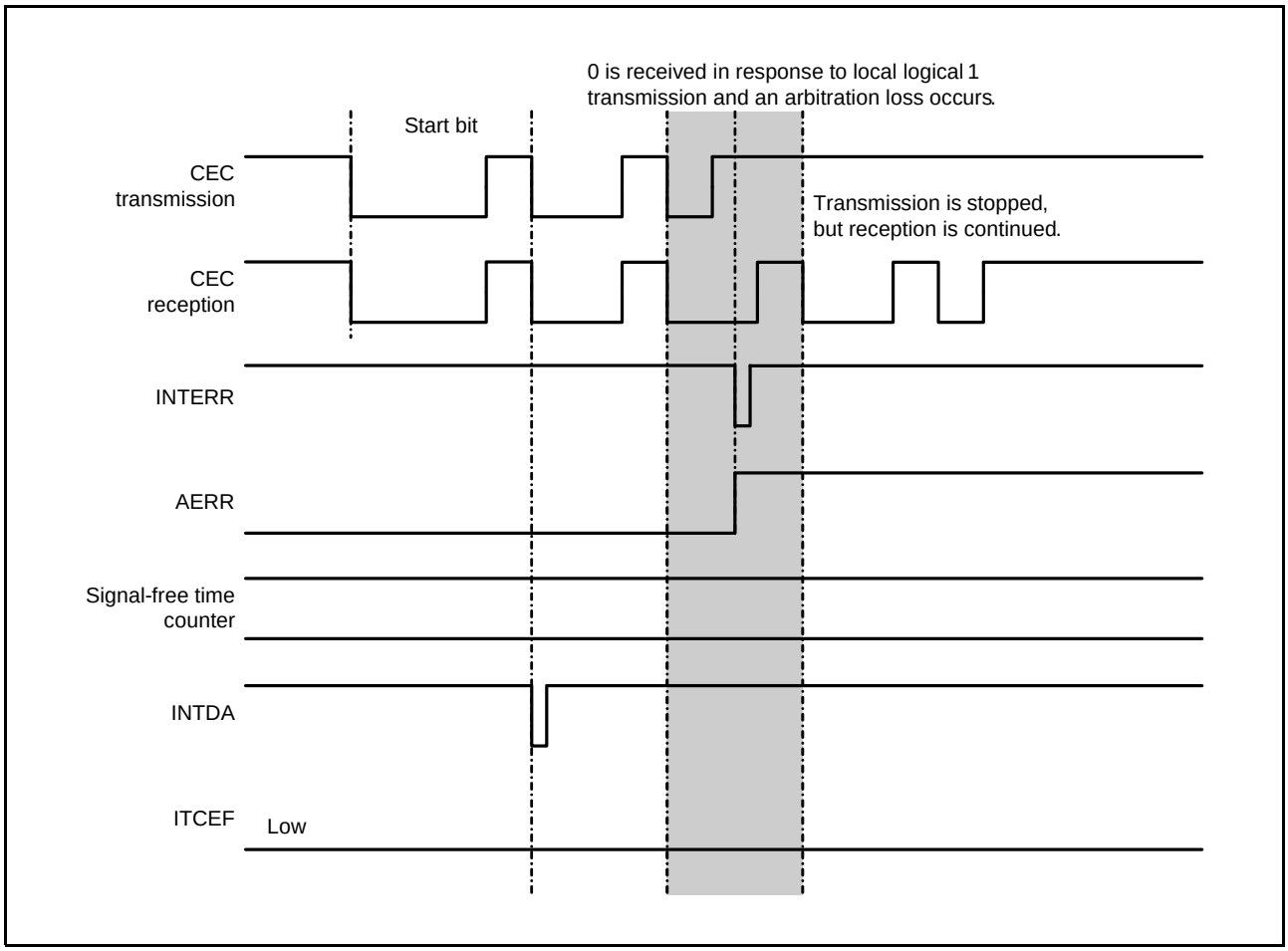
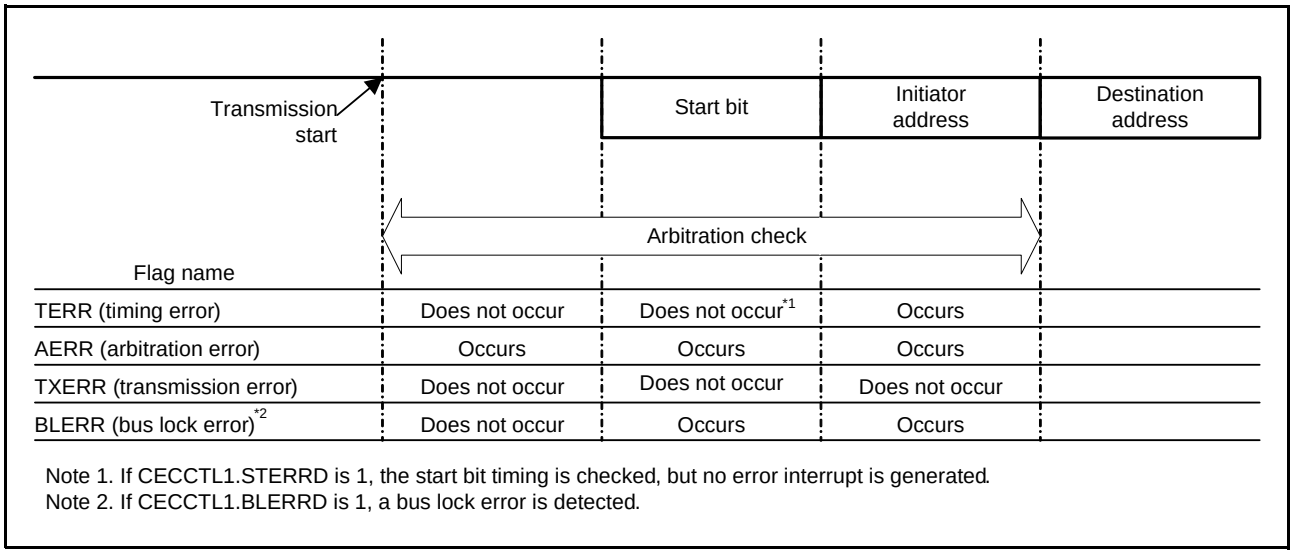


Figure 32.41 Arbitration Timing



Flag name				
TERR (timing error)	Does not occur	Does not occur*1	Occurs	
AERR (arbitration error)	Occurs	Occurs	Occurs	
TXERR (transmission error)	Does not occur	Does not occur	Does not occur	
BLERR (bus lock error)*2	Does not occur	Occurs	Occurs	

Note 1. If CECCTL1.STERRD is 1, the start bit timing is checked, but no error interrupt is generated.
Note 2. If CECCTL1.BLERRD is 1, a bus lock error is detected.

Figure 32.42 Arbitration Error and Other Errors

[Detailed explanation of arbitration error]
Details about the arbitration check performed from the time when the transmission start trigger (CECCTL0.TXTRG) is set until the initiator address output period are provided below.

(a) Arbitration check by setting transmission start trigger (CECCTL0.TXTRG)
The arbitration check is performed when two CECCLK cycles have elapsed after the transmission start trigger (CECCTL0.TXTRG) is set. If an arbitration loss is judged, an error interrupt (INTERR) is generated, the arbitration loss detection flag (CECES.AERR) is set, and the mode is switched to reception mode.

(b) Start bit output period
If low level is detected at the reception line when the transmission start trigger (CECCTL0.TXTRG) is set and the start bit is actually output, the arbitration loss detection flag (CECES.AERR) is set and the mode is switched to reception mode. If the rising edge of the reception line is detected beyond the maximum low-level width of the start bit set by STATLH, the arbitration loss detection flag (CECES.AERR) is set and the mode is switched to reception mode.

(c) Initiator address output period
After transmitting of the start bit is completed, a logic check is performed at the same time as starting to transmit the initiator address. If an address earlier than the local address is detected, an error interrupt (INTERR) is generated, arbitration loss detection flag (CECES.AERR) is set, and the mode is switched to reception mode.

(5) Timing error

Timing errors of the CEC reception signal are checked during whether initiator or follower operation. A timing error occurs if the CEC reception signal is outside the range of the compare register set. As shown in Figure 32.43, low-level width timing errors are detected when the rising edge is detected. Timing errors for the minimum bit width are detected when the falling edge is detected, and timing errors for the maximum bit width are detected if there is no falling edge even though the maximum bit width has been exceeded. Whether to check the ACK bit timing can be selected using the CECCTL0.ACKTEN bit. However, even if CECCTL0.ACKTEN is set to 1, the maximum bit width is not checked only for the ACK bit of the last data block (when EOM is 1). The minimum bit width is checked. As shown in Figure 32.44, if a timing error with a short bit width is detected during follower operation, a low-level pulse (error handling pulse) that has a bit width 1.5 times the bit width specified using the NOMP register is transmitted.

An error handling pulse is not transmitted if a start bit timing error is detected.

If a timing error other than one that has a short bit width is detected during initiator operation, transmission immediately stops. During follower operation, reception is continued as shown in Figure 32.45, and logical 1 is transmitted during direct address communication and logical 0 is transmitted during broadcast communication at the ACK bit timing.

The generation of a communication complete interrupt (INTCE) depends on the set values of CECCTL1.CESEL[1:0]. If a timing error for the minimum bit width of the last ACK bit is detected, a communication complete interrupt (INTCE) is output at the same time as an error interrupt (INTERR).

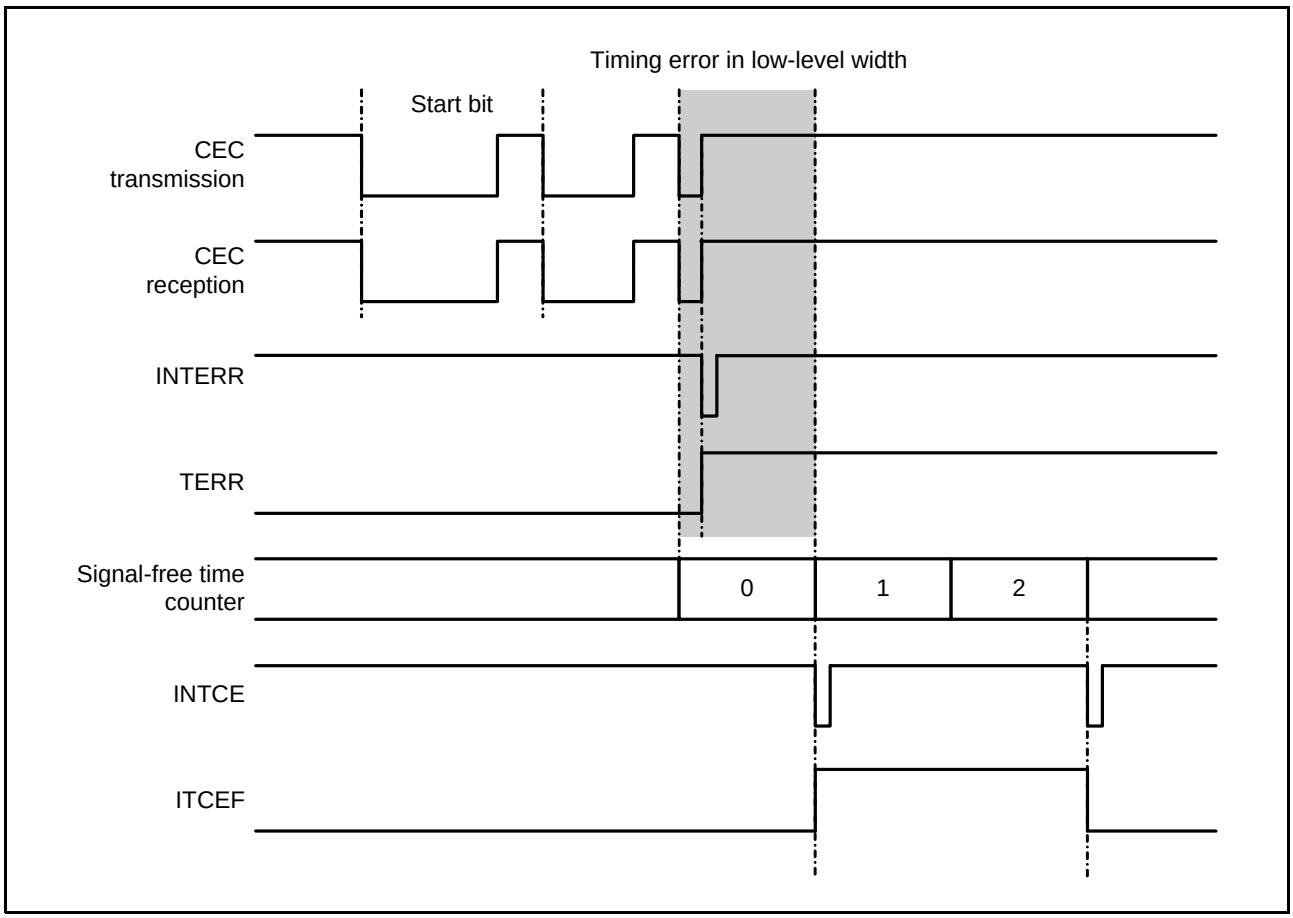


Figure 32.43 Timing Error during Initiator Operation

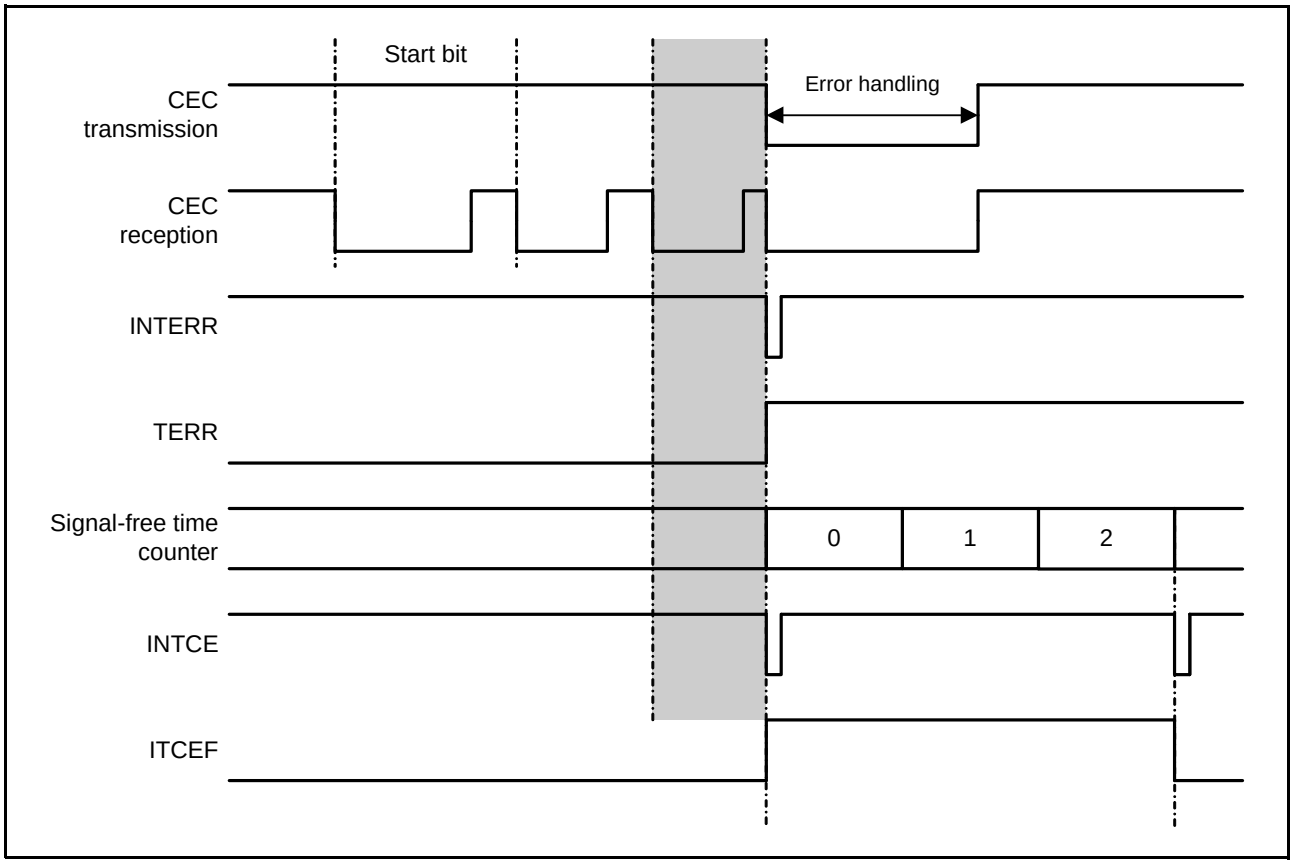


Figure 32.44 Timing Error When Bit Width is Short during Follower Operation

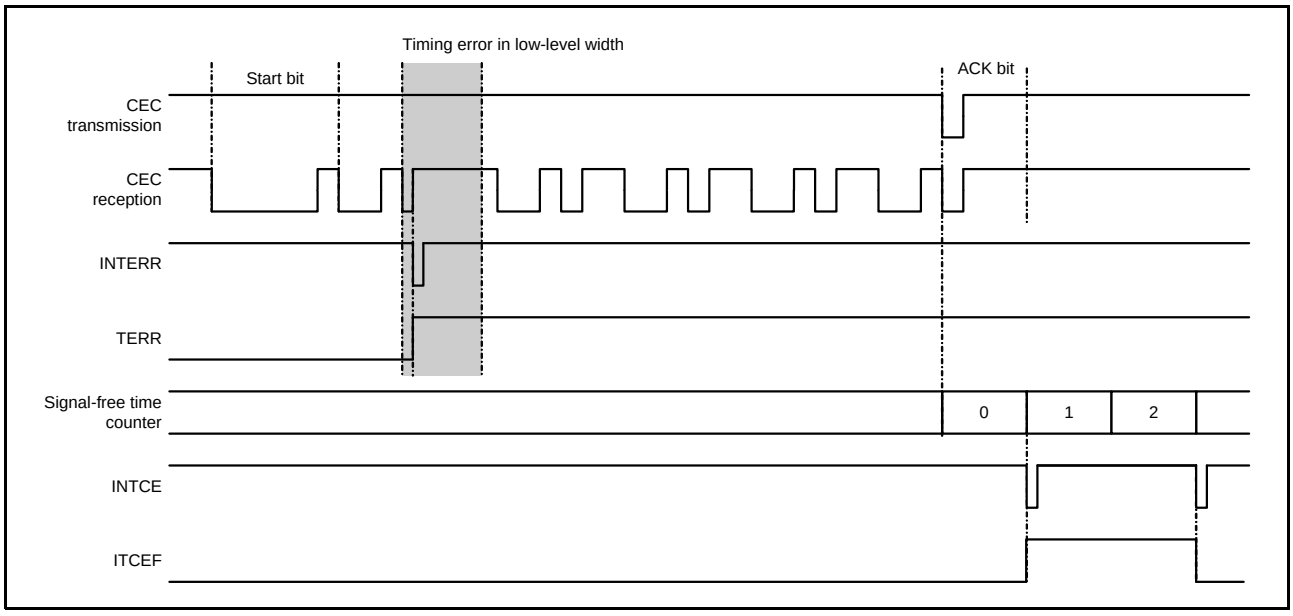


Figure 32.45 Timing Error When Bit Width is Not Short during Follower Operation

(6) Overrun error

If receiving the next data is completed before reading data from the reception buffer register (CRXD) during follower operation, an overrun error occurs. As shown in Figure 32.46, an error interrupt (INTERR) is generated, and the overrun error detection flag (CECES.OERR) is set. Afterward, logical 1 is returned during direct address communication and logical 0 is returned during broadcast communication at the ACK transmission timing of the block in which an overrun error occurred. The failure of reception is reported to the initiator, and the reception standby state is entered. A communication complete interrupt (INTCE) operates according to the setting of CECCTL1.CESEL[1:0].

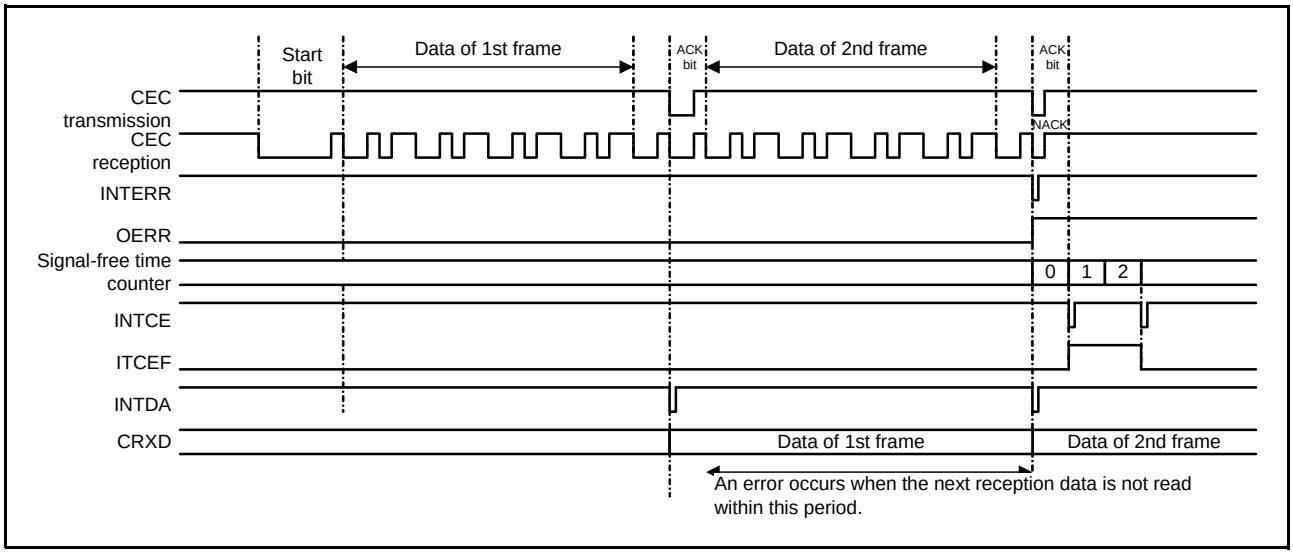


Figure 32.46 Overrun Error (When 3 Bits are Set as Signal-Free Time)

(7) Bus lock error

When bus lock errors are set to be detected (CECCTL1.BLERRD = 1), a bus lock error occurs if the bus is in the communication status (CECS.BUSST = 1) and the CEC reception signal stays at high or low level for a period corresponding to 2.5 times the data bit width specified by using NOMP. Figure 32.47 shows the timing of bus lock error detection.

If an error is detected, an error interrupt (INTERR) is generated, the bus lock error detection flag (CECES.BLERR) is set, the communication standby state is entered, and the signal-free time is counted. A communication complete interrupt (INTCE) operates according to the setting of CECCTL1.CESEL[1:0].

Bus lock errors are detected only for the follower.

When bus lock errors are not set to be detected (CECCTL1.BLERRD = 0), no bus lock error is detected. To determine whether the bus is locked, monitor the CECEXMON.CECLNMON bit, and use software to determine and handle bus locking.

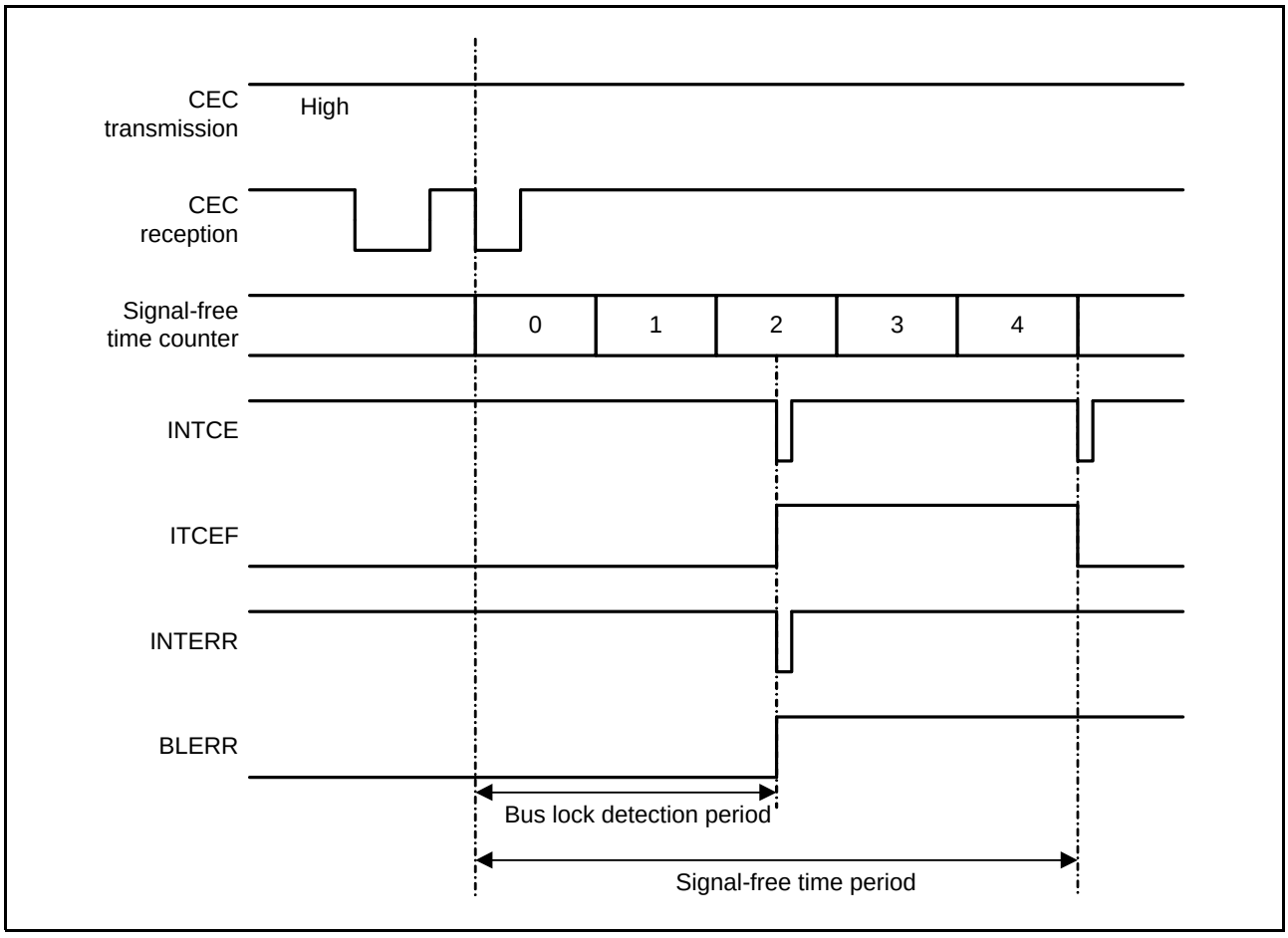


Figure 32.47 During Follower Operation (When 5 Bits are Set as Signal-Free Time)

32.3.3.7.2 Clearing Error Flag

An error flag set to the CEC communication error status register (CECES) can be cleared by setting 1 to the corresponding bit in the CEC communication error flag clear trigger register (CECFC). Figure 32.48 shows the case when an arbitration error occurs. An arbitration loss detection flag can be cleared by setting 20h to CECFC.

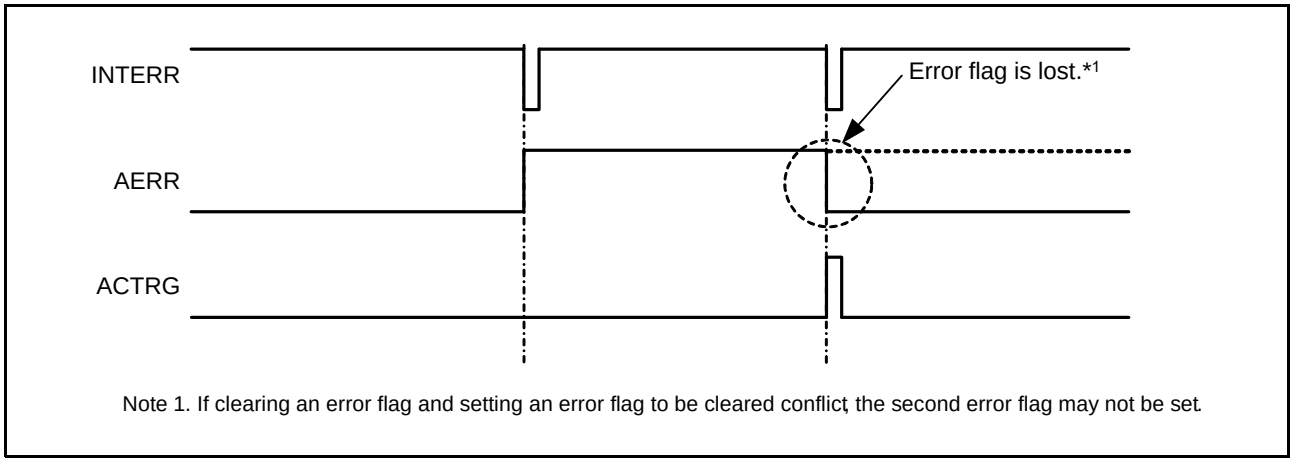


Figure 32.48 When Two Identical Errors Have Occurred and Conflict with Clear Trigger

32.3.3.8 Signal-Free Time

The end of the signal-free time is reported by generating a communication complete interrupt by detecting a match with the specified time (3, 5, or 7 bits of the bit width specified using the NOMP register). The number of bits of the signal-free time is specified using CECCTL1.SFT[1:0]. When a communication complete interrupt is generated can be selected by setting up CECCTL1.CESEL[1:0]. Counting the signal-free time always starts when the falling edge of the received data is detected. During normal communication, counting the signal-free time starts after the falling edge of the ACK bit is detected, if EOM is 1.

Even if an error occurs, counting the signal-free time is started after communication is stopped.

If an error handling pulse (a low-level pulse that has a bit width of 1.5 times the bit width) is acknowledged, the signal-free time is counted starting at the falling edge of the error handling pulse signal. Figure 32.49 shows an operation example in which the detection of a 7-data bit width signal-free time when CECCTL1.CESEL[1:0] = 00b and CECCTL1.SFT[1:0] = 10b.

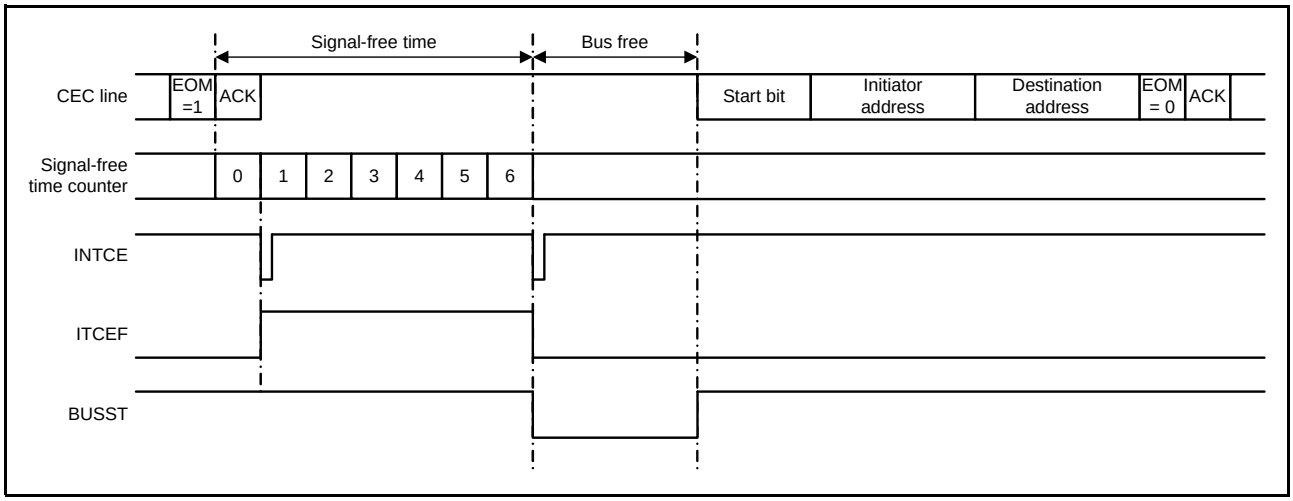


Figure 32.49 Signal-Free Time Operation

Rewriting the values of the CECCTL1.SFT[1:0] bits to the number of bits smaller than the current number of bits while counting the signal-free time must be completed until the rewritten bit count values match. If rewriting is not performed in time, the counter overflows and the signal-free time period continues until the number of bits match again. Figure 32.50 shows an example when the data bit width is changed from 5 to 3 bits.

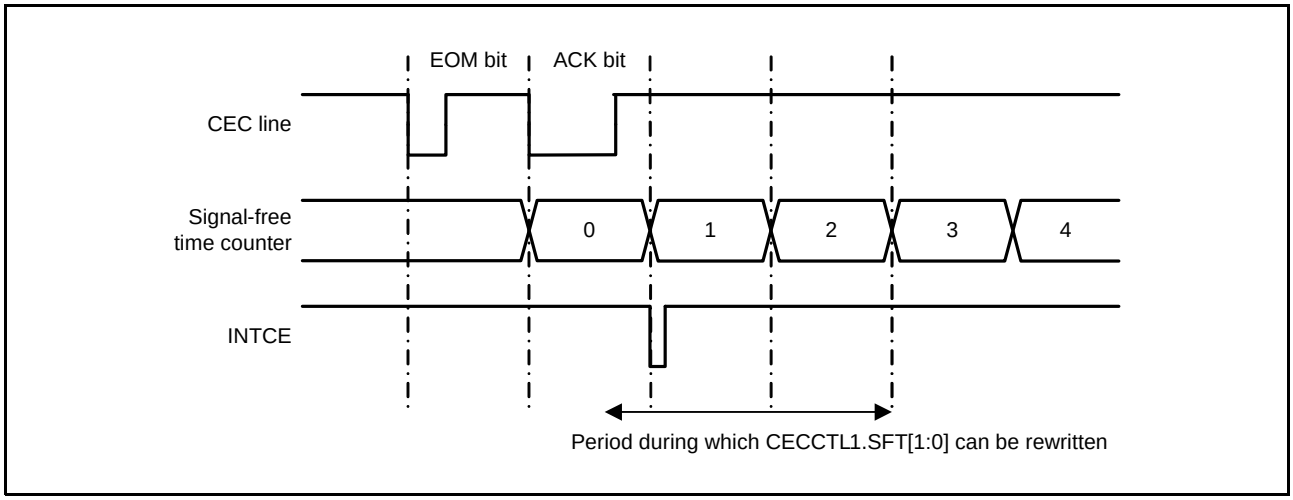


Figure 32.50 Period during Which CECCTL1.SFT[1:0] Can be Rewritten While Counting Signal-Free Time

[Starting receive operation during signal-free time]

If a falling edge of the CEC reception signal is detected while counting the signal-free time, a receive operation is started as shown in Figure 32.51. A communication complete interrupt (INTCE) is not output even if the output of INTCE after counting the signal-free time is set, because the count operation of the signal-free time counter is stopped at this time.

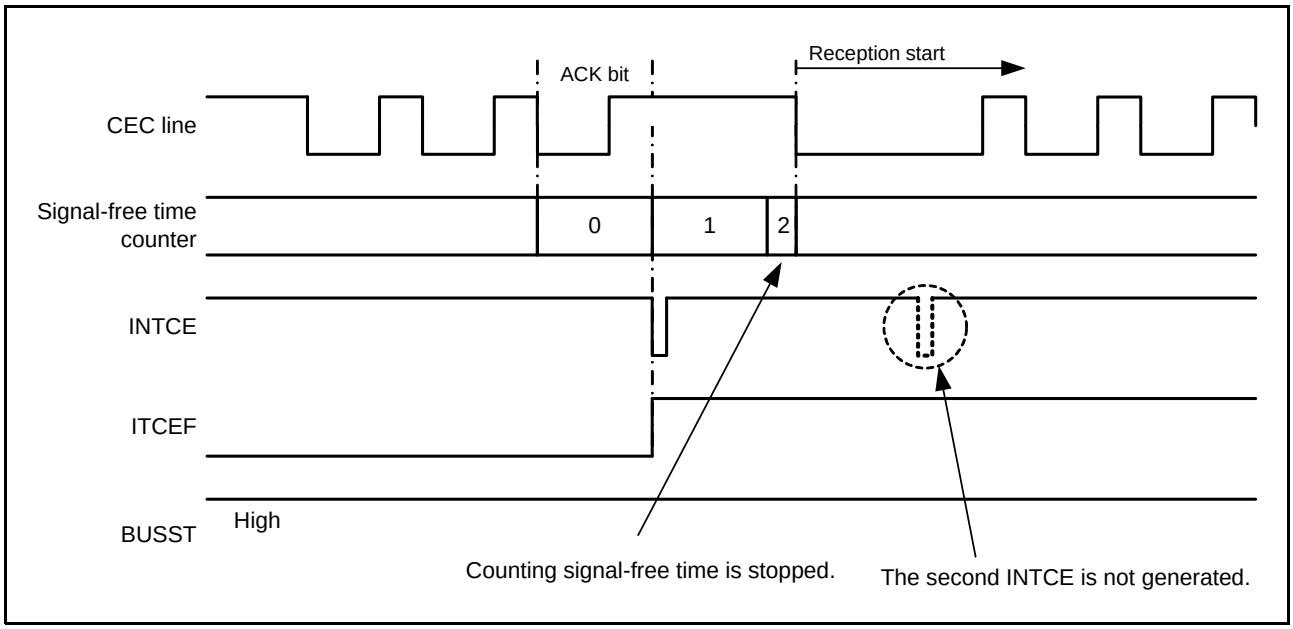


Figure 32.51 Starting Receive Operation during Signal-Free Time

32.3.3.9 Function That Restarts Reception by Detecting Start Bit during Reception

This function is used to restart reception starting from the detected start bit if a new start bit is detected during receive operation (follower operation), as shown in Figure 32.52.
This function is enabled by enabling the start detection restart reception enable bit (CECEXMD.RERCVEN = 1).

When detecting a start bit, the bit is determined as the start bit when received data matches the set value of the registers (STATLL, STATLH, STATBL, or STATBH) of the received start bit width, and reception is restarted.
Since the start bit is received during reception, a timing error occurs. An error occurs is either at the falling edge at the beginning of the start bit or the header block.

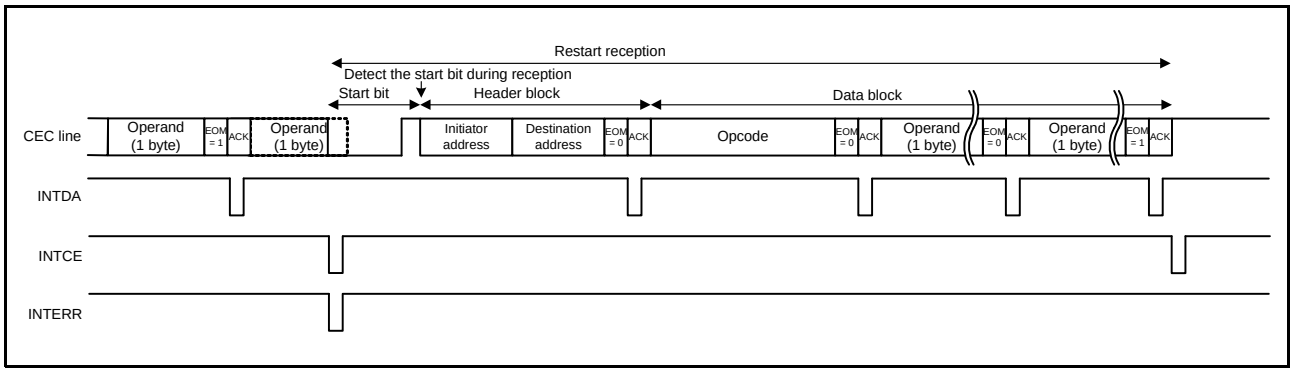


Figure 32.52 Reception Restart Function Operation

32.3.4 Data Reception in Low Power Consumption State

In this MCU, data can be received in a low power consumption state (sleep mode, all-module clock stop mode, or software standby mode).

To receive data in a low power consumption state, initial CEC communication settings should be performed before transitioning to the state. For details on the initial CEC communication settings, refer to section 32.3.3.3, Initial CEC Communication Settings. In this case, set the CECCTL0.CCL[2:0] bits to 110b or 111b, and select the divided CECILCLK or CECMCLK clock as the CEC operating clock (CECCLK).

Use the data reception interrupt (INTDAA) to be output during header block reception as the source for returning from the low power consumption state (see Figure 32.53). When returning from the low power consumption state using the CEC interrupt, the CEC operating clock (CECCLK) needs to be supplied constantly. Set the IWDTCSTPR.SLCSTP bit when the divided CECILCLK clock is selected as the CEC operating clock, and set the MOFCR.MOFXIN bit when the divided CECMCLK clock is selected as the CEC operating clock. The CEC interrupt uses the INTDAA, INTCEA, and INTERRA that can be set as the source for returning all-module clock stop mode or software standby mode.

When data other than data for the local address is received during direct address reception, no CEC interrupt is generated if CECCTL1.CINTMK has been cleared to 0, so the low power consumption state is continued. Figure 32.54 shows operation of receiving data other than data for the local station during the low power consumption state.

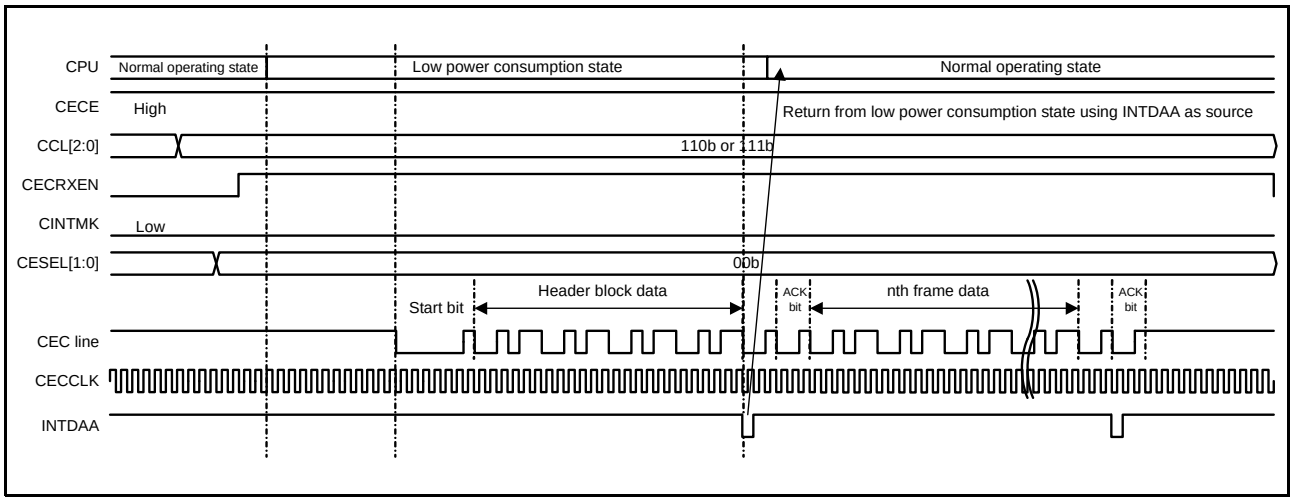


Figure 32.53 Returning from Low Power Consumption State Using CEC Interrupt

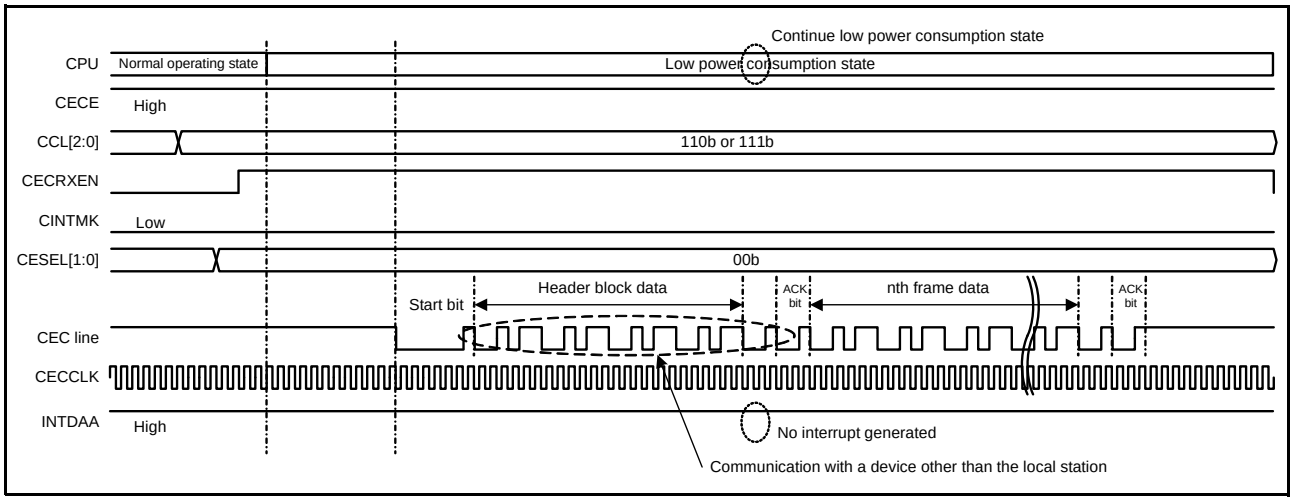


Figure 32.54 Receiving Data other than Data for Local Station during Low Power Consumption State

32.4 Usage Notes

32.4.1 Module Stop Function Setting

CEC operation can be disabled or enabled by setting the module stop control register.
The initial setting is for the CEC to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

32.4.2 Clock Division Ratio Setting

Set the peripheral module clock (PCLKB) frequency to higher than the CEC operating clock (CECCLK) frequency.

32.4.3 Restriction on Using IWDT Function

Do not use the IWDT function when using the divided CECILCLK clock as the CEC operating clock.

32.4.4 Notes on Using CECMCLK as Operating Clock

When the CECMCLK is selected as an operating clock by setting the CECCTL0.CCL[2:0] bits, the ranges of the frequency and the pulse width that can be input as the main clock differ, depending on whether the noise filter is enabled or disabled by the MONFCR register. For details, see section 41, Electrical Characteristics.

33. I2C Bus Interface (RIIC)

This MCU has a three-channel I2C bus interface (RIIC).
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.

33.1 Overview

Table 33.1 lists the specifications of the RIIC, Figure 33.1 shows a block diagram of the RIIC, and Figure 33.2 shows an example of I/O pin connections to external circuits (I2C bus configuration example). Table 33.2 lists the I/O pins of the RIIC.

Table 33.1 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)
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4 to 96%.
Issuing 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.
Acknowledgement	• For transmission, the acknowledge bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit. • For reception, the acknowledge bit is automatically transmitted. If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.
Wait function	• In reception, the following periods of waiting can be obtained by holding the SCL clock at the low level: Waiting between the eighth and ninth clock cycles Waiting between the ninth clock cycle and the first clock cycle of the next transfer
SDA output delay function	Timing of the output of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	• For multi-master operation Operation to synchronize the SCL clock in cases of conflict with the SCL signal from another master is possible. When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line. In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line. • Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions). • Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable. • Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL clock.
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: Error in transfer or occurrence of events Detection of arbitration, 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) • Transmit end

Table 33.1 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): Error in transfer or occurrence of events Detection of arbitration, 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) • Transmit end

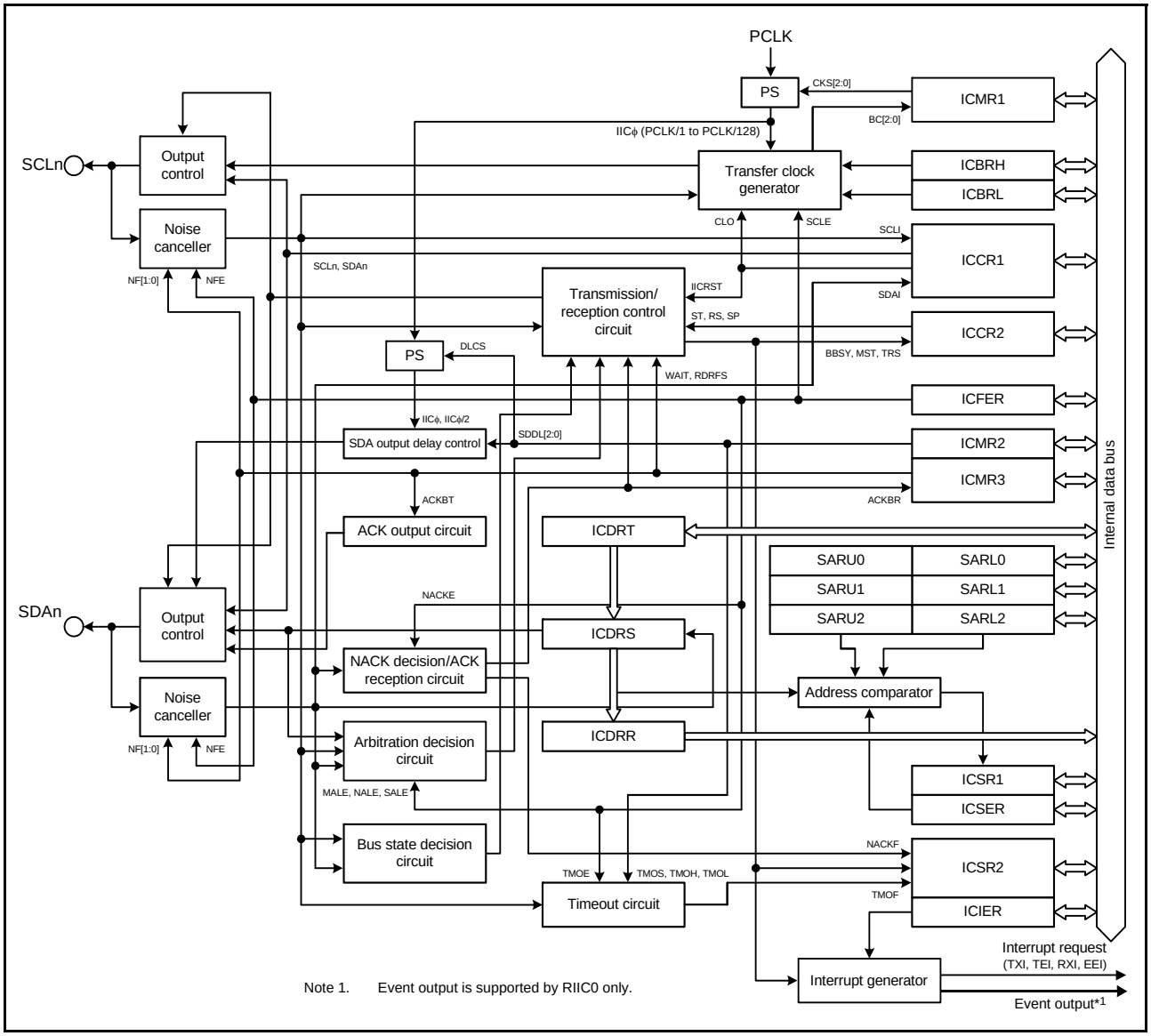


Figure 33.1 RIIC Block Diagram

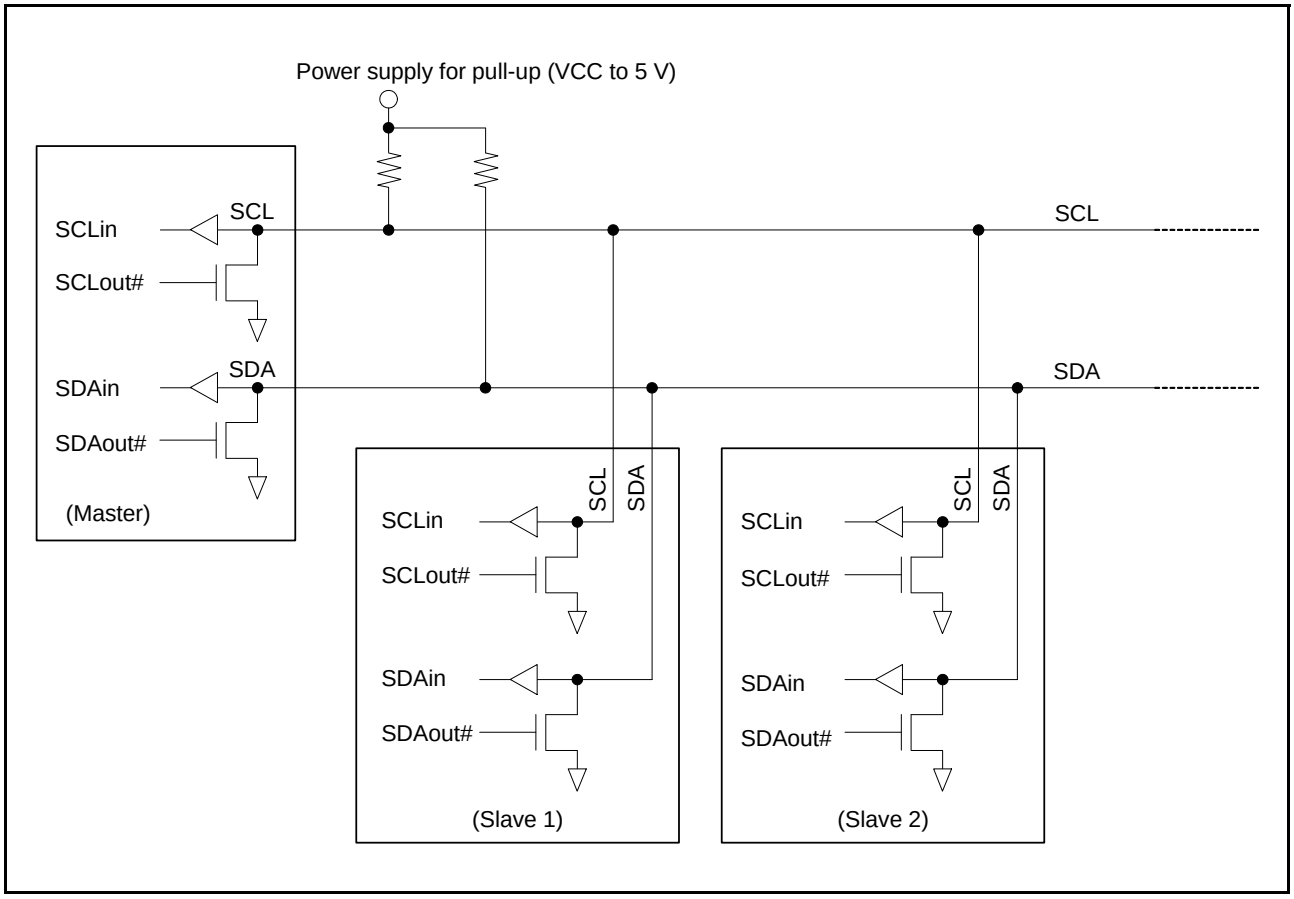


Figure 33.2 I/O Pin Connection to the External Circuit (I2C Bus Configuration Example)

The input level of the signals for RIIC is CMOS when I2C bus is selected (ICMR3.SMBS bit = 0), or TTL when SMBus is selected (ICMR3.SMBS bit = 1).

Table 33.2 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
RIIC1	SCL1	I/O	RIIC1 serial clock I/O pin
	SDA1	I/O	RIIC1 serial data I/O pin
RIIC3	SCL3	I/O	RIIC3 serial clock I/O pin
	SDA3	I/O	RIIC3 serial data I/O pin

33.2 Register Descriptions

33.2.1 I2C Bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC1.ICCR1 0008 8320h, RIIC3.ICCR1 0008 8360h

	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: SDAn line is low. 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low. 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDAn pin low. 1: The RIIC has released the SDAn pin. • Write: 0: The RIIC drives the SDAn pin low. 1: The RIIC releases the SDAn pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCLn pin low. 1: The RIIC has released the SCLn pin. • Write: 0: The RIIC drives the SCLn pin low. 1: The RIIC releases the SCLn 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	Extra SCL Clock Cycle Output	0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle 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 SCLn/SDAn output latch)	R/W
b7	ICE	I2C Bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn 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 SDAn and SCLn 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 (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle 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 33.11.2, Extra SCL Clock Cycle 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 33.3 lists the resets of the RIIC.
The RIIC reset resets all registers and internal states of the RIIC, and the internal reset resets 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 33.14, Resets and Register and Function States When Issuing Each Condition.
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 SCLn pin and SDAn 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 an internal reset is necessary because the RIIC hangs up with the SCLn line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 33.3 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, the ICSR1, ICSR2, and ICDRS registers, and the internal states of the RIIC.

ICE Bit (I2C Bus Interface Enable)

This bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 33.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCLn and SDAn 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 SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn 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.

33.2.2 I2C Bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC1.ICCR2 0008 8321h, RIIC3.ICCR2 0008 8361h

	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 Issuance Request	0: Does not request to issue a start condition. 1: Requests to issue a start condition.	R/W
b2	RS	Restart Condition Issuance Request	0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.	R/W
b3	SP	Stop Condition Issuance Request	0: Does not request to issue a stop condition. 1: Requests to issue 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, the MST and TRS bits can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free state).

For details on the start condition issuance, refer to section 33.10, Start Condition/Restart Condition/Stop Condition Issuing 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 issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.
When this bit is set to 1 to request to issue a restart condition, a restart condition is issued 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 issuance, refer to section 33.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (requests to issue a restart condition) is written to the RS bit in slave mode, the restart condition is not issued 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 issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.
When this bit is set to 1 to request to issue a stop condition, a stop condition is issued 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 issuance, refer to section 33.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 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 issued.

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 TRS bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally according to the restart condition issuance 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 ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- The AL (arbitration-lost) flag in ICSR2 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 ICSER 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 (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 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 issuing of a start condition and issuing or detection of a stop condition, etc. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 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 SDAn line changes from high to low under the condition of SCLn line = high, assuming that a start condition has been issued.
When the SDAn line changes from low to high under the condition of SCLn line = high, this bit is set to 0 after the bus free time (specified in ICBRL) start condition is not detected, assuming that a stop condition has been issued.
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in ICBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (RIIC reset)

33.2.3 I2C Bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC1.ICMR1 0008 8322h, RIIC3.ICMR1 0008 8362h

	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 source (IICϕ) for the RIIC. b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock	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 SCLn 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 acknowledge bit) between transferred frames when the SCLn line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

33.2.4 I2C Bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC1.ICMR2 0008 8323h, RIIC3.ICMR2 0008 8363h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			TMWE	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 SCLn line is at a low level. 1: Count-up is enabled while the SCLn line is at a low level.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCLn line is at a high level. 1: Count-up is enabled while the SCLn line is at a high level.	R/W
b3	TMWE	Timeout Internal Counter Write Enable	0: Writing to internal counter of timeout detection function is disabled. 1: Writing to internal counter of timeout detection function is enabled.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When ICMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 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 setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 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 = 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 SCLn line is in the state that enables this counter as specified by the TMOH and TMOL bits, 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 33.11.1, 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 SCLn line is held low when the timeout function is enabled (ICFER.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 SCLn line is held high when the timeout function is enabled (ICFER.TMOE bit = 1).

TMWE Bit (Timeout Internal Counter Write Enable)

This bit is used to select whether or not to allocate the timeout internal counter (TMOCNTL/TMOCNTU) to the address of the slave address register (SARL0/SARU0).
When this bit is set to 1, the addresses of the timeout internal counters (TMOCNTL and TMOCNTU) are allocated to the addresses of SARL0 and SARU0.

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 acknowledge bit.
Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1) or the SMBus standard (within the data hold time: 300 ns or more, and SCL-clock low-level period - the data setup time: 250 ns). Note that, if a value outside the standard 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 33.5, SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: Standard-mode [Sm])
900 ns (up to 400 kbps: Fast-mode [Fm])

33.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC1.ICMR3 0008 8324h, RIIC3.ICMR3 0008 8364h

	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	Receive Acknowledge	0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge 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 clock cycle. (The SCLn line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCLn line is held low at the falling edge of the eighth clock cycle.) 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 cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading ICDRR.	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 33.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock (IICϕ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.
[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge 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 issuance is detected (when a stop condition is detected with the SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 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 SCLn line low at the falling edge of the eighth SCL clock cycle.
When the RDRFS bit is 0, the SCLn line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCLn line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle 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 clock cycle 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 SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR 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 beforehand.

SMBS Bit (SMBus/I2C Bus Select)

Setting this bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

33.2.6 I2C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h, RIIC3.ICFER 0008 8365h

	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 Circuit Enable	0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.	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 33.11.1, 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 transfer operation when NACK is received from the slave device 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 received acknowledge content.
For details on the NACK reception transfer suspension function, refer to section 33.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.
When the SCLE bit is set to 0 (no SCL synchronous circuit used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the bus load of the I2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.
This bit must not be set to 0 except for checking the output of the set transfer rate.

33.2.7 I2C Bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC1.ICSER 0008 8326h, RIIC3.ICSER 0008 8366h

	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 SARL0 and SARU0 is disabled. 1: Slave address in SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in SARL1 and SARU1 is disabled. 1: Slave address in SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in SARL2 and SARU2 is disabled. 1: Slave address in 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 SARLy and SARUy.

When this bit is set to 1, the slave address set in 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 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 [W]: 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 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 frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, refer to section 33.7.3, 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 SMBS bit in ICMR3 is 1.
When this bit is set to 1 while the SMBS bit in ICMR3 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 SARLy and SARUy (y = 0 to 2) and performs the receive operation.
When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

33.2.8 I2C Bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h, RIIC1.ICIER 0008 8327h, RIIC3.ICIER 0008 8367h

	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	Transmit End Interrupt Request Enable	0: Transmit end interrupt (TEI) request is disabled. 1: Transmit 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 TMOF flag in ICSR2 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 AL flag in ICSR2 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 START flag in ICSR2 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 STOP flag in ICSR2 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 NACKF flag in ICSR2 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 RDRF flag in ICSR2 is set to 1.

TEIE Bit (Transmit End Interrupt Request Enable)

This bit is used to enable or disable transmit end interrupt (TEI) requests when the TEND flag in ICSR2 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 TDRE flag in ICSR2 is set to 1.

33.2.9 I2C Bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h, RIIC1.ICSR1 0008 8328h, RIIC3.ICSR1 0008 8368h

	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 frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]).	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 = 0

- When the received slave address matches the SVA[6:0] value in SARLy with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: SARUy.FS = 1

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address matches the SARLy value with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS = 0

- When the received slave address does not match the SVA[6:0] value in SARLy with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: SARUy.FS = 1

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy) with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address does not match the SARLy value with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the DIDE bit in ICSER set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the DIDE bit in ICSER set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the DIDE bit in ICSER set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 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 HOAE bit in ICSER set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the SMBS bit in ICMR3 or the HOAE bit in ICSER

- When the received slave address does not match the host address (0001 000b) with the HOAE bit in ICSER set to 1 (host address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

33.2.10 I2C Bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC1.ICSR2 0008 8329h, RIIC3.ICSR2 0008 8369h

	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: ICDRR contains no receive data. 1: ICDRR contains receive data.	R/(W) *1
b6	TEND	Transmit End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: ICDRT contains transmit data. 1: ICDRT 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 SCLn line state remains unchanged for a certain period.
[Setting condition]

- When the SCLn line state remains unchanged for the period specified by the ICMR2.TMOH, TMOL, and TMOS bits 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 IICRST bit in ICCR1 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 issued or an address and data are transmitted. The RIIC monitors the level on the SDAn 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 set the flag to indicate the detection of loss of arbitration during NACK transmission in master 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 SDAn line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDAn line is driven low while the internal SDA output is at a high level (the SDAn pin is in the high-impedance state))
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition issuance request) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is set to 1 (start condition issuance request) with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock 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 SDAn line level at the rising edge of SCL clock 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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Table 33.4 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 issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 with BBSY in ICCR2 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 IICRST bit in ICCR1 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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the NACKE bit in ICFER 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 IICRST bit in ICCR1 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 ICDRT in transmit mode or reading from ICDRR 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 ICDRS to ICDRR
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RDRFS bit in ICMR3)
- When the received slave address matches after a start condition (or a restart condition) is detected with the TRS bit in ICCR2 set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TEND Flag (Transmit End Flag)

[Setting condition]

- At the rising edge of the ninth SCL clock cycle 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 ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1

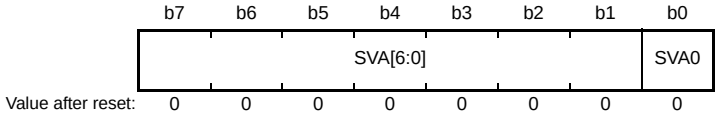
[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the ICDRS register and the ICDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

33.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC1.SARL0 0008 832Ah, RIIC3.SARL0 0008 836Ah, RIIC0.SARL1 0008 830Ch, RIIC1.SARL1 0008 832Ch, RIIC3.SARL1 0008 836Ch, RIIC0.SARL2 0008 830Eh, RIIC1.SARL2 0008 832Eh, RIIC3.SARL2 0008 836Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	A slave address is set.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set.	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 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 SARyE bit in ICSER 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 = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the SARyE bit in ICSER is 0, the setting of these bits is ignored.

33.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC1.SARU0 0008 832Bh, RIIC3.SARU0 0008 836Bh, RIIC0.SARU1 0008 830Dh, RIIC1.SARU1 0008 832Dh, RIIC3.SARU1 0008 836Dh, RIIC0.SARU2 0008 830Fh, RIIC1.SARU2 0008 832Fh, RIIC3.SARU2 0008 836Fh

	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	A slave address is set.	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 SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (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 SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

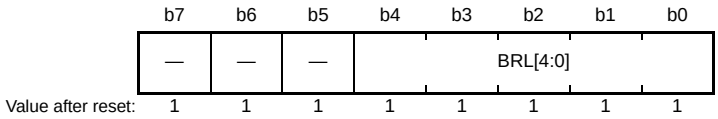
When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.

When the SARyE bit in ICSER 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.

33.2.13 I2C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC1.ICBRL 0008 8330h, RIIC3.ICBRL 0008 8370h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-Level Period	Low-level period of SCL clock	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-level period of SCL clock.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 33.8, 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-level period with the internal reference clock source (IICϕ) specified by the CKS[2:0] bits in ICMR1.
If the digital noise filter is enabled (the NFE bit in ICFER 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])

33.2.14 I2C Bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC1.ICBRH 0008 8331h, RIIC3.ICBRH 0008 8371h

	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-Level Period	High-level period of SCL clock	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-level period of SCL clock. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.
ICBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the CKS[2:0] bits in ICMR1.
If the digital noise filter is enabled (the NFE bit in ICFER 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 clock duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICϕ*1 + SCLn line rising time [tr] + SCLn line falling time [tf]}
Duty cycle = {SCLn line rising time [tr]*2 + (ICBRH + 1) / IICϕ} / {SCLn line falling time [tf]*2 + (ICBRL + 1) / IICϕ}

Note 1. IICϕ = PCLK × Division ratio
Note 2. The SCLn line rising time [tr] and SCLn line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C bus standard from NXP Semiconductors.

Table 33.5 lists examples of ICBRH/ICBRL settings.

Table 33.5 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		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)
100	010b	2 (E2h)	3 (E3h)	011b	15 (EFh)	18 (F2h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCLn line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCLn line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCLn line rising time (tr) and SCLn line falling time (tf), see the I2C bus standard from NXP Semiconductors.

33.2.15 I2C Bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC1.ICDRT 0008 8332h, RIIC3.ICDRT 0008 8372h

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

When ICDRT detects a space in the I2C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.
The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data has been written to ICDRT while the ICDRS data is being transmitted.
ICDRT can always be read and written. Write transmit data to ICDRT once when a transmit data empty interrupt (TXI) request is generated.

33.2.16 I2C Bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC1.ICDRR 0008 8333h, RIIC3.ICDRR 0008 8373h

	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 ICDRR to enable the next data to be received.
The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data has been read from ICDRR while ICDRS is receiving data.
ICDRR cannot be written. Read data from ICDRR once when a receive data full interrupt (RXI) request is generated.
If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

33.2.17 I2C Bus Shift Register (ICDRS)

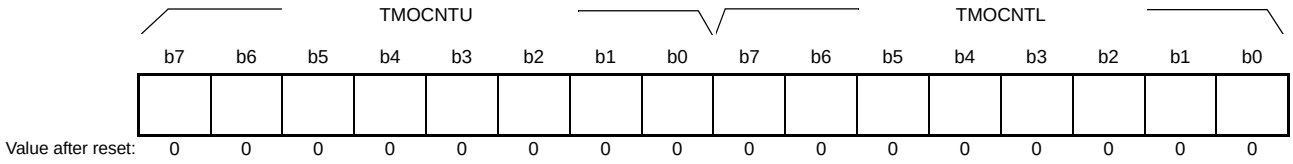
	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.
ICDRS cannot be accessed directly.

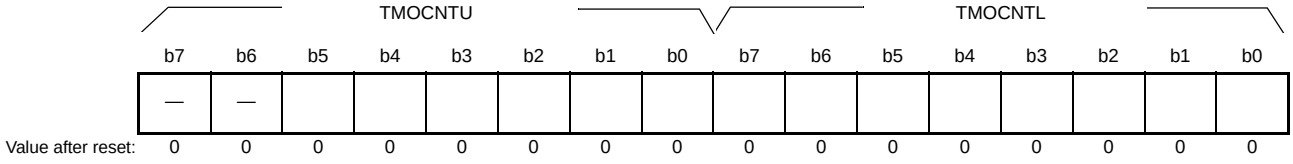
33.2.18 Timeout Internal Counter (TMOCNT)

Address(es): RIIC0.TMOCNTL 0008 830Ah, RIIC0.TMOCNTU 0008 830Bh,
RIIC1.TMOCNTL 0008 832Ah, RIIC1.TMOCNTU 0008 832Bh,
RIIC3.TMOCNTL 0008 836Ah, RIIC3.TMOCNTU 0008 836Bh

- TMOS = 0 (Long mode)



- TMOS = 1 (Short mode)



Note: Same address with ones of the slave address registers, SARL0, SARU0. Care should be taken.

- TMOCNTL register

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMOCNTL	Timeout internal counter	Timeout internal counter low-order	W*1

Note 1. The value in timeout internal counter cannot be read. When the value is read, the read value is FFh.

- TMOCNTU register

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMOCNTU	Timeout internal counter	Timeout internal counter high-order*1	W*2

Note 1. When TMOS = 1 (short mode is selected), bits b7 and b6 are reserved bits. They are writable. However, the value written is disabled.
Note 2. The value in timeout internal counter cannot be read. When the value is read, the read value is FFh.

The timeout internal counter (TMOCNTL/TMOCNTU) is initialized (TMOCNTL = 00h, TMOCNTU = 00h) after a reset, while ICCR1.IICRST = 1 or ICFER.TMOE = 1 and PCLK/1 is selected with ICMR1.CKS[2:0] = 000b setting, and when the counter clear conditions specified by the ICMR2.TMOH and TMOL bits (SCL rising edge/falling edge detection) are met.
Since the timeout internal counter is not automatically initialized when the ICMR1.CKS[2:0] bits are not 000b (PCLK/1), write 00h to the TMOCNTL and TMOCNTU counters as necessary for initialization.
The TMOCNTL and TMOCNTU counters can be accessed as 16-bit registers in 16-bit units.
For 16-bit access, access the address shown in the table below.

Table 33.6 Register Allocation for 16-Bit Access

Address	Upper 8 Bits	Lower 8 Bits
0008 830Ah	RIIC0.TMOCNTU	RIIC0.TMOCNTL

33.3 Operation

33.3.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame 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 issued.
Figure 33.3 shows the I2C bus format, and Figure 33.4 shows the I2C bus timing.

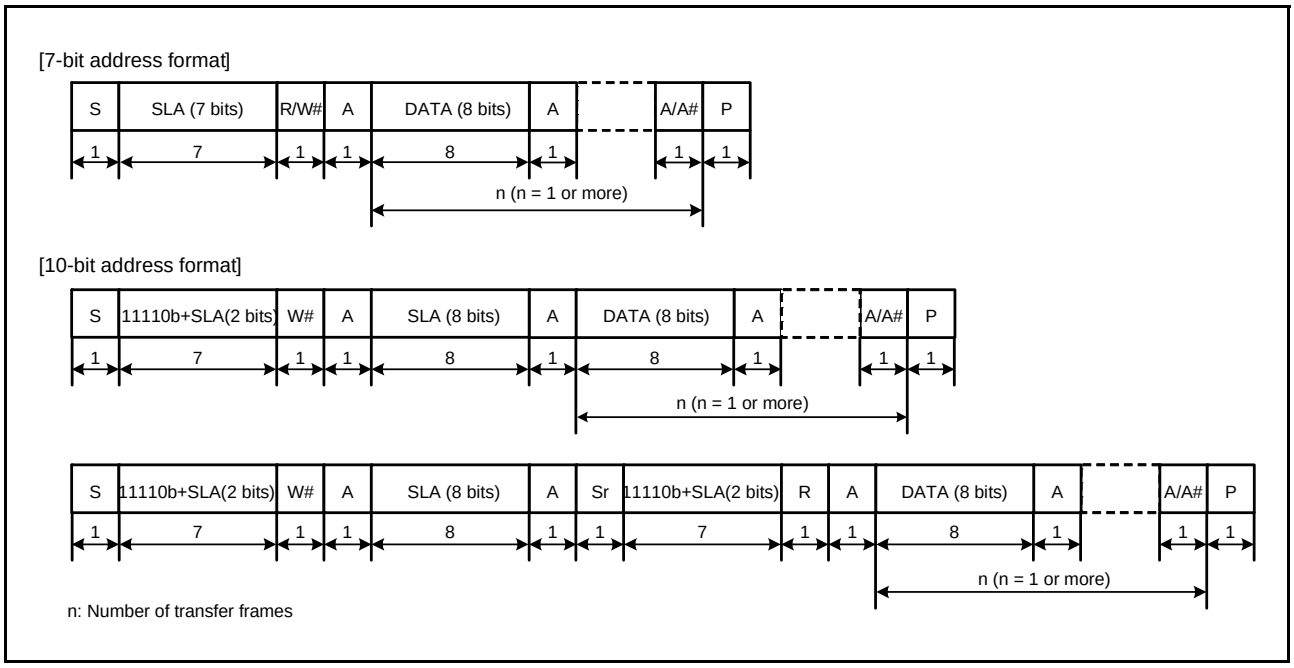


Figure 33.3 I2C Bus Format

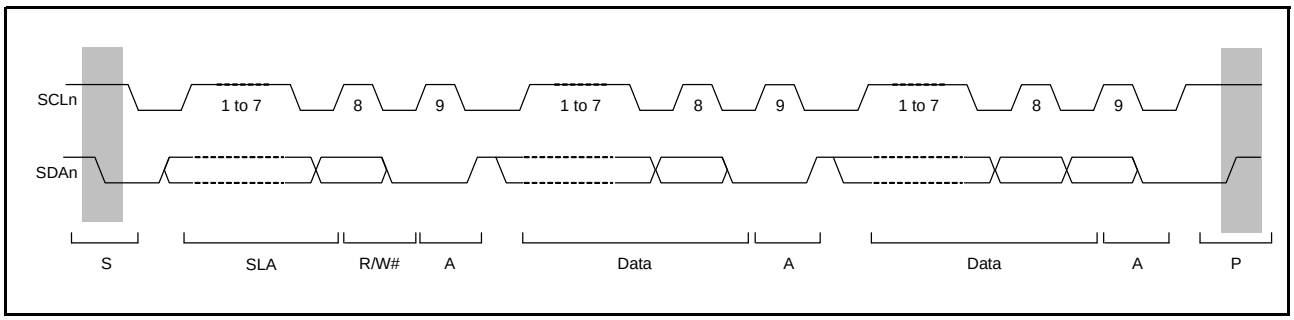


Figure 33.4 I2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDAn line low from high level while the SCLn line is at a high level.

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 receive device drives the SDAn line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receive device drives the SDAn line high.

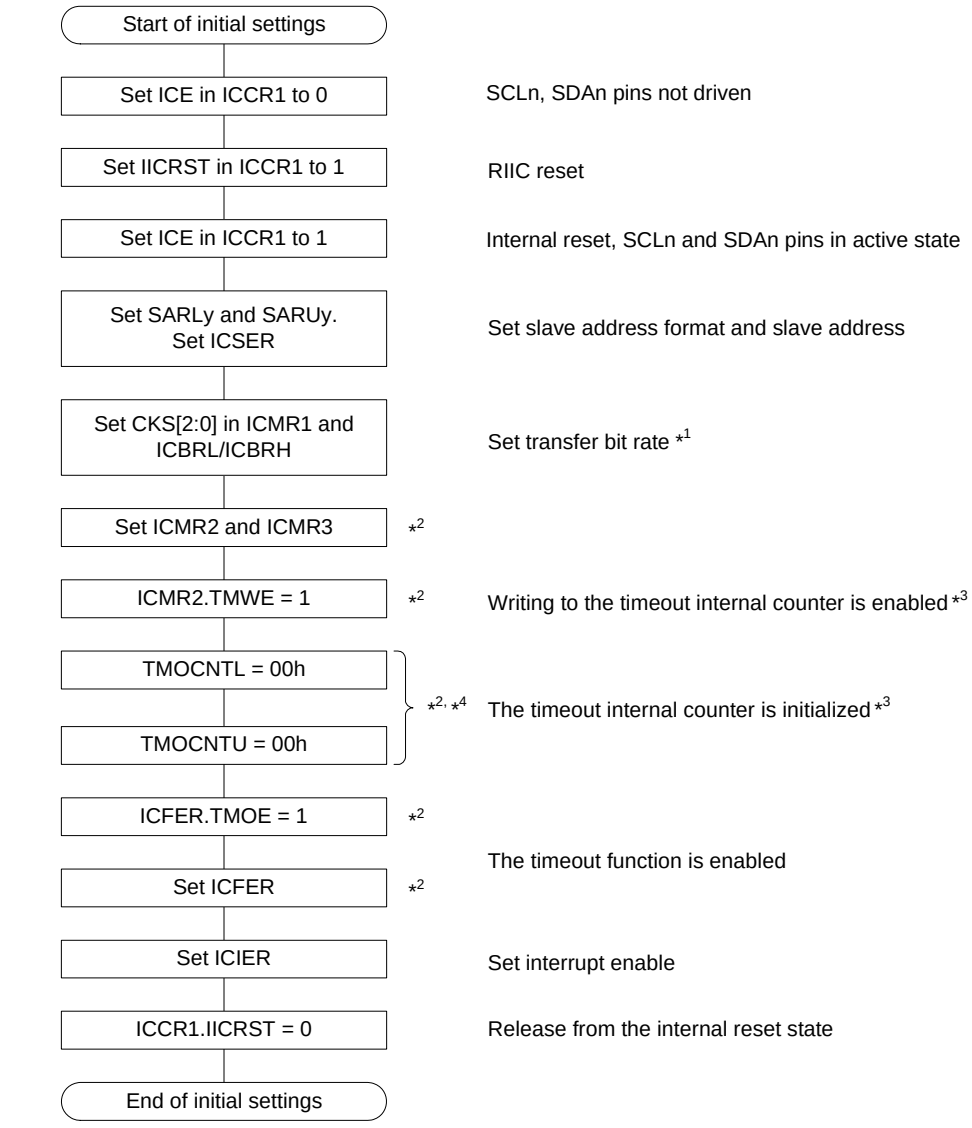
Sr: Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDAn line high from low level while the SCLn line is at a high level.

33.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 33.5.
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 (SCLn and SDAn 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 33.5). 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.



y = 0 to 2

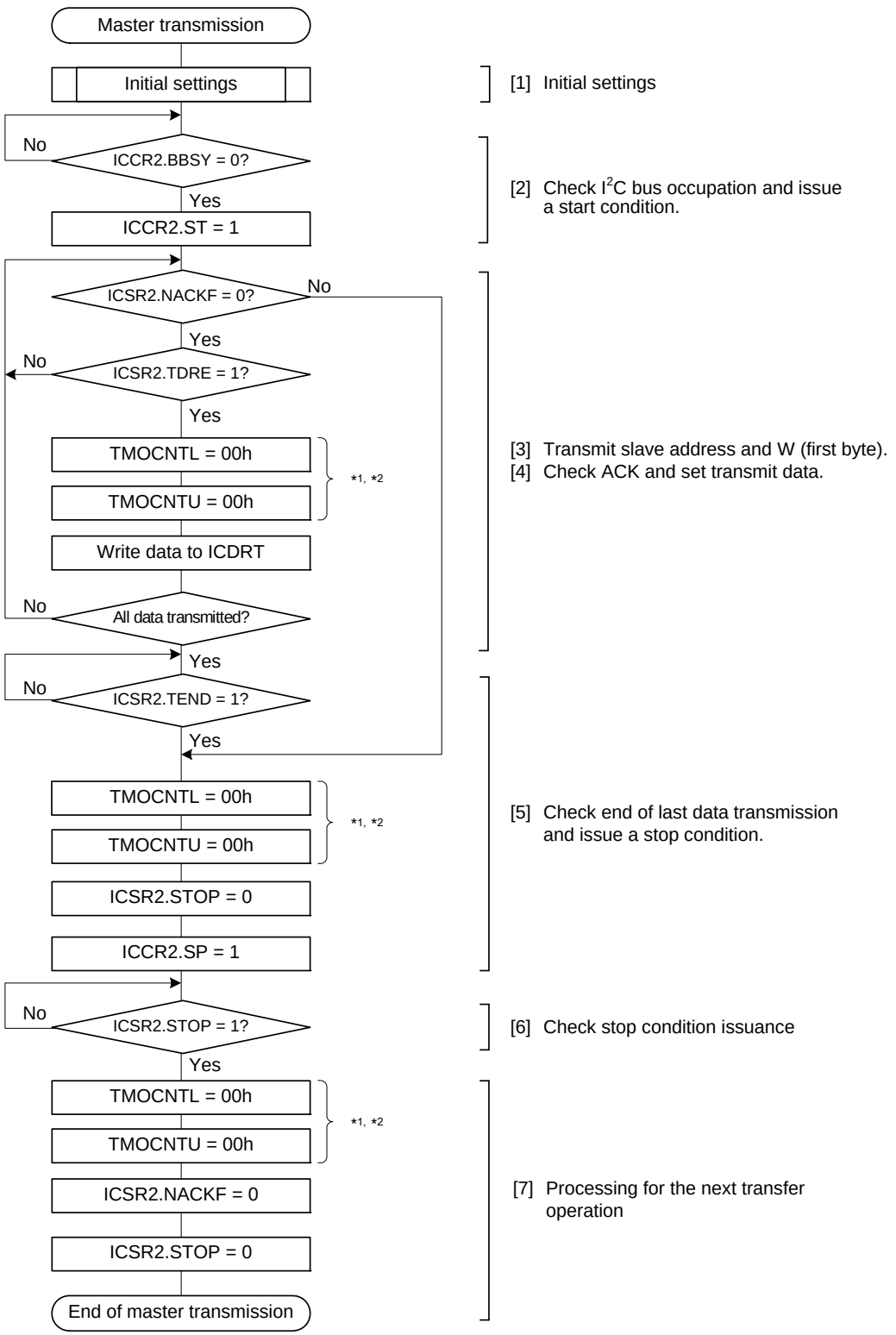
Note 1. When the RIIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as necessary.
Note 3. These processes are necessary when the timeout function is used and the ICMR1.CKS[2:0] bits are not 000b.
Note 4. Can be accessed as a 16-bit register in words.

Figure 33.5 Example of RIIC Initialization Flowchart

33.3.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgements. Figure 33.6 shows an example of usage of master transmission and Figure 33.7 to Figure 33.9 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 33.3.2, Initial Settings.
(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 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 SDAn line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically set to 0, the data are transferred from ICDRT to ICDRS, 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.
Since 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 issue 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 ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.
(4) After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCLn line low until the data for transmission are ready or a stop condition is issued.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 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.



Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.

Figure 33.6 Example of Master Transmission Flowchart

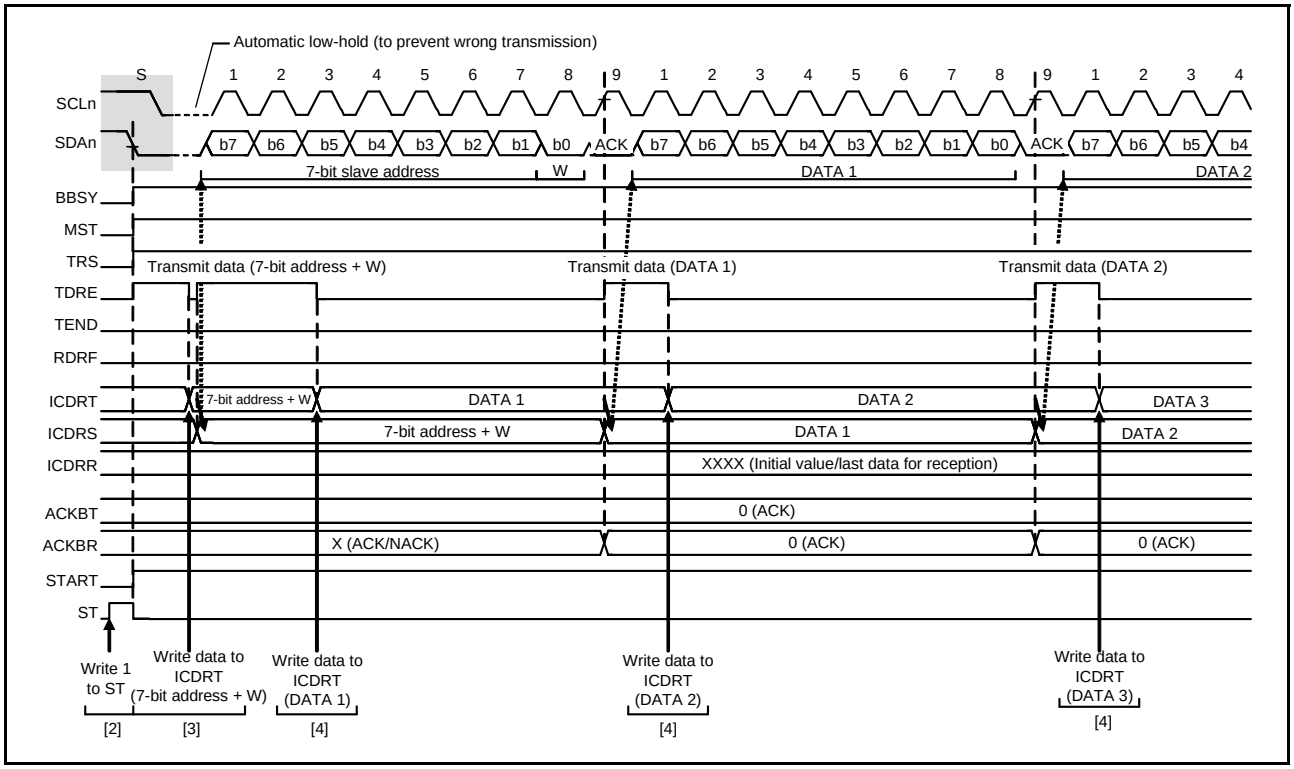


Figure 33.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

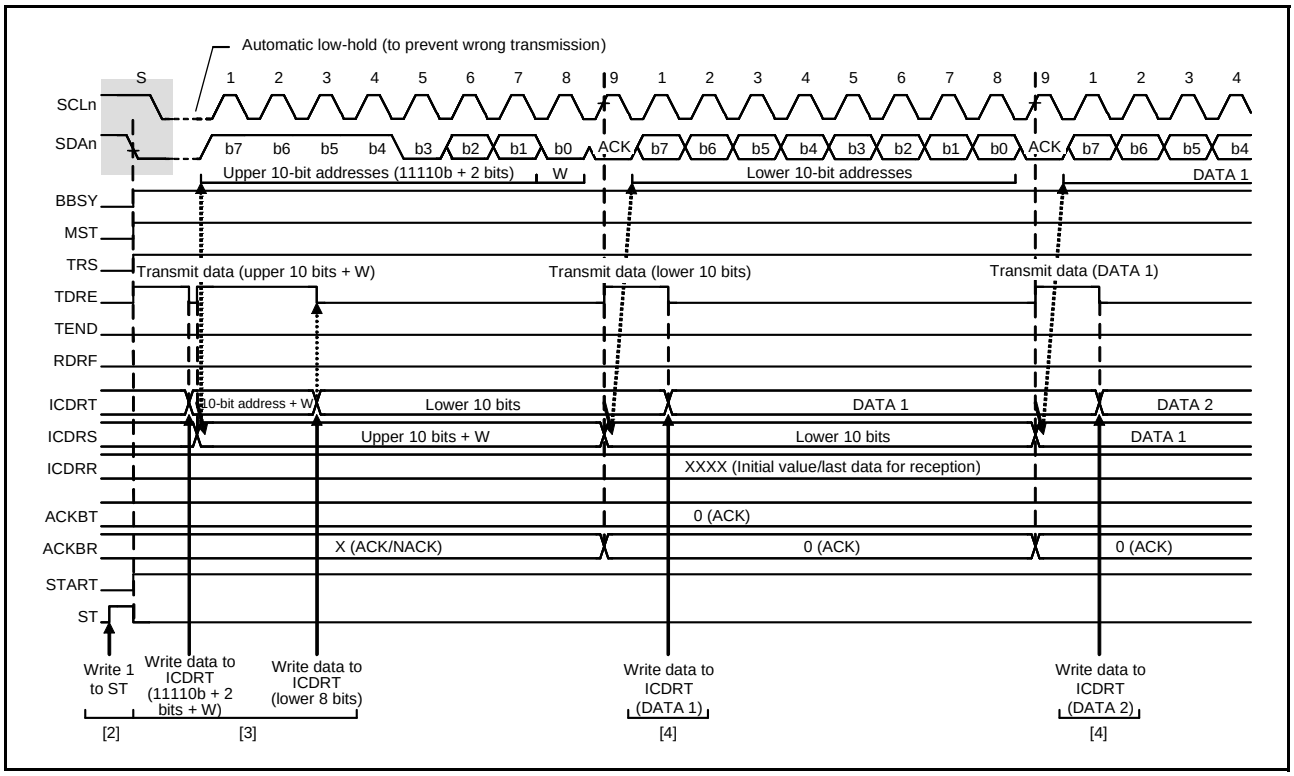


Figure 33.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

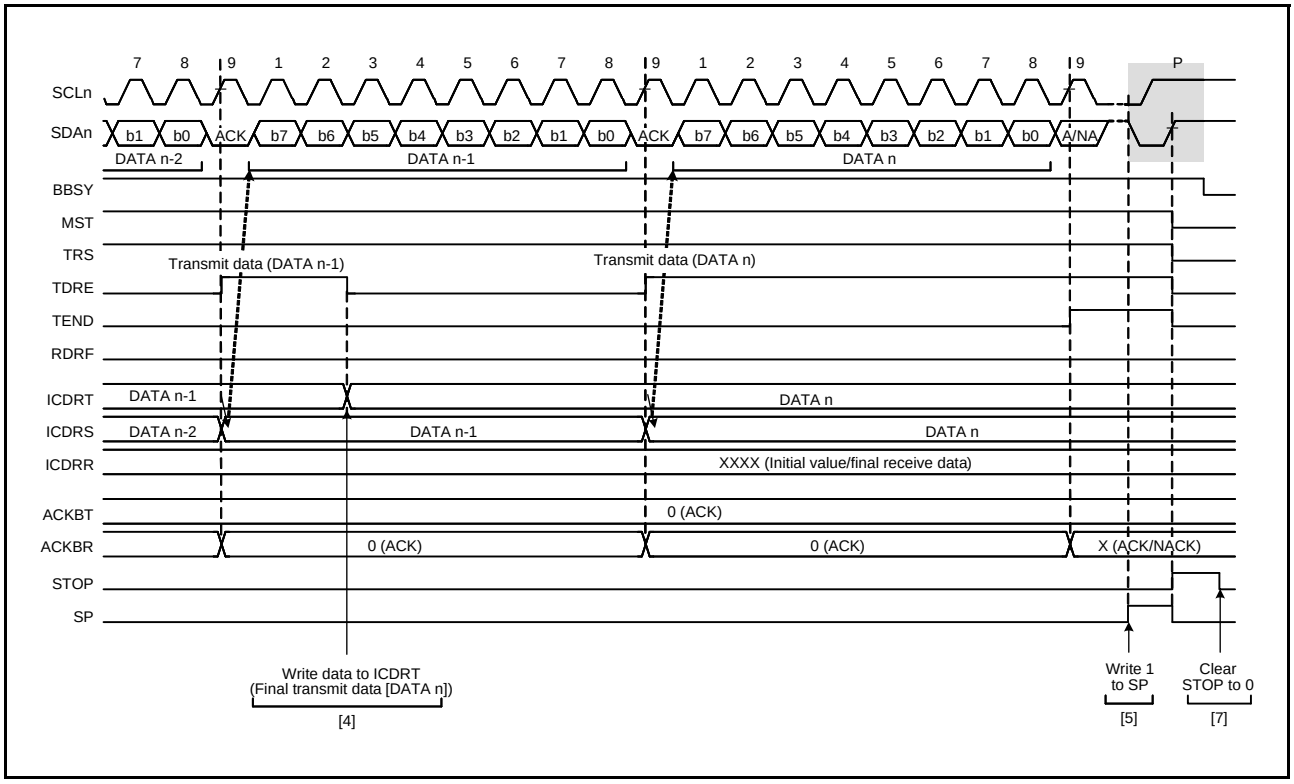


Figure 33.9 Master Transmit Operation Timing (3)

33.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgements. Since 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 33.10 and Figure 33.11 show examples of usage of master reception (7-bit address format) and Figure 33.12 to Figure 33.14 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 33.3.2, Initial Settings.

(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the START flag in ICSR2 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 SDAn line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically set to 0, the data are transferred from ICDRT to ICDRS, 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 cycle of SCL clock, 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.

Since 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 issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue 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 ICDRR after confirming that the RDRF flag in ICSR2 is 1; this makes the RIIC start output of the SCL clock and start data reception.

(5) After 1 byte of data has been received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RDRFS bit in ICMR3. Reading ICDRR 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 acknowledgement field received during the ninth cycle of SCL clock 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 (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 SCLn line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing 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 SP bit in ICCR2 (stop condition issuance request) and then read the last byte from ICDRR. When ICDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCLn line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets the MST and TRS bits in ICCR2 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.

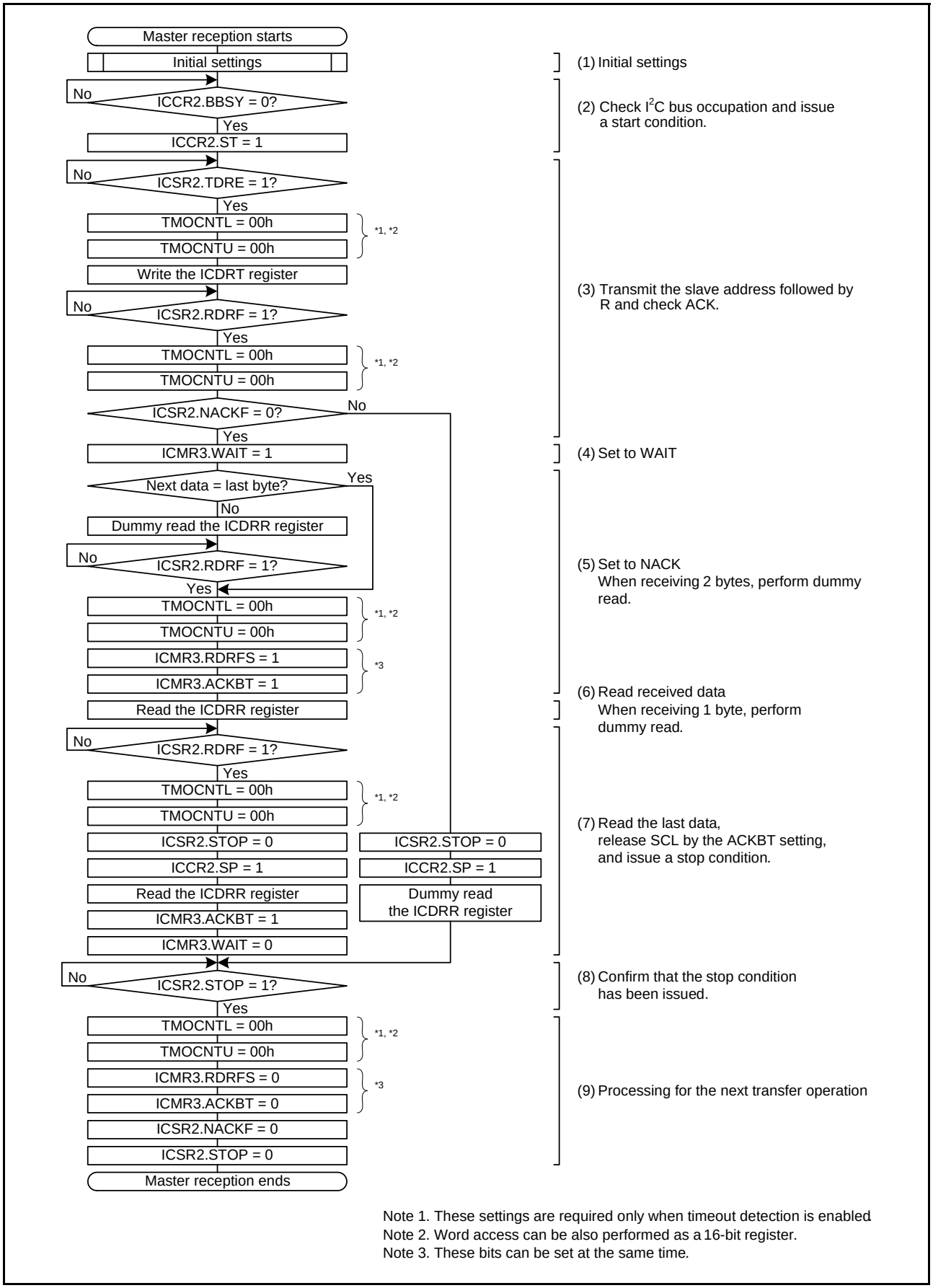
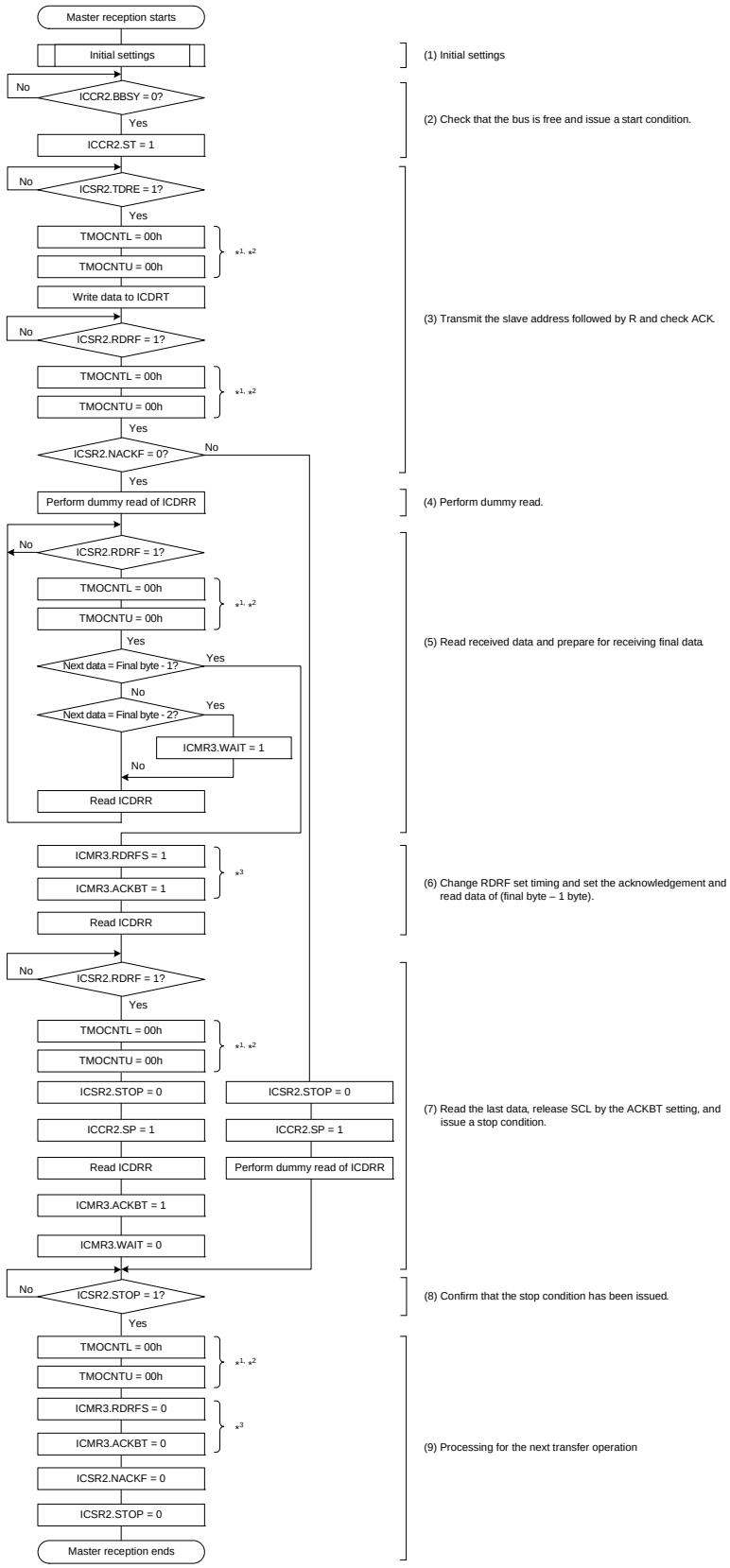


Figure 33.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)



Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.
Note 3. These bits can be set at the same time.

Figure 33.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

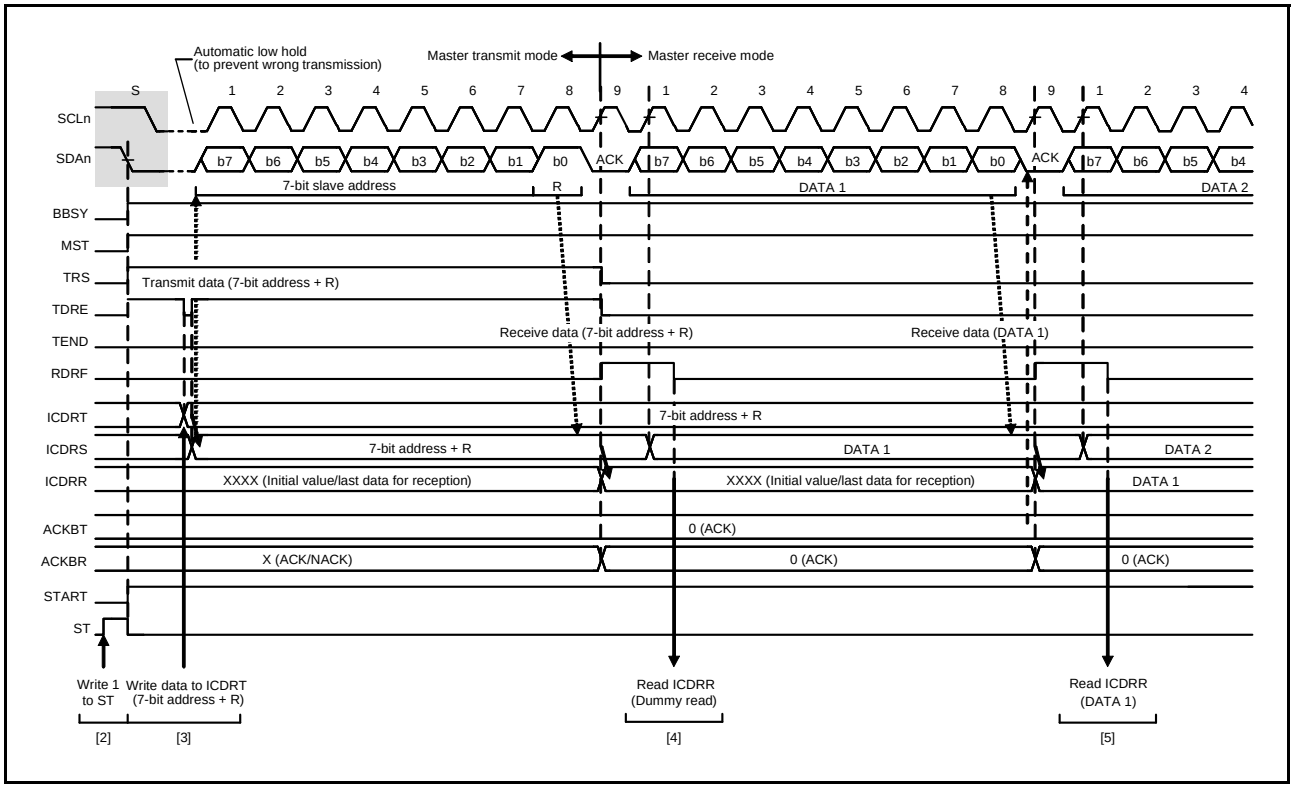


Figure 33.12 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS = 0)

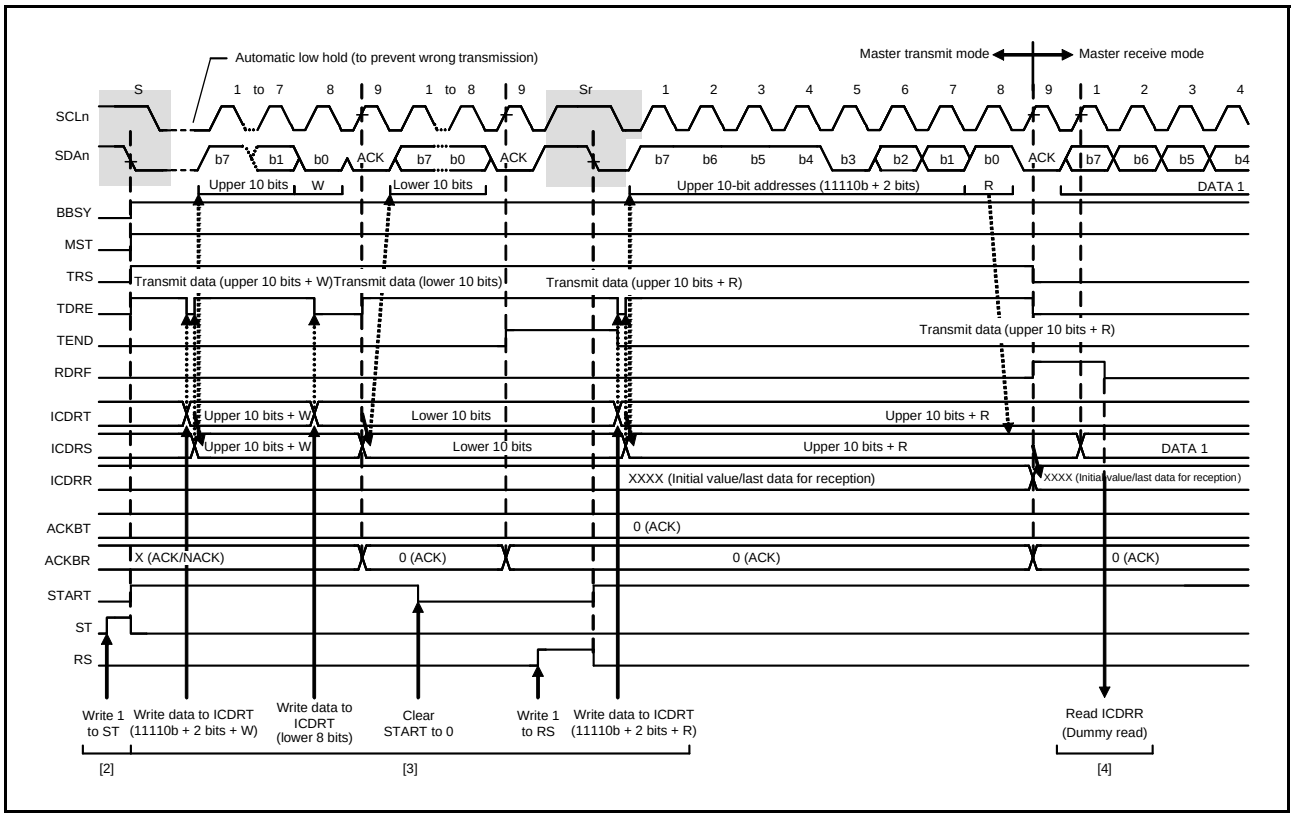


Figure 33.13 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS = 0)

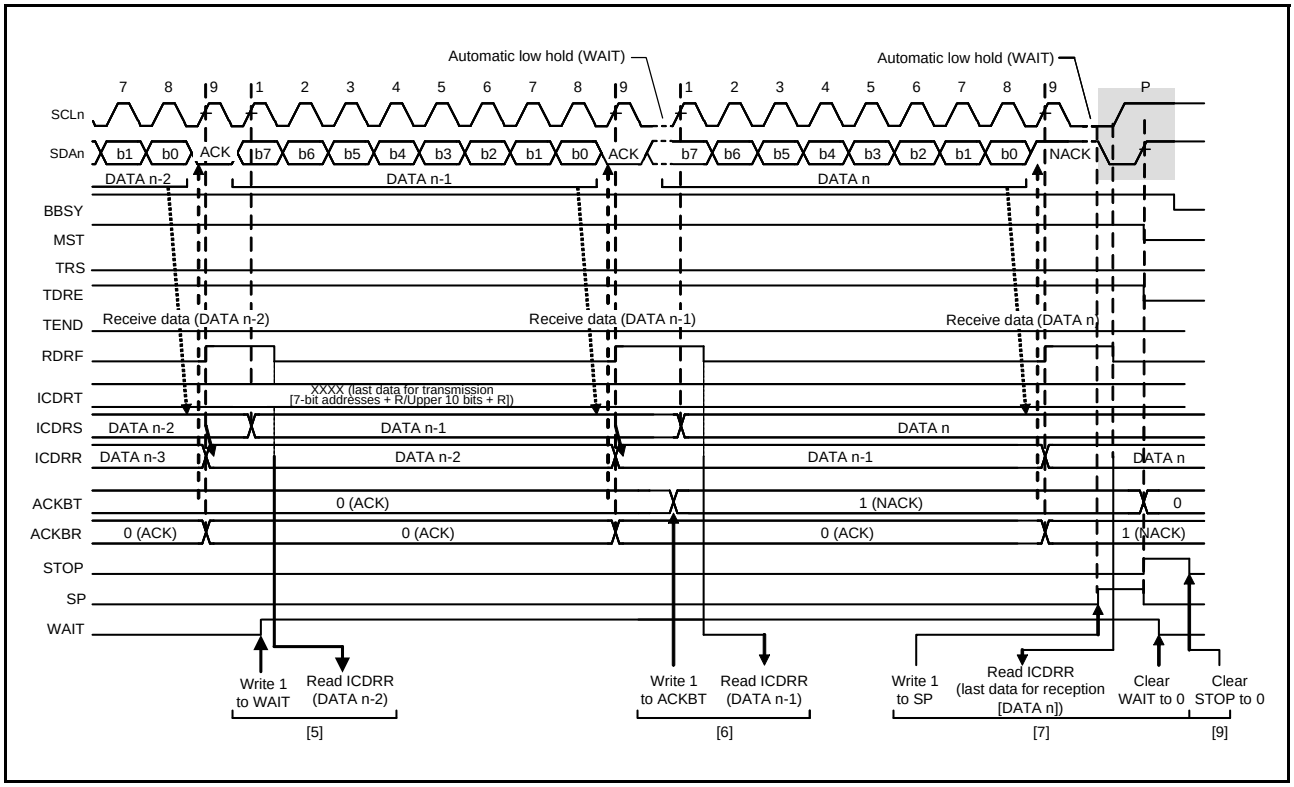
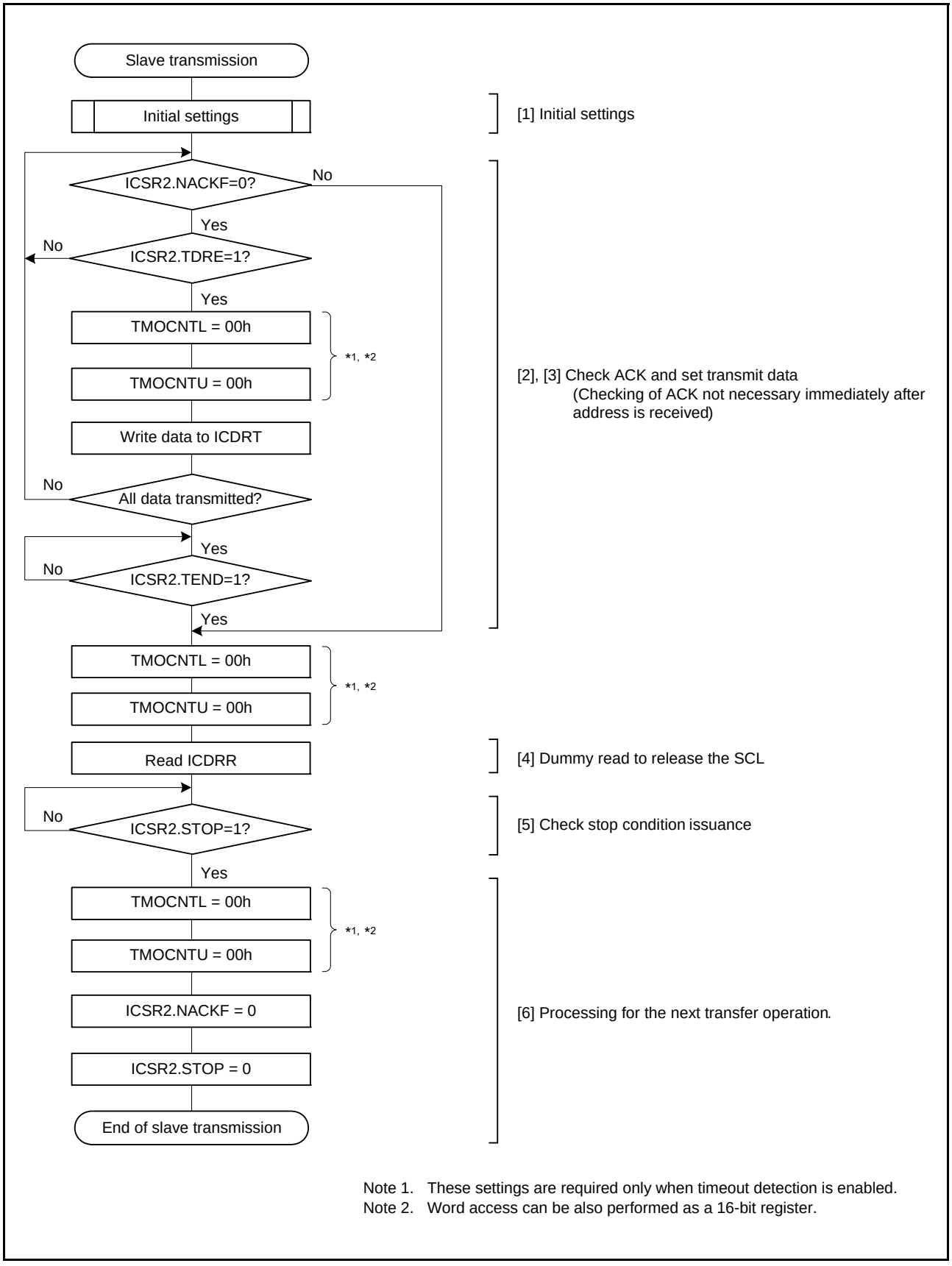


Figure 33.14 Master Receive Operation Timing (3) (When RDRFS = 0)

33.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the RIIC transmits data as a slave device, and the master device returns acknowledgements.
Figure 33.15 shows an example of usage of slave transmission and Figure 33.16 and Figure 33.17 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 33.3.2, 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 cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. 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 receives no acknowledge 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 SCLn line low on the ninth falling edge of SCL clock.
(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn 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.



Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.

Figure 33.15 Example of Slave Transmission Flowchart

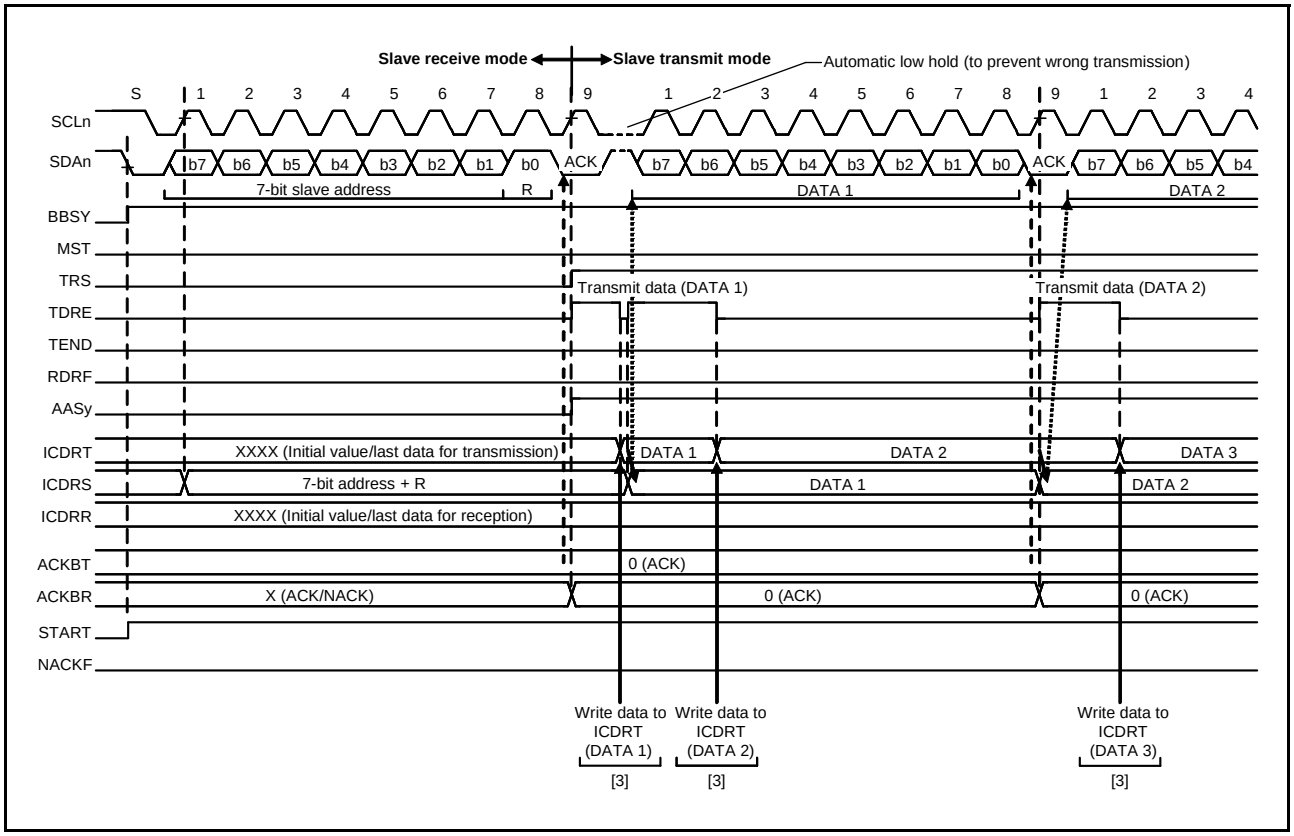


Figure 33.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

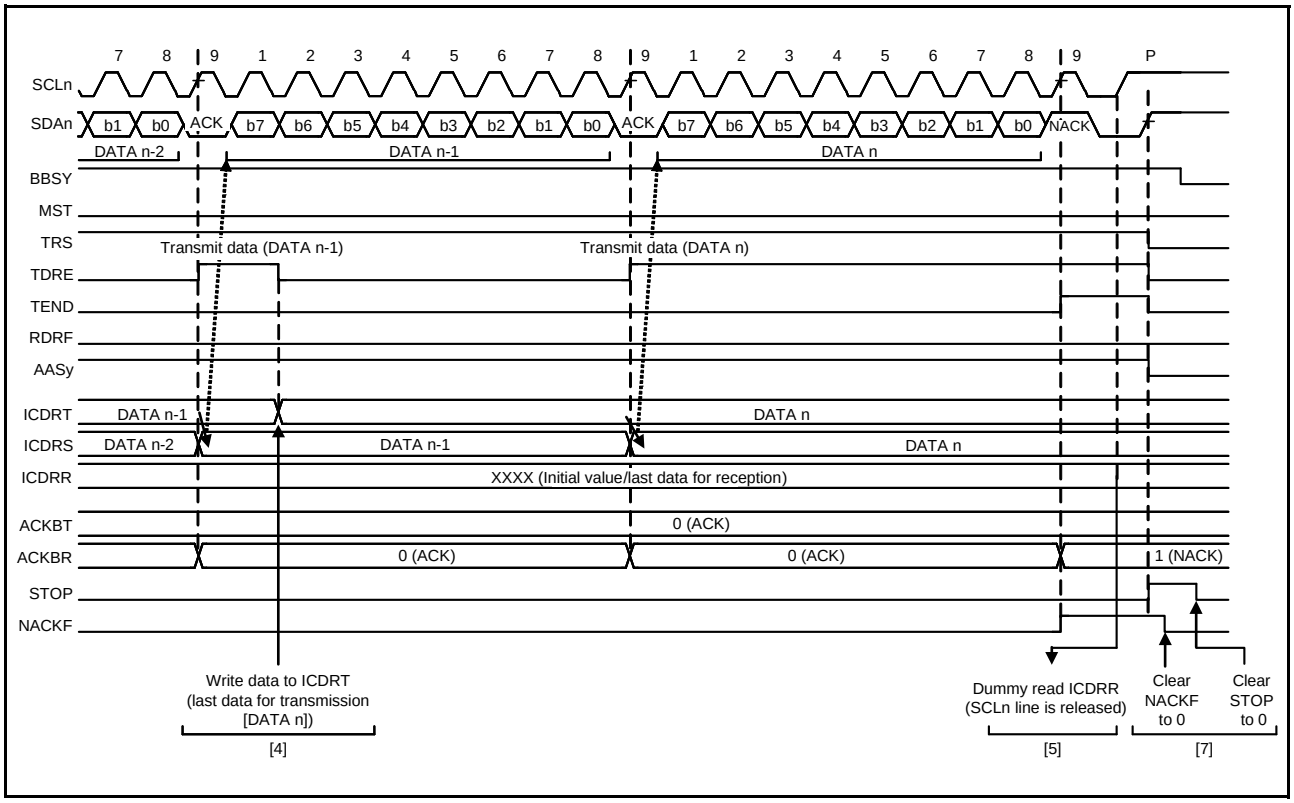


Figure 33.17 Slave Transmit Operation Timing (2)

33.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 33.18 shows an example of usage of slave reception and Figure 33.19 and Figure 33.20 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 33.3.2, 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 cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. 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 RDRF flag in ICSR2 to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read ICDRR (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 ICDRR is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCLn line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCLn line from being held at the low level.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.

(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.

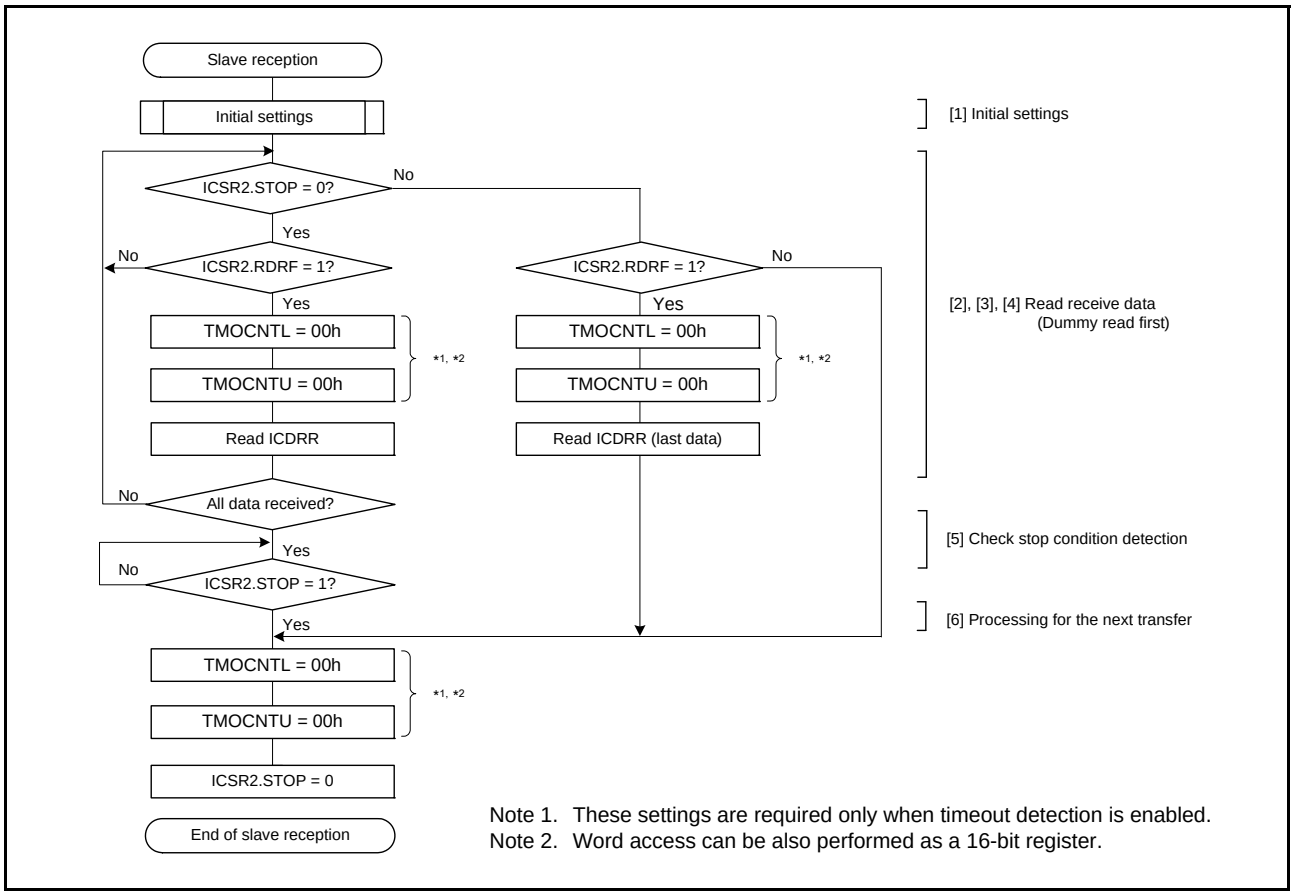


Figure 33.18 Example of Slave Reception Flowchart

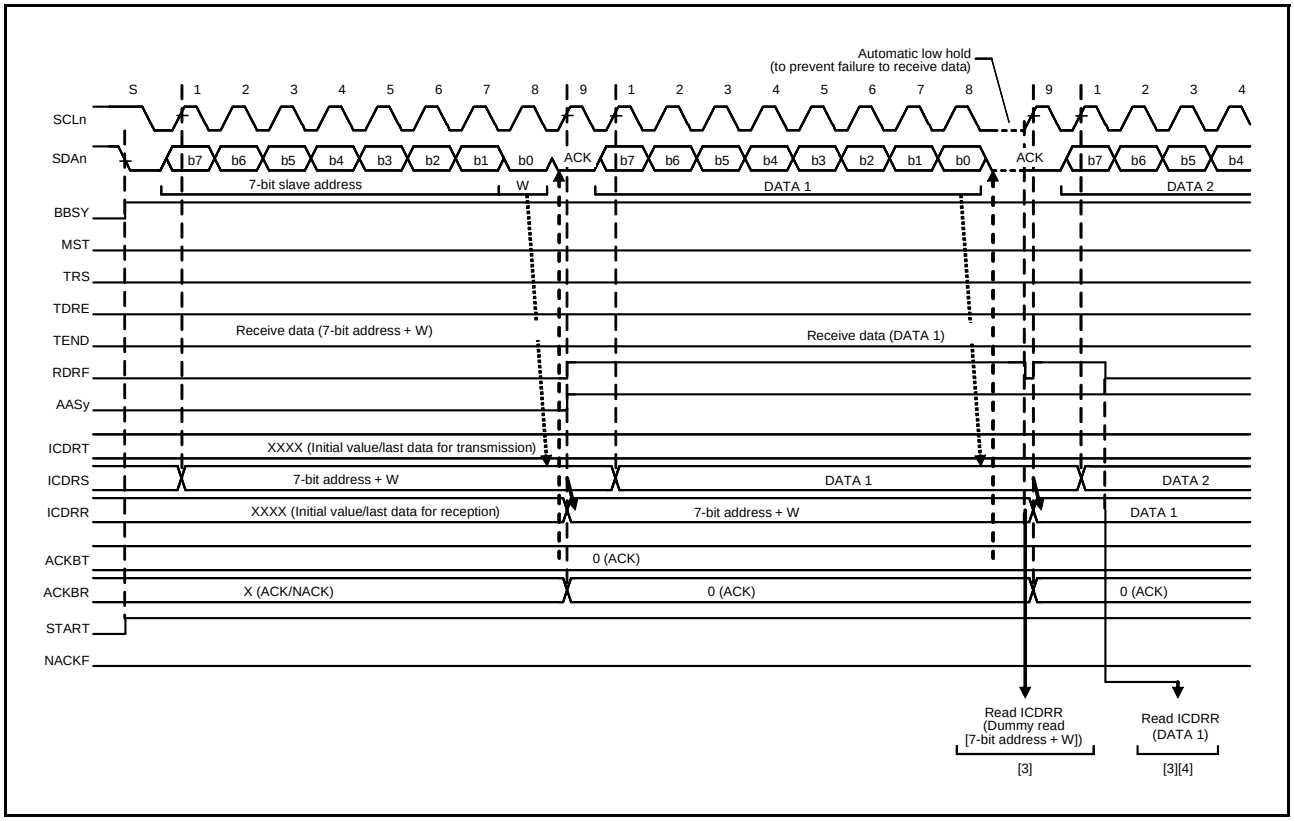


Figure 33.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

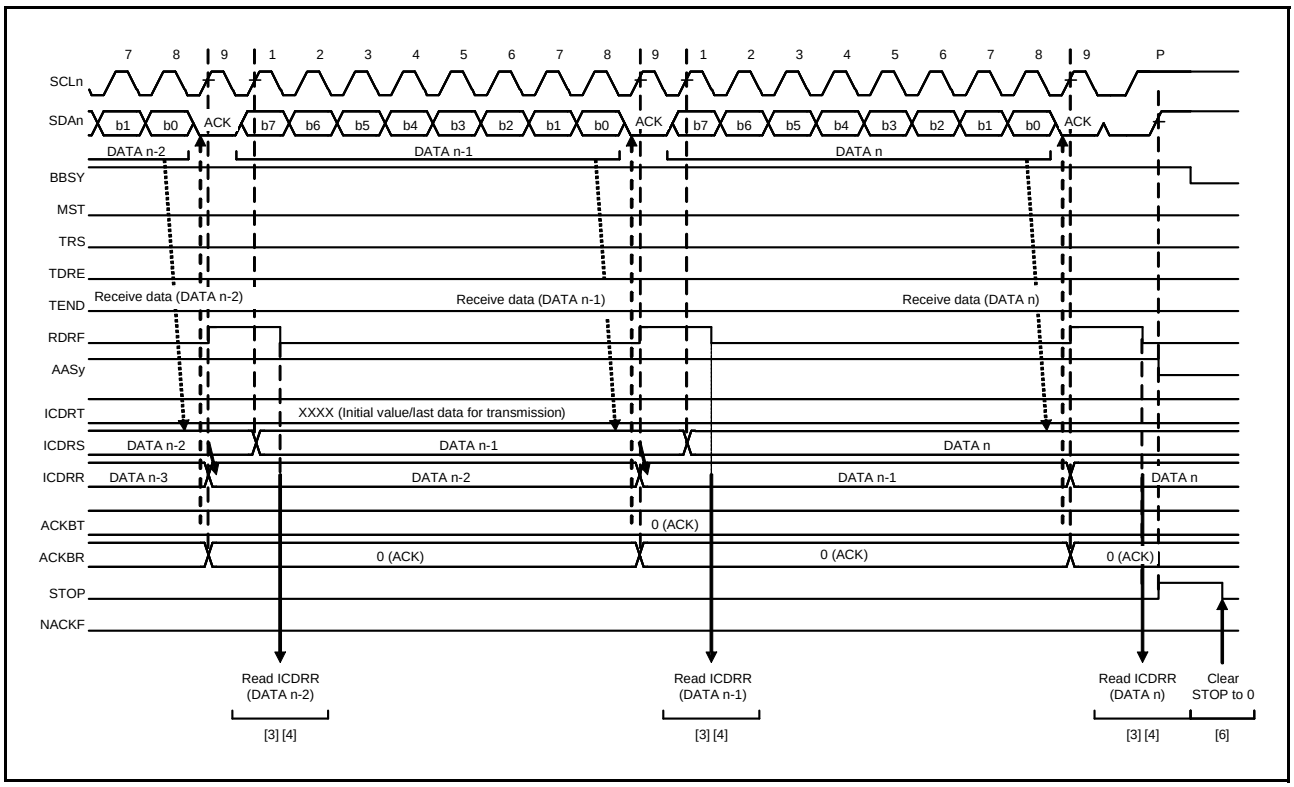


Figure 33.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

33.4 SCL Synchronization Circuit

In generation of the SCL clock, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCLn line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL clock 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 this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the RIIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in ICBRH, and the level on the SCLn 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 SCLn line low, and starts counting out the width at low level specified in ICBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCLn line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock 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 SCLn line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock 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 SCLE bit in ICFER is set to 1.

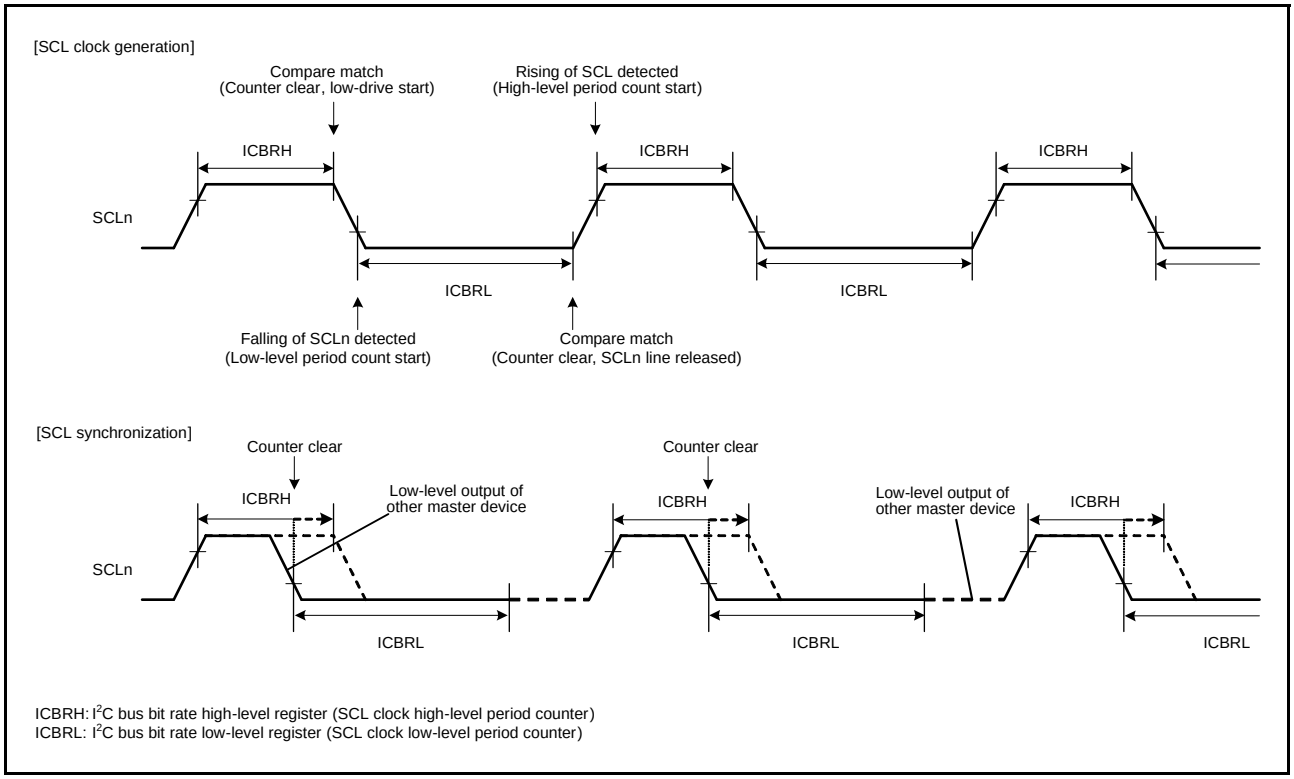


Figure 33.21 Generation and Synchronization of the SCL Signal from the RIIC

33.5 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 (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay function, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay function is enabled (i.e. while the SDDL[2:0] bits in ICMR2 are set to any value other than 000b), the DLCS bit in ICMR2 selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

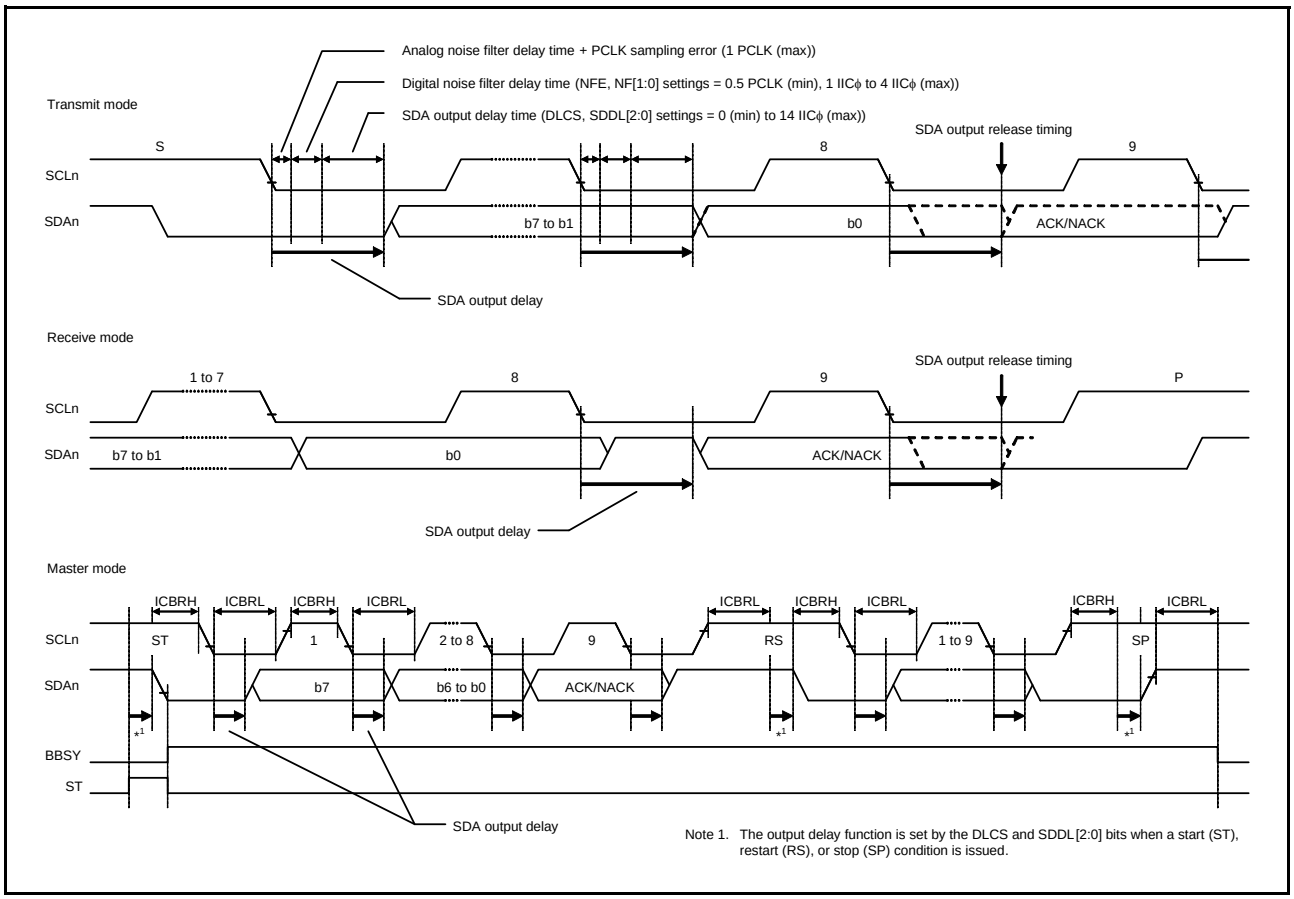


Figure 33.22 SDA Output Delay Function

33.6 Digital Noise Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 33.23 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the 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 NF[1:0] bits in ICMR3. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. 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 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. In such cases, it is possible to disable the digital noise-filter circuit (by setting the ICFER.NFE bit to 0) and use only the analog noise filter circuit.

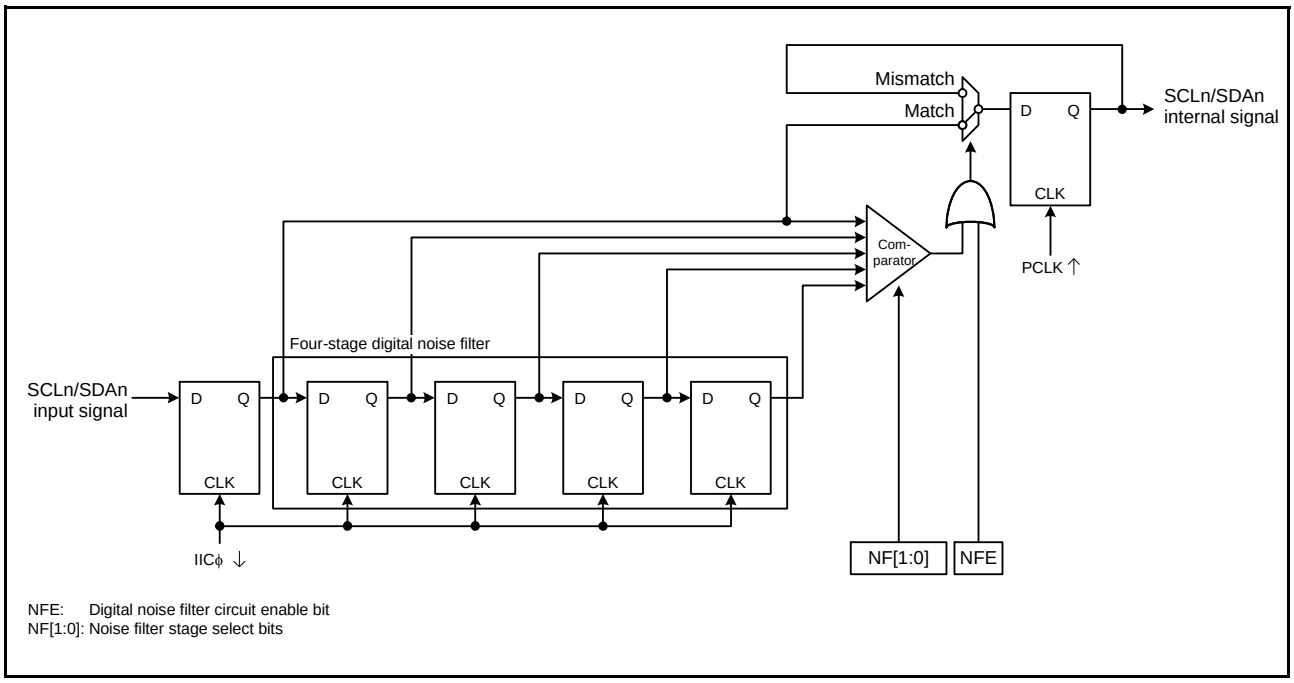


Figure 33.23 Block Diagram of Digital Noise Filter Circuit

33.7 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.

33.7.1 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 SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding AASy flag (y = 0 to 2) in ICSR1 is set to 1 at the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the following 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 33.24 to Figure 33.26 show the AASy flag set timing in three cases.

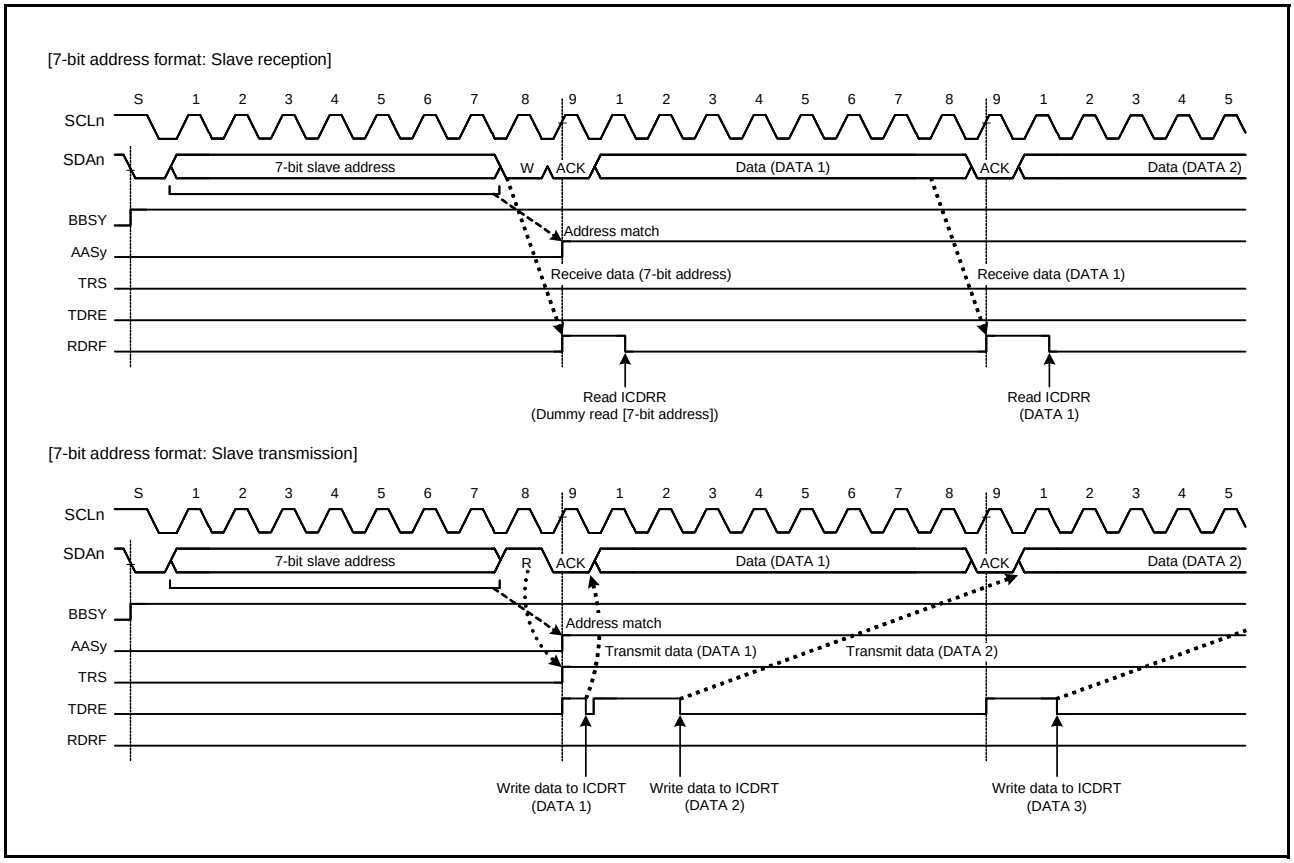


Figure 33.24 AASy Flag Set Timing with 7-Bit Address Format Selected

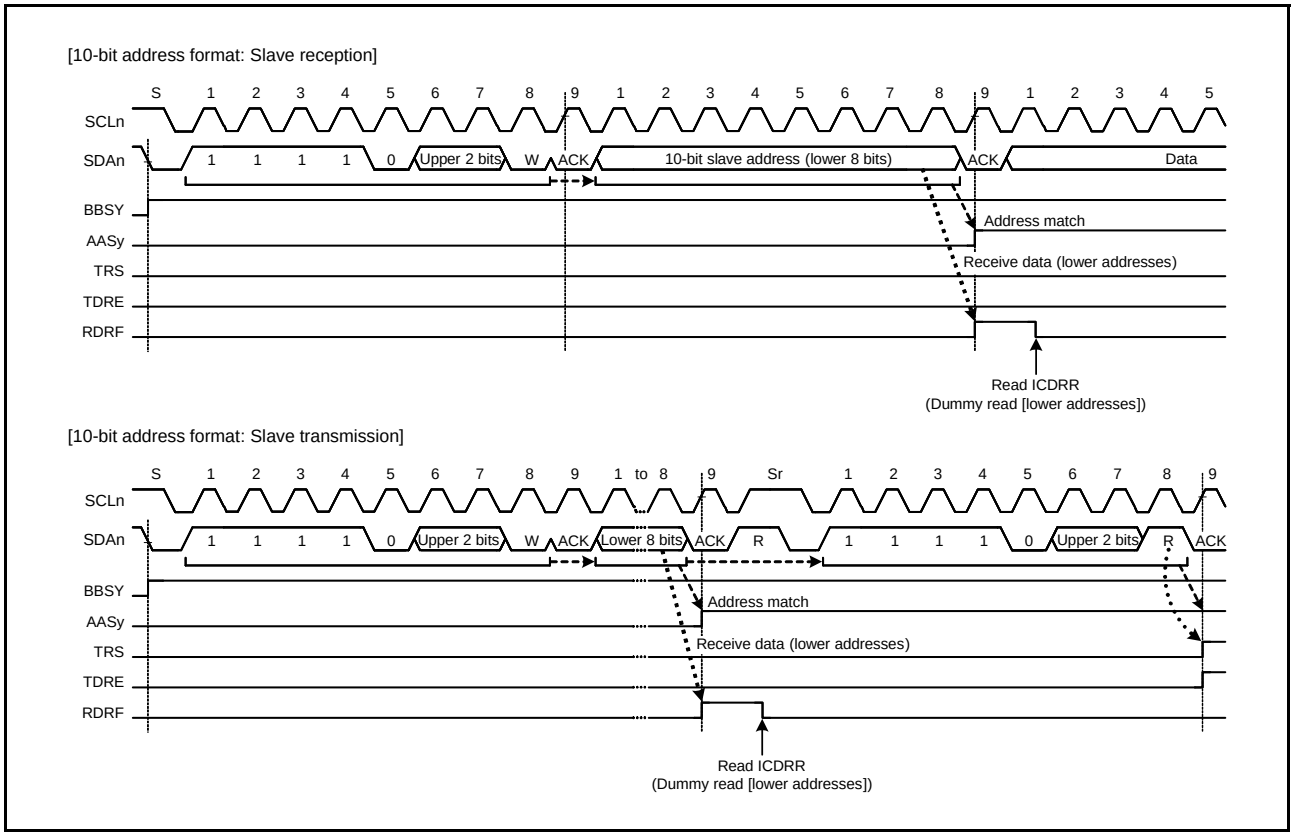


Figure 33.25 AASy Flag Set Timing with 10-Bit Address Format Selected

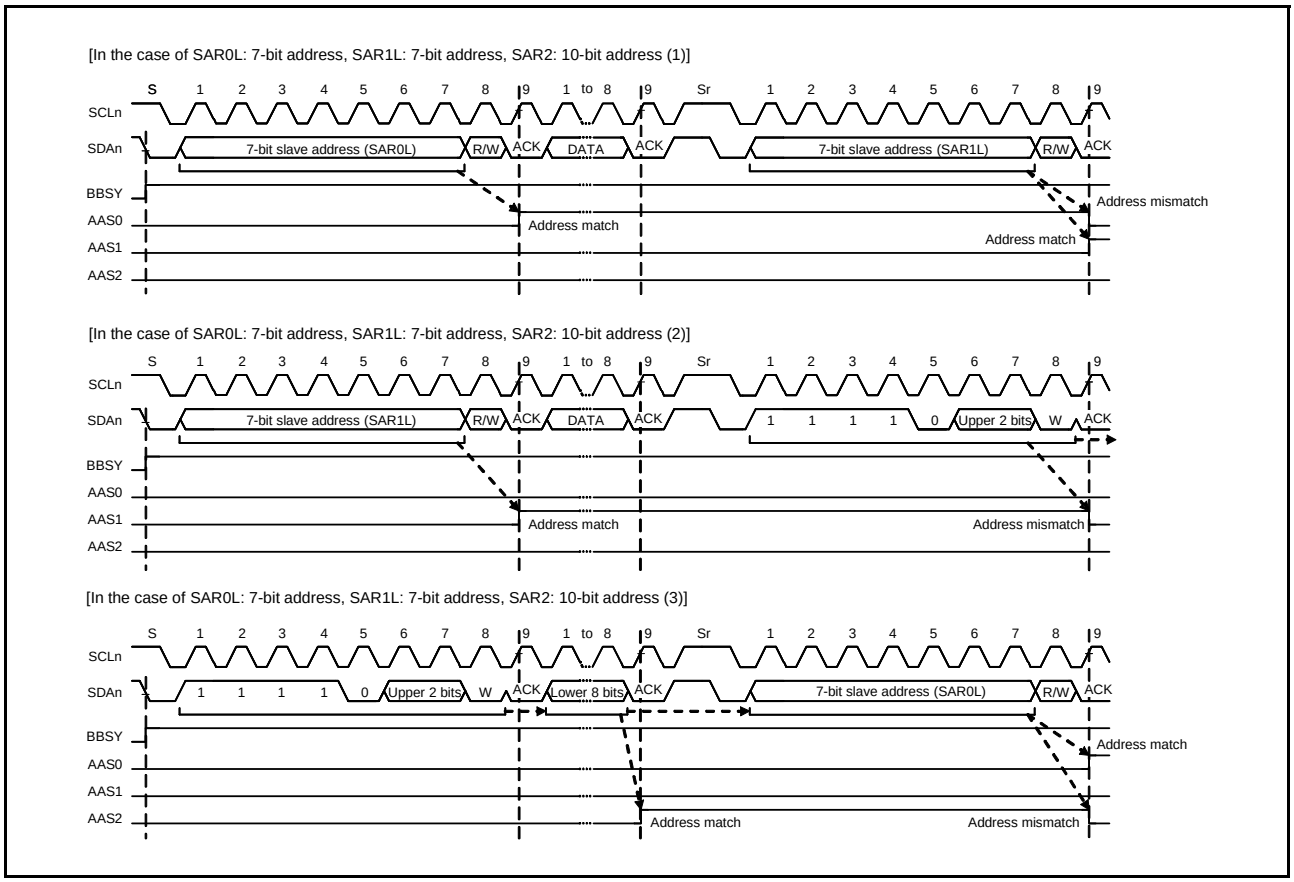


Figure 33.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

33.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (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 GCA flag in ICSR1 and the RDRF flag in ICSR2 are set to 1 on the rising edge of the ninth cycle of SCL clock. 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.

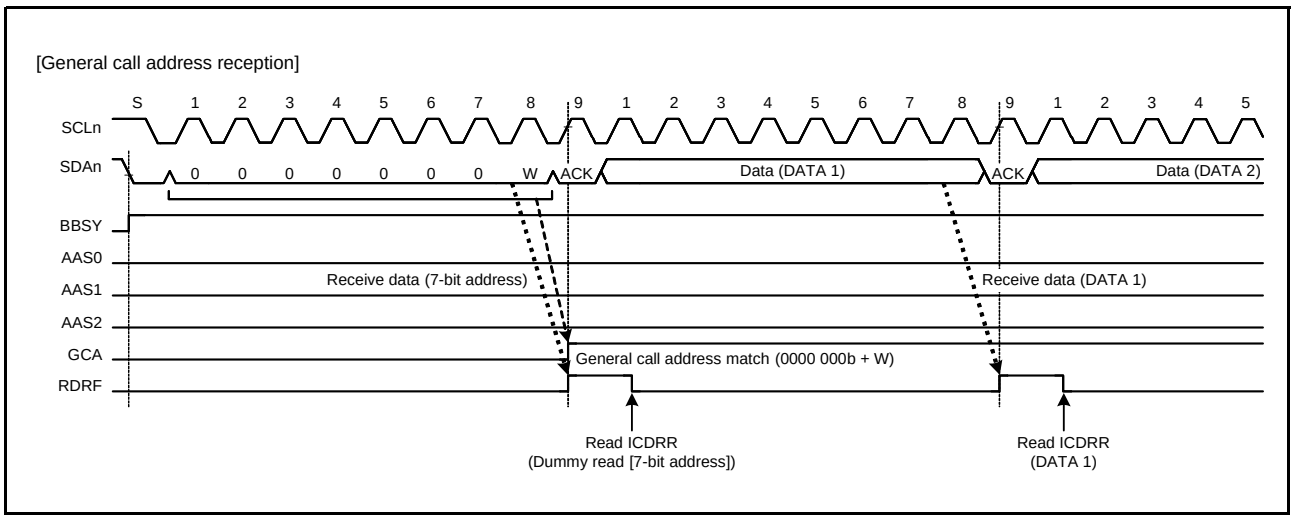


Figure 33.27 Timing of GCA Flag Setting during Reception of General Call Address

33.7.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I2C bus specification (Rev. 03). When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the DIDE bit in ICSER set to 1, the RIIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the eighth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding AASy flag (y = 0 to 2) in ICSR1 to 1.

After that, when the first byte received after a start or restart condition is issued 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 = 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.

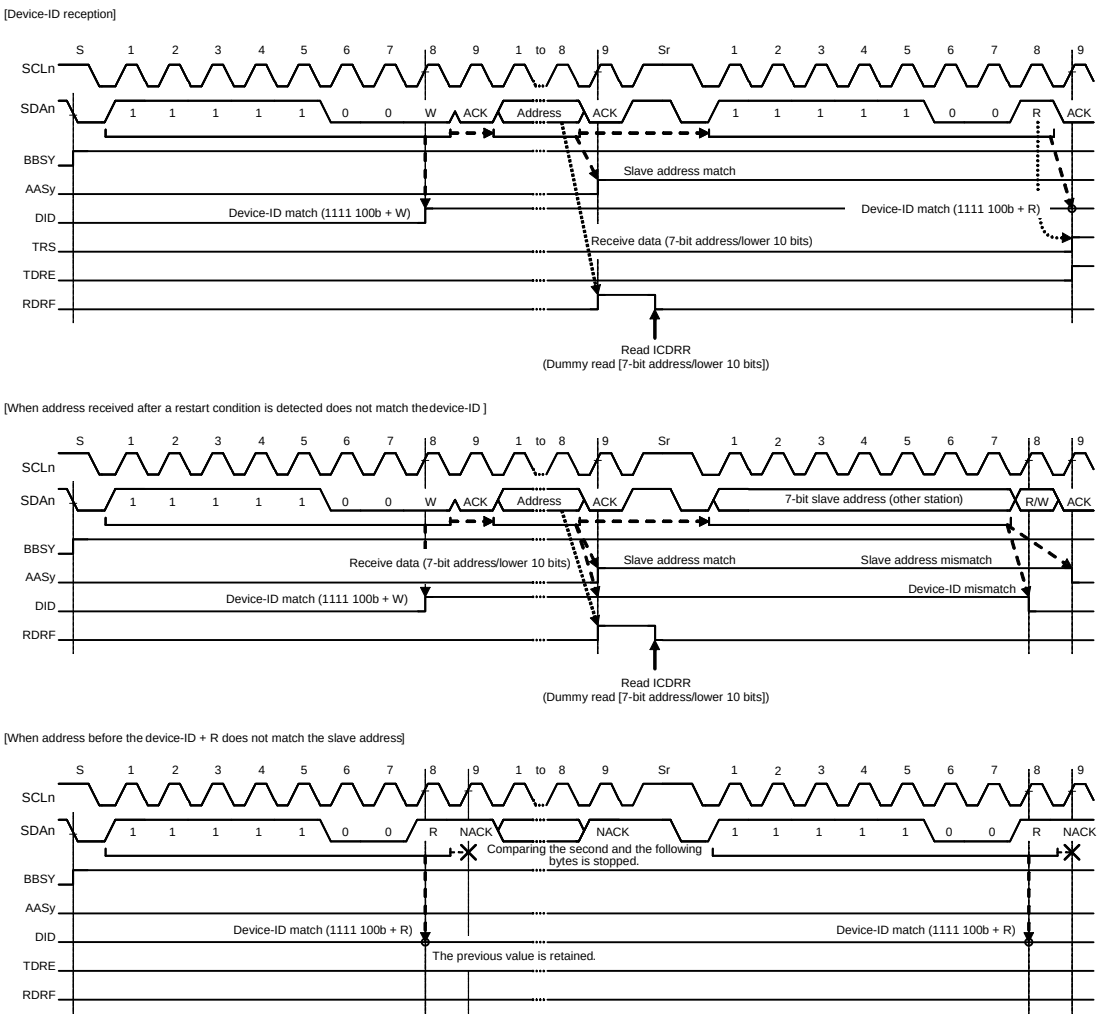


Figure 33.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

33.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the RIIC detects the host address, the HOA flag in ICSR1 is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the RDRF flag in ICSR2 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 = 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.

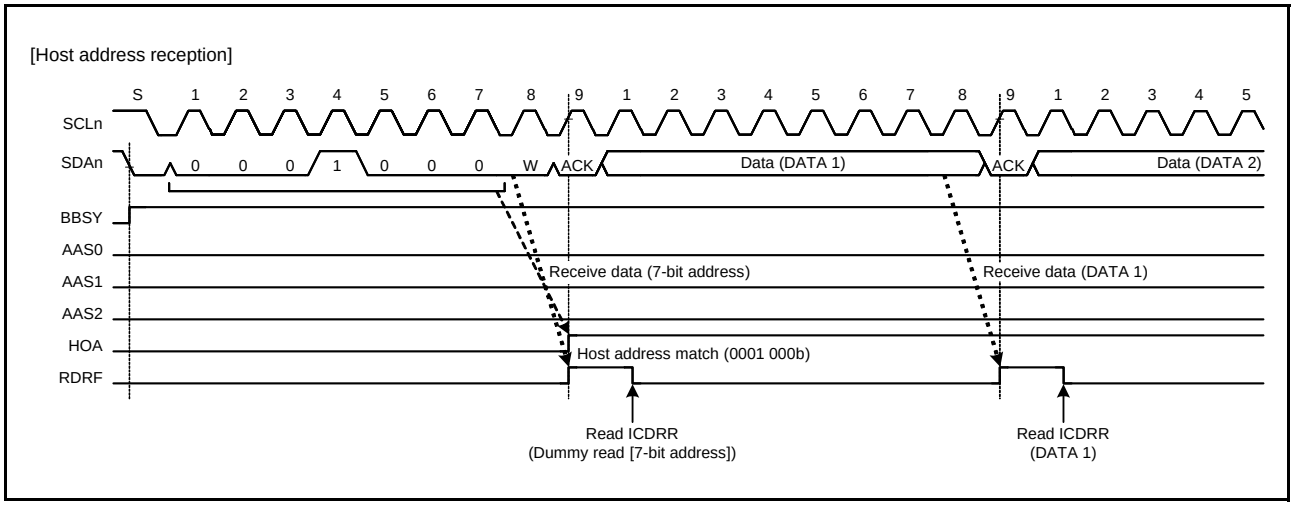


Figure 33.29 HOA Flag Set Timing during Reception of Host Address

33.8 Automatic Low-Hold Function for SCL

33.8.1 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 (TRS bit = 1 in ICCR2), the SCLn line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

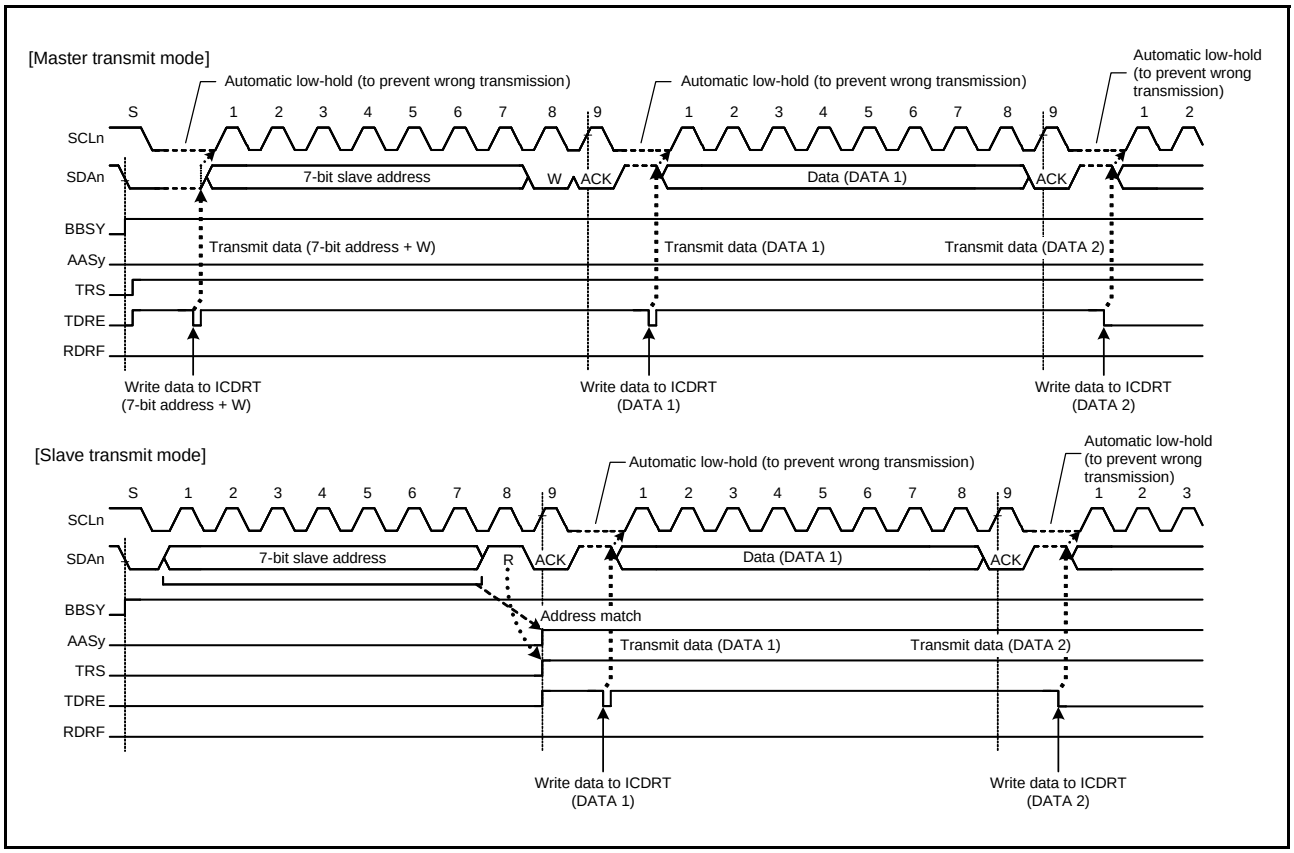


Figure 33.30 Automatic Low-Hold Operation in Transmit Mode

33.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data has already been written (TDRE flag = 0 in ICSR2) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, after setting the NACKF flag to 0, issue a restart condition, or issue a stop condition and then issue a start condition again.

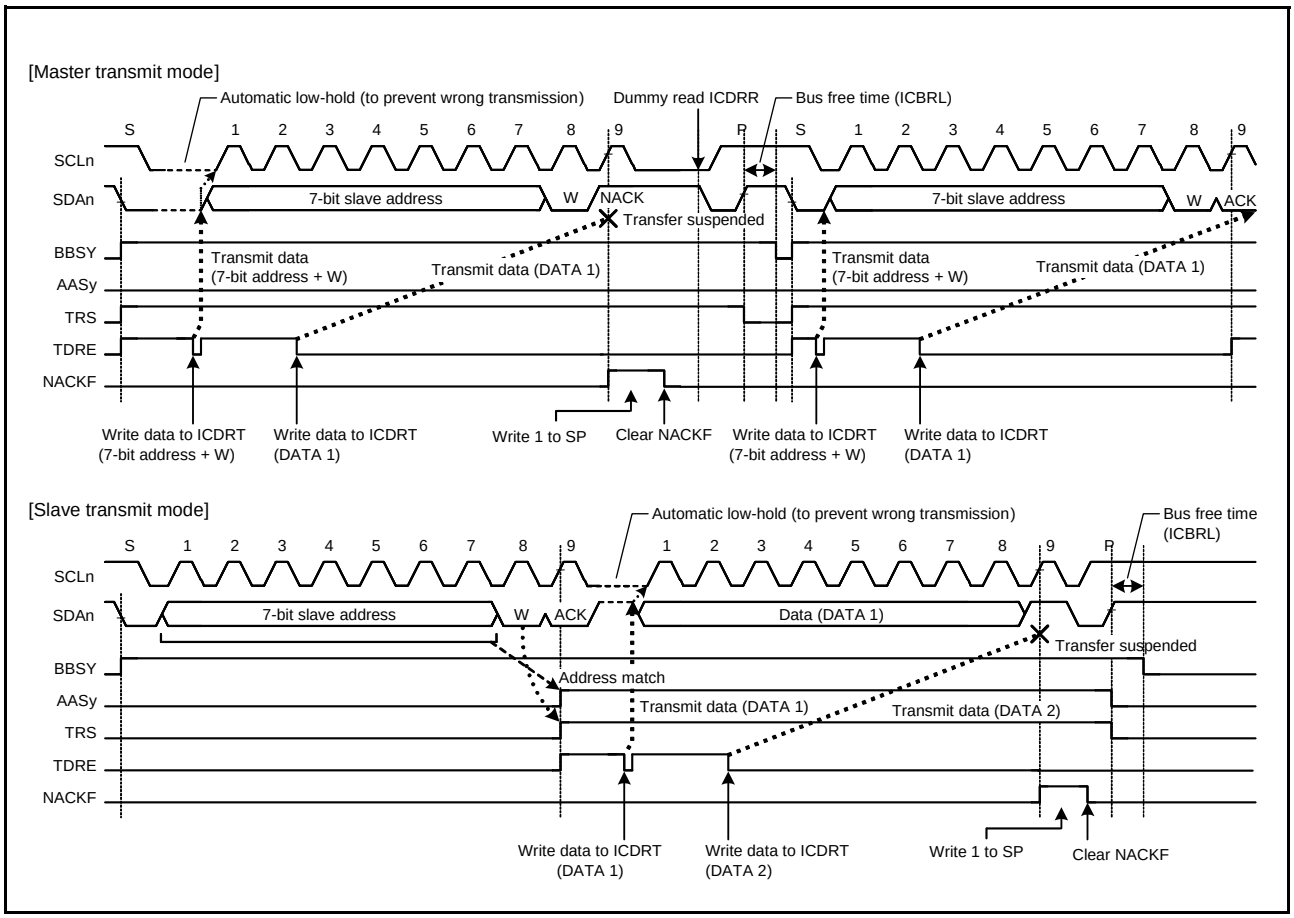


Figure 33.31 Suspension of Data Transfer When NACK is Received (NACKE = 1)

33.8.3 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 frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the RIIC holds the SCLn 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 issued. This function does not disturb other communication because the RIIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 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 ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables bytewise receive operation.
The WAIT bit function is enabled for receive frames 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 RDRFS bit in ICMR3 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 RDRF flag (receive data full) in ICSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, 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 frames 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.

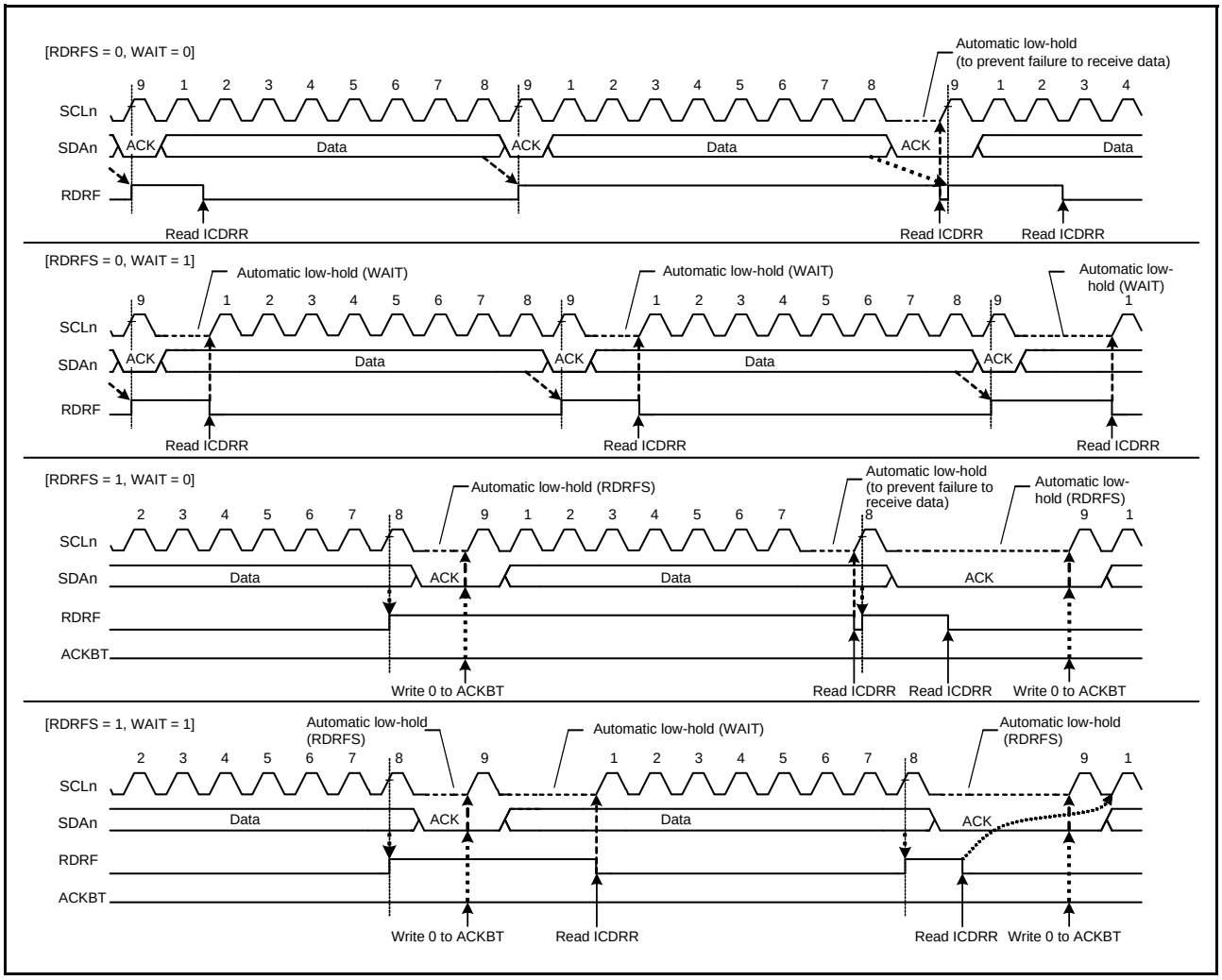


Figure 33.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

33.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

33.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDAn line low to issue a start condition. However, if the SDAn line has already been driven low by another master device issuing 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 issued in this case.
When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line, the RIIC loses in arbitration.
After a loss in arbitration of mastership, 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 loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).
[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2)

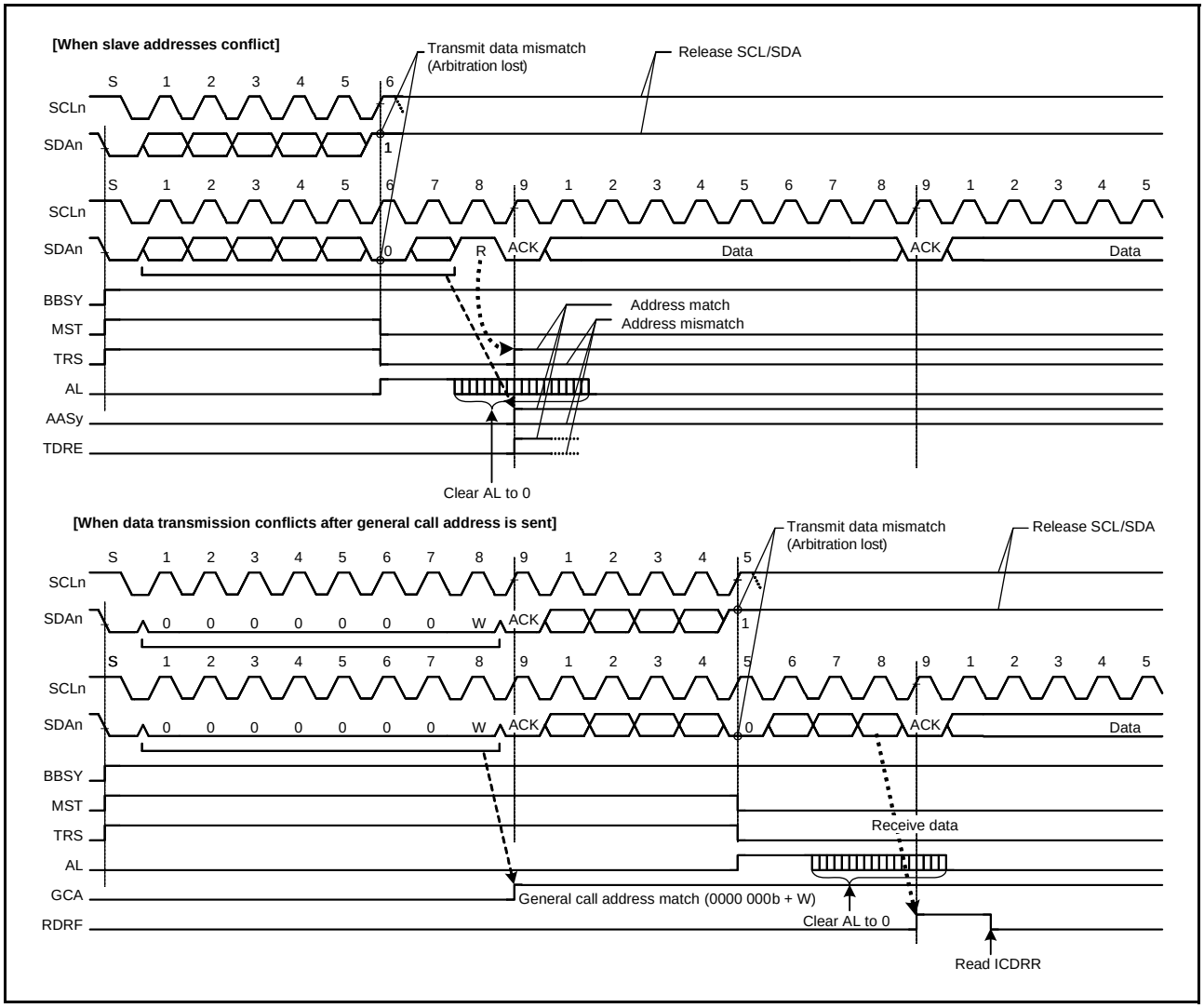


Figure 33.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

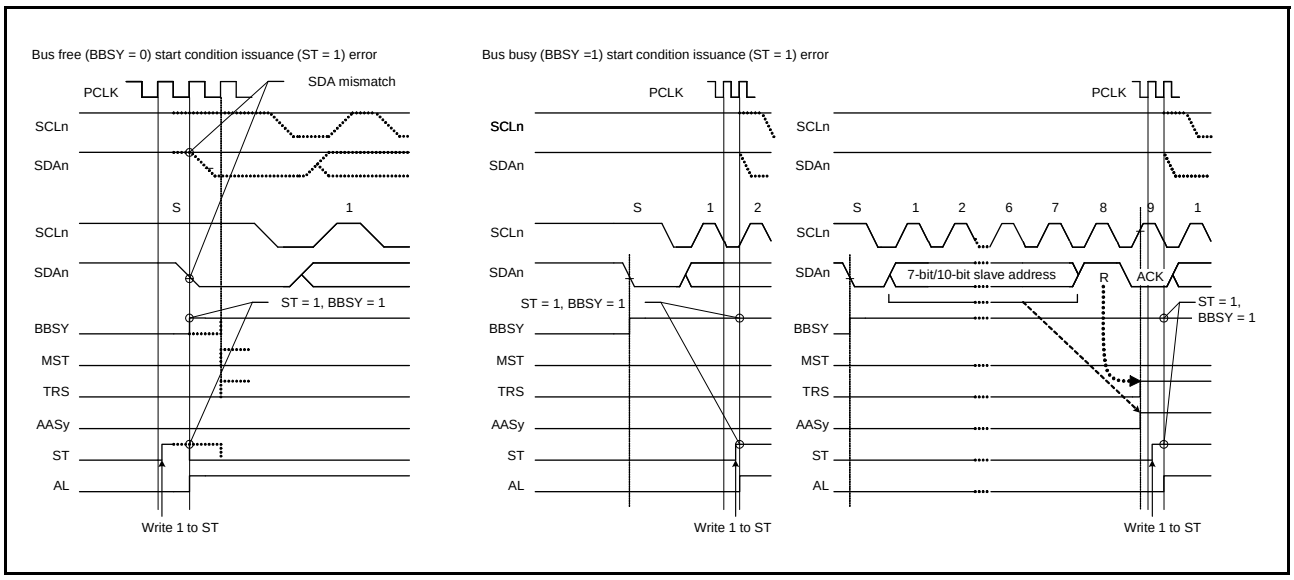


Figure 33.34 Arbitration-Lost When a Start Condition is Issued (MALE = 1)

33.9.2 Function to Detect Loss of Arbitration during NACK Transmission (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 SDAn line (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn 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 33.35 shows an example of arbitration-lost detection during transmission of NACK.

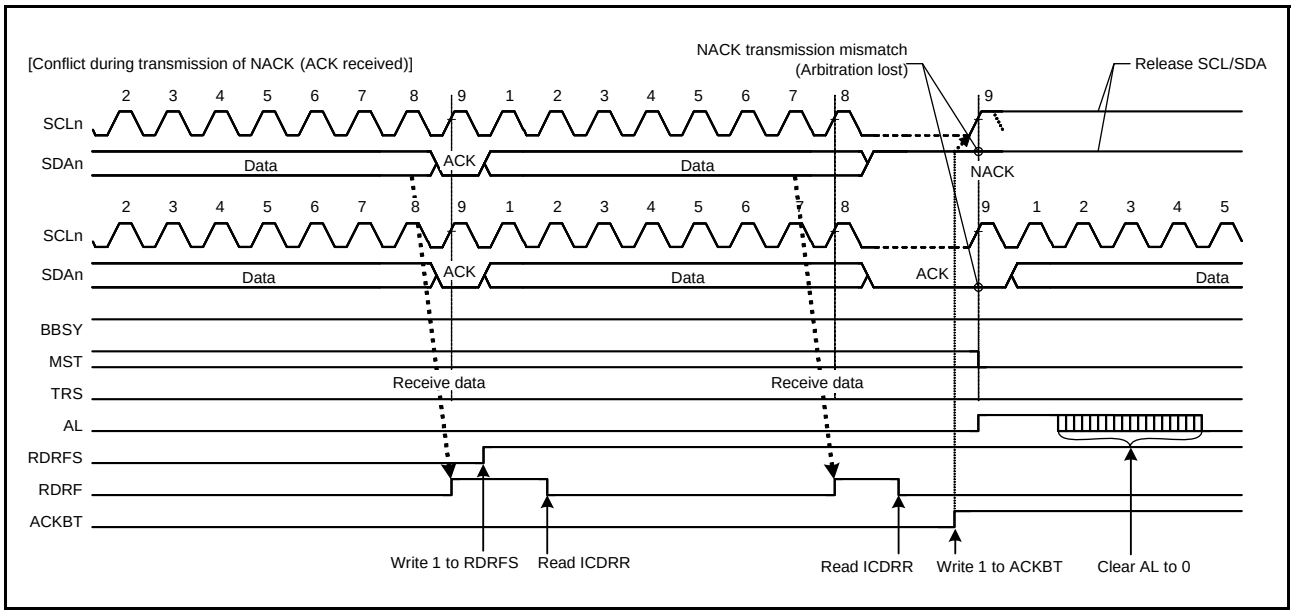


Figure 33.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

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 2 bytes of data from the slave device, and master B receives 4 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 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary 4 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 issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock 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 arbitration is lost during transmission of NACK, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

33.9.3 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 SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state and the low level is detected on the SDAn line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.
When it loses slave arbitration, the RIIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of FFh).
The RIIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

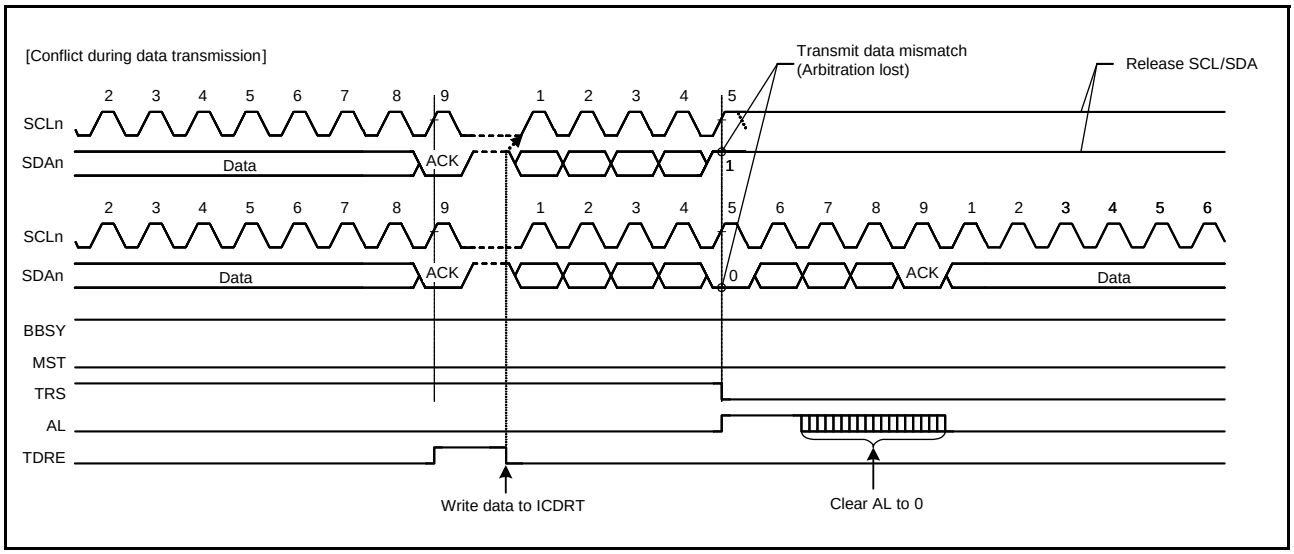


Figure 33.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

33.10 Start Condition/Restart Condition/Stop Condition Issuing Function

33.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ST bit in ICCR2 is set to 1.
When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.
A start condition is issued in the following sequence.

[Start condition issuance]
(1) Drive the SDAn line low (high level to low level).
(2) Ensure the time set in ICBRH and the start condition hold time.
(3) Drive the SCLn line low (high level to low level).
(4) Detect low level of the SCLn line and ensure the low-level period of SCLn line set in ICBRL.

33.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RS bit in ICCR2 is set to 1.
When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).
A restart condition is issued in the following sequence.

[Restart condition issuance]
(1) Release the SDAn line.
(2) Ensure the low-level period of SCLn line set in ICBRL.
(3) Release the SCLn line (low level to high level).
(4) Detect a high level of the SCLn line and ensure the time set in ICBRL and the restart condition setup time.
(5) Drive the SDAn line low (high level to low level).
(6) Ensure the time set in ICBRH and the restart condition hold time.
(7) Drive the SCLn line low (high level to low level).
(8) Detect a low level of the SCLn line and ensure the low-level period of SCLn line set in ICBRL.

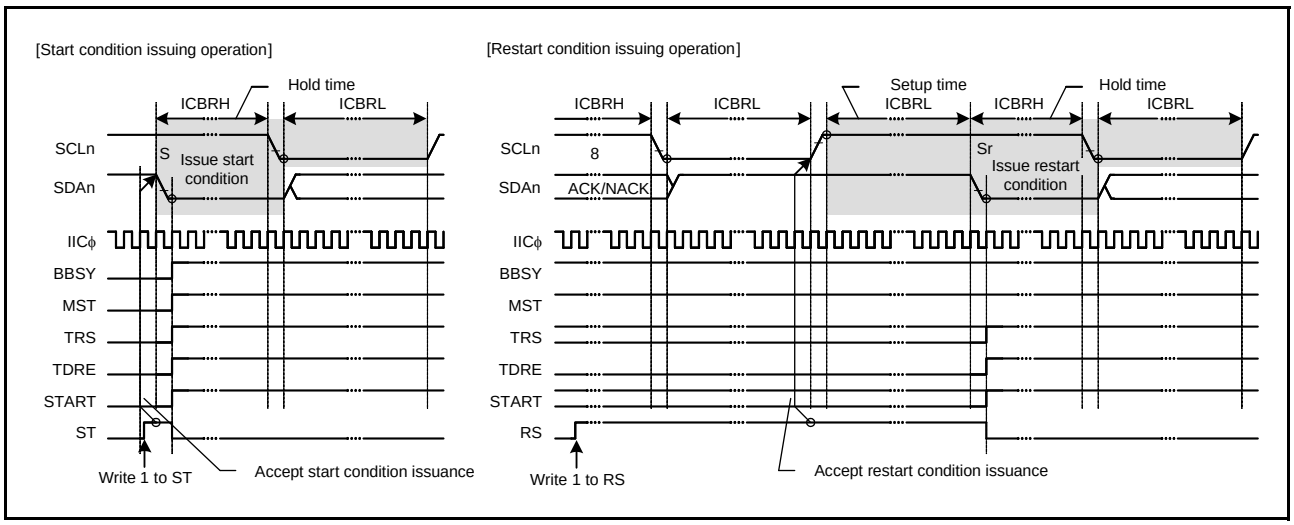


Figure 33.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

33.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the SP bit in ICCR2 is set to 1.
When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).
A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDAn line low (high level to low level).
- Ensure the low-level period of SCLn line set in ICBRL.
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDAn line (low level to high level).
- Ensure the time set in ICBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

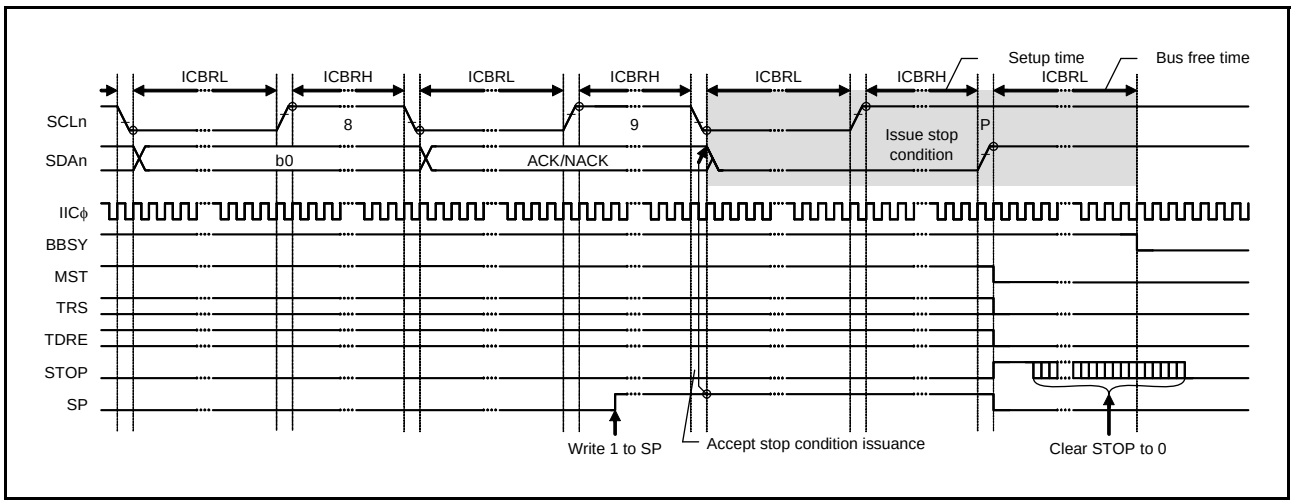


Figure 33.38 Stop Condition Issue Timing (SP Bit)

33.11 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 SCLn line and/or SDAn line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCLn line, a function for the output of an extra SCL clock cycle 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 the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCLn or SDAn lines.

33.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCLn line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.
The timeout function monitors the SCLn 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 SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn 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 SCLn 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 CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).
The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter does not work.

Note: When using the timeout detection function, refer to section 33.2.4, I2C Bus Mode Register 2 (ICMR2), section 33.2.18, Timeout Internal Counter (TMOCNT), and section 33.3.2, Initial Settings.

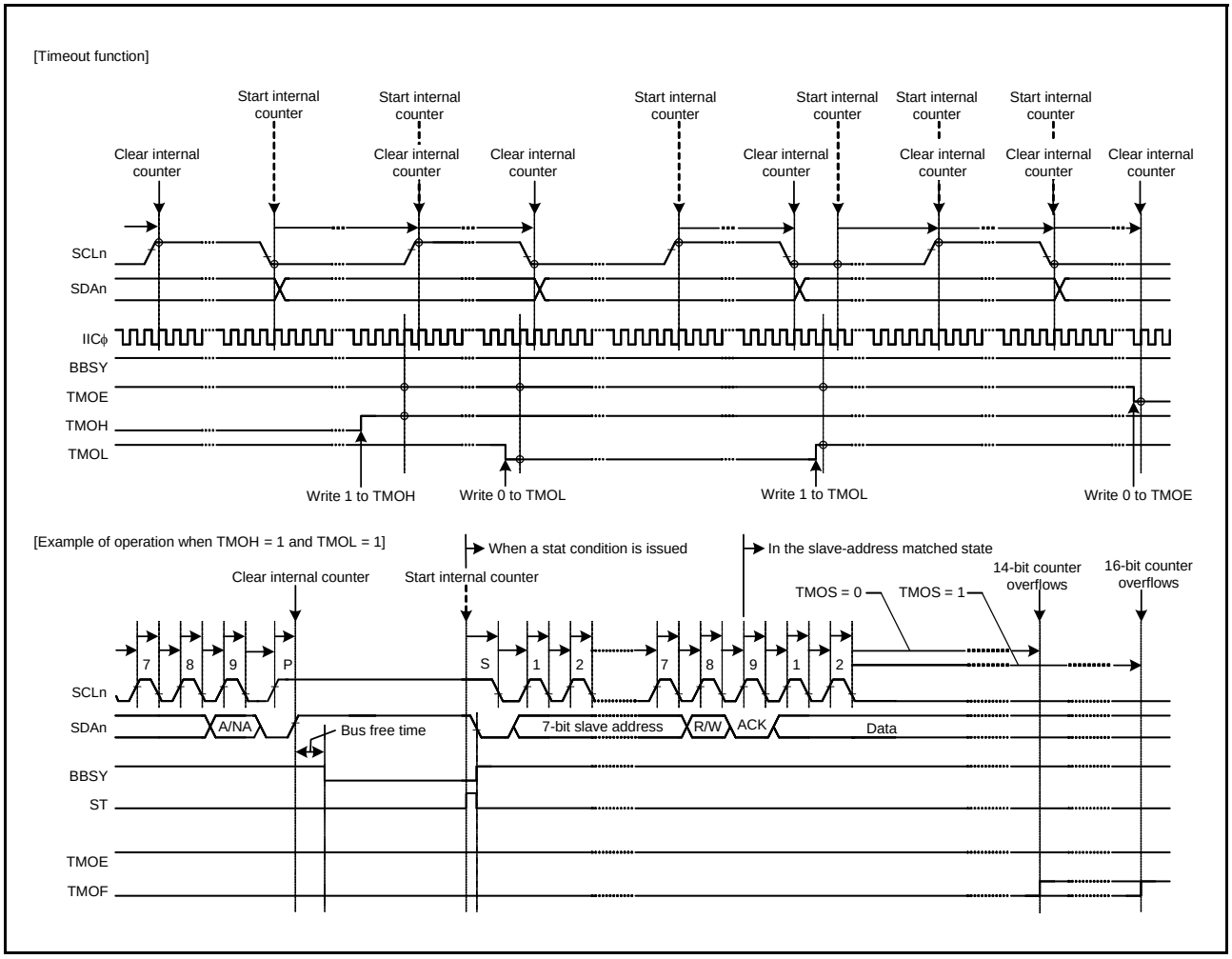


Figure 33.39 Timeout Function (ICMR1.CKS[2:0] = 000b)

33.11.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDAn line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL clock as the unit if the RIIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the CKS[2:0] bits in ICMR1, and of the ICBRH and ICBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.
When the RIIC module is in master mode and the slave device is holding the SDAn line at the low level 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 facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition.
Use this facility with the MALE bit (master arbitration-lost detection disabled) in ICFER set to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line, so take care on this point.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low

Figure 33.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

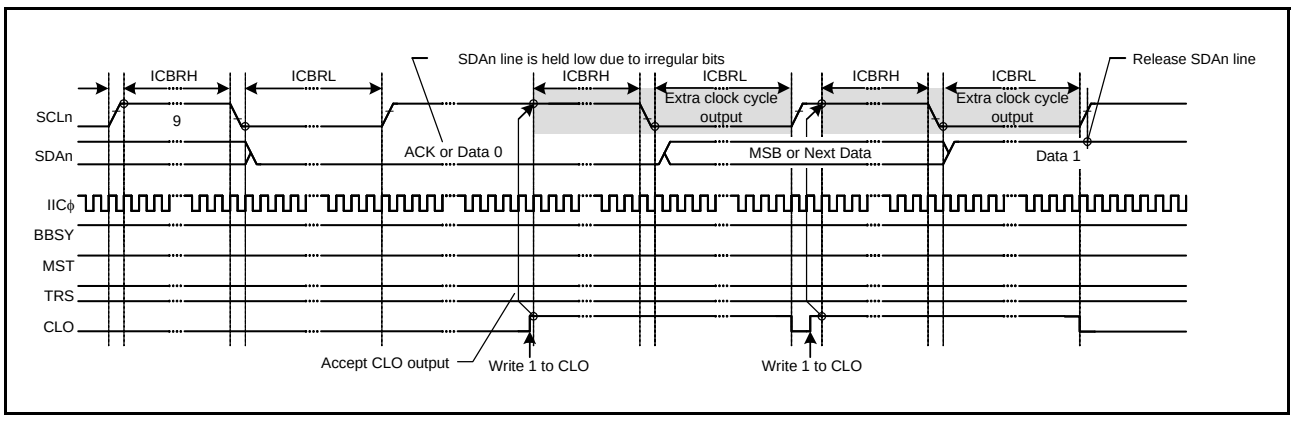


Figure 33.40 Extra SCL Clock Cycle Output Function (CLO Bit)

33.11.3 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 BBSY flag in ICCR2. 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 issuing a reset, be sure to set the IICRST bit in ICCR1 to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing 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 (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the RIIC and internal resets, refer to section 33.14, Resets and Register and Function States When Issuing Each Condition.

33.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, determine the values of the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL 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 (SARL0, SARL1, and SARL2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

33.12.1 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 total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (max.) of the SMBus standard.
If the time measured with the MTU or TMR exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCLn pin and SDAn pin and make the SCLn/SDAn pin outputs high-impedance, which releases 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 acknowledge bit
- Between acknowledge bits
- From acknowledge 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 transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (max.) of the SMBus standard, 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 clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the RDRFS bit in ICMR3 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 clock cycle.
If the period measured with the MTU or TMR exceeds the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDRT).

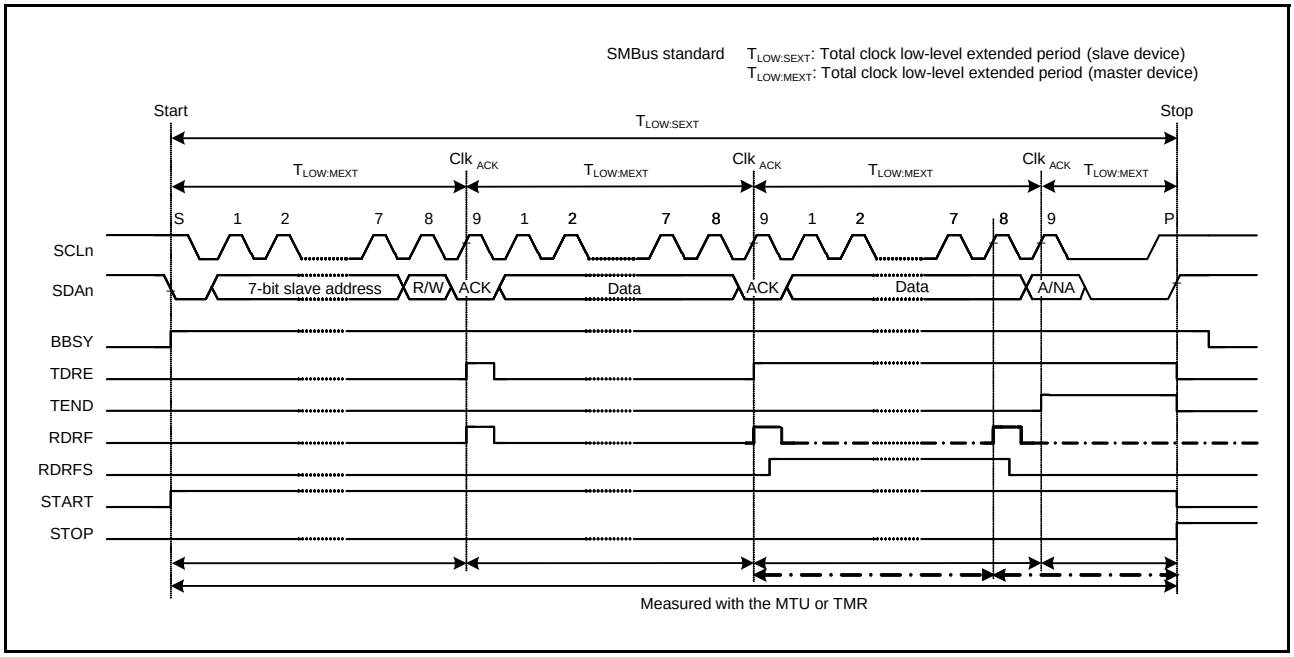


Figure 33.41 SMBus Timeout Measurement

33.12.2 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 35, CRC Calculator (CRC).

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 RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

33.12.3 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 SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address has been detected is the same as normal slave operation.

33.13 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 33.7 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of activating data transfer by the DTC or DMAC.

Table 33.7 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority	Interrupt Condition
EEI	Transfer error/ event generation	AL	Not possible	Not possible	High	AL = 1 • ALIE = 1
		NACKF				NACKF = 1 • NAKIE = 1
		TMOF				TMOF = 1 • TMOIE = 1
		START				START = 1 • STIE = 1
		STOP				STOP = 1 • SPIE = 1
RXI*2	Receive data full	RDRF	Possible	Possible		RDRF = 1 • RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	Possible		TDRE = 1 • TIE = 1
TEI*3	Transmit end	TEND	Not possible	Not possible	Low	TEND = 1 • 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 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. Since TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for TXI) is automatically set to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 2. Since RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for RXI) is automatically set to 0 when data are read from ICDRR.

Note 3. When using the TEI interrupt, clear the TEND flag in ICSR2 in the TEI interrupt handling.
Note that the TEND flag in ICSR2 is automatically set to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Clear or mask the each flag during interrupt handling.

33.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding 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 within the ICU is output when the value of the ICU.IRn.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 interrupt enable bit within the given peripheral module.

33.14 Resets and Register and Function States When Issuing Each Condition

The RIIC has reset, RIIC reset, and internal reset functions. Table 33.8 lists the resets and register and function states when issuing each condition.

Table 33.8 Resets and Register and Function States When Issuing Each Condition

		Chip Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	ICE, IICRST	At a reset	Retained	Retained	Retained	Retained
	SCLO, SDAO		At a reset	At a reset		
	Others			Retained		
ICCR2	BBSY	At a reset	At a reset	Retained	Retained	Retained
	ST			At a reset	At a reset	Retained
	Others					At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Retained
	Others			Retained	Retained	
ICMR2		At a reset	At a reset	Retained	Retained	Retained
ICMR3		At a reset	At a reset	Retained	Retained	Retained
ICFER		At a reset	At a reset	Retained	Retained	Retained
ICSER		At a reset	At a reset	Retained	Retained	Retained
ICIER		At a reset	At a reset	Retained	Retained	Retained
ICSR1		At a reset	At a reset	At a reset	Retained	At a reset
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Retained	At a reset
	START				Retained	
	STOP				Retained	Retained
	Others				Retained	Retained
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		At a reset	At a reset	Retained	Retained	Retained
ICBRH, ICBRL		At a reset	At a reset	Retained	Retained	Retained
ICDRT		At a reset	At a reset	Retained	Retained	Retained
ICDRR		At a reset	At a reset	Retained	Retained	Retained
ICDRS		At a reset	At a reset	At a reset	Retained	Retained
Timeout function		At a reset	At a reset	Operation	Operation	Operation
Bus free time measurement		At a reset	At a reset	Operation	Operation	Operation

33.15 Event Link Output

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Transfer error event

When a transfer error event occurs, the corresponding event signal can be output for another module via the ELC.

(2) Receive data full

When a receive data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

33.15.1 Interrupt Handling and Event Linking

The RIIC module produces four kinds of interrupt: transfer error (arbitration-lost detection, detection of NACK, detection of timeout, or detection of a stop condition) event, receive data full, transmit data empty, and transmit end interrupts detection of a start condition. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals 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 33.7.

33.16 Usage Notes

33.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC). 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 registers B and C, refer to section 11, Low Power Consumption.

33.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 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.

34. Serial Peripheral Interface (RSPI)

In this section, “PCLK” is used to refer to PCLKB.

34.1 Overview

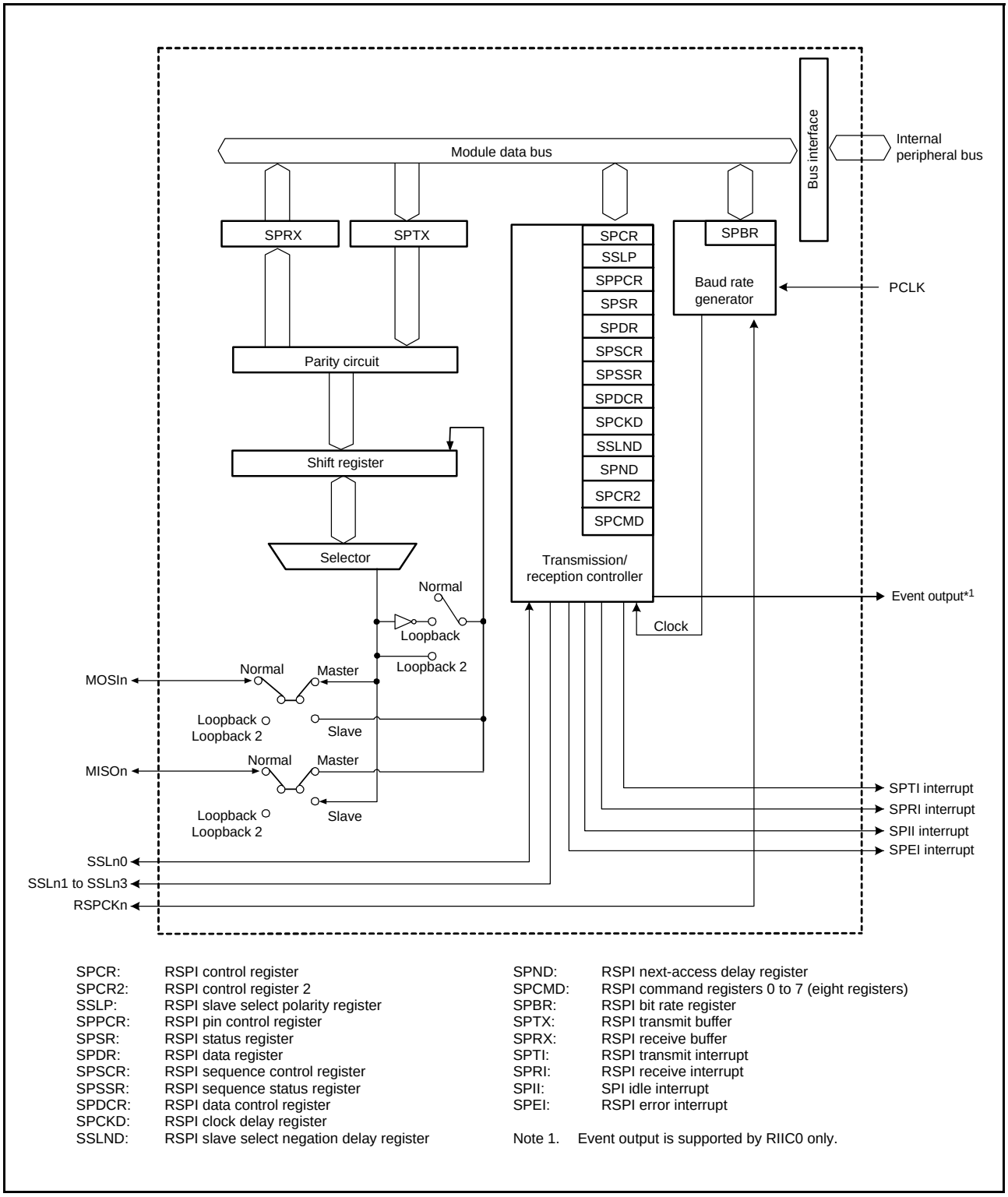
This MCU includes two channels of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 34.1 lists the specifications of the RSPI, and Figure 34.1 shows a block diagram of the RSPI.
In this section, n indicates A or B, and i indicates 0 to 3. Also, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

Table 34.1 RSPI Specifications (1/2)

Item	Description
Number of channels	Two channels
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). • Transmit-only operation is available. • Communication mode: Full-duplex or transmit-only 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).
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 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8). Width at high level: 4 cycles of PCLK; width at low level: 4 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 • Parity error detection
SSL control function	• Four SSL pins (SSLn0 to SSLn3) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output. • In multi-master mode: SSLn0 pin for input, and SSLn1 to SSLn3 pins for either output or unused. • In slave mode: SSLn0 pin for input, and SSLn1 to SSLn3 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

Table 34.1 RSPI Specifications (2/2)

Item	Description
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt RSPI error interrupt (mode fault, overrun, parity error) RSPI idle interrupt (RSPI idle)
Event link function (output)	• The following events can be output to the event link controller. (RSPI0) Receive buffer full signal Transmit buffer empty signal Mode fault, overrun, or parity error signal RSPI idle signal Transmission-completed signal
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.



SPCR:	RSPI control register	SPND:	RSPI next-access delay register
SPCR2:	RSPI control register 2	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SSLP:	RSPI slave select polarity register	SPBR:	RSPI bit rate register
SPPCR:	RSPI pin control register	SPTX:	RSPI transmit buffer
SPSR:	RSPI status register	SPRX:	RSPI receive buffer
SPDR:	RSPI data register	SPTI:	RSPI transmit interrupt
SPSCR:	RSPI sequence control register	SPRI:	RSPI receive interrupt
SPSSR:	RSPI sequence status register	SPII:	SPI idle interrupt
SPDCR:	RSPI data control register	SPEI:	RSPI error interrupt
SPCKD:	RSPI clock delay register		
SSLND:	RSPI slave select negation delay register		

Note 1. Event output is supported by RIIC0 only.

Figure 34.1 RSPI Block Diagram

Table 34.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 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 RSPCKn, MOSIn, and MISOn are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLn0 pin.
Refer to section 34.3.2, Controlling RSPI Pins for details.

Table 34.2 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
RSPI1	RSPCKB	I/O	Clock I/O
	MOSIB	I/O	Master transmit data I/O
	MISOB	I/O	Slave transmit data I/O
	SSLB0	I/O	Slave selection I/O
	SSLB1	Output	Slave selection output
	SSLB2	Output	Slave selection output
	SSLB3	Output	Slave selection output

34.2 Register Descriptions

34.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h, RSPI1.SPCR 0008 83A0h

	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	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	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	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI 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	RSPI Receive Buffer Full Interrupt Enable	0: Disables the generation of RSPI receive buffer full interrupt requests 1: Enables the generation of RSPI receive buffer full interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The RSPCKn, MOSIn, and MISOn 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). Operation should not be performed if the CPHA bit is set 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 synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 34.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 34.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLn0 to SSLn3 pins based on combinations of the MODFEN and MSTR bits (refer to section 34.3.2, 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 RSPCKn, MOSIn, MISOn, and SSLn0 to SSLn3.

SPEIE Bit (RSPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault 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 34.3.8, 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 RSRI 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 34.3.8, 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 34.3.9, 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 (RSPI 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 an RSPI receive buffer full interrupt request.

34.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h, RSPI1.SSLP 0008 83A1h

	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

If the contents of SSLP are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

34.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h, RSPI1.SPPCR 0008 83A2h

	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 MOSIn pin during MOSI idling corresponds to low 1: The level output on the MOSIn 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

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

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 MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn 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 MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn 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 MOSIn 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 MOSIn 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 MOSIn pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIn pin.

34.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h, RSPI1.SPSR 0008 83A3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PERF	MODF	IDLNF	OVRF
Value after reset:	x	0	x	0	0	0	0	0

x: Undefined

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	RSPI Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: No mode fault error occurs 1: A mode fault error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	—	Reserved	The read value is undefined. The write value should be 1.	R/W

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

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (RSPI is initialized)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (RSPI is initialized)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error.

[Setting condition]

Multi-master mode

- When the input level of the SSLni 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 SSLni 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

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR 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 serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

34.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h, RSPI1.SPDR 0008 83A4h

	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

Address(es): RSPI0.SPDR 0008 8384h, RSPI1.SPDR 0008 83A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
									H							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR 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), access SPDR.
When accessing in words (the SPLW bit is 0), access the higher-order 16 bits (H) of SPDR.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 34.2 shows the Configuration of SPDR.

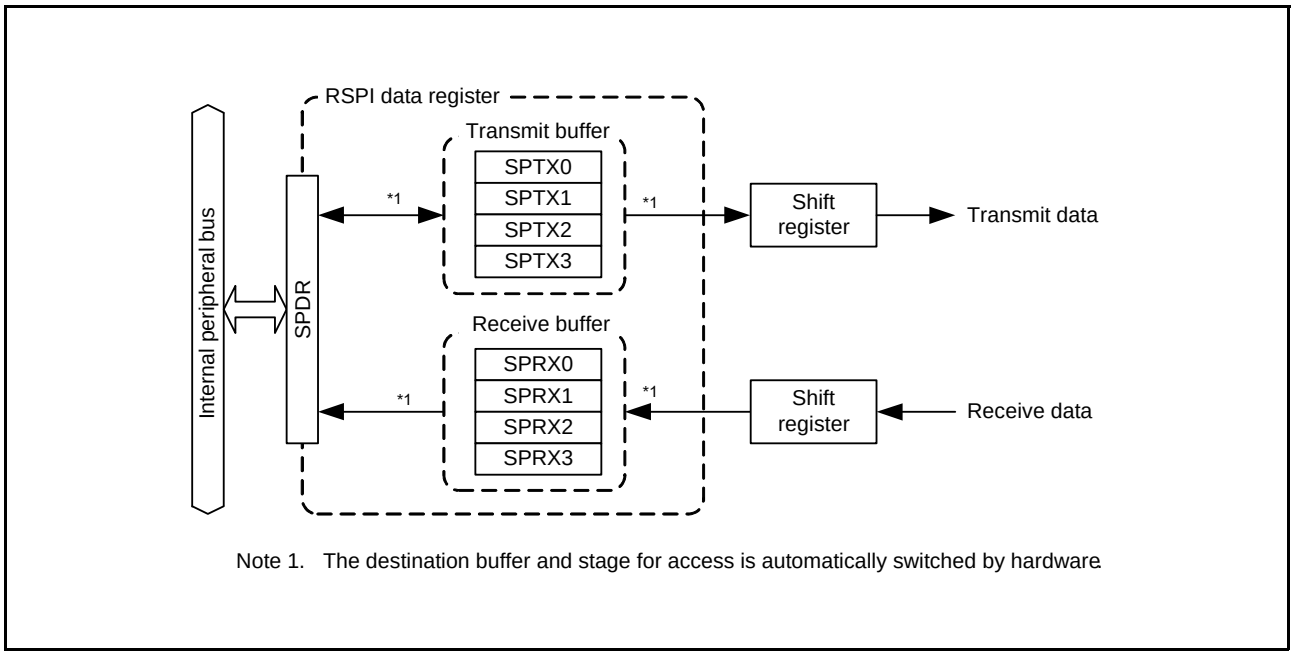


Figure 34.2 Configuration of SPDR

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 SPDR.
Data written to SPDR 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

SPDR 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 SPDR. Furthermore, the unit of access for SPDR is selected by the RSPI longword access/word access specification bit in the RSPI data control register (SPDCR.SPLW). 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 SPDR are described below.

(a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 34.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

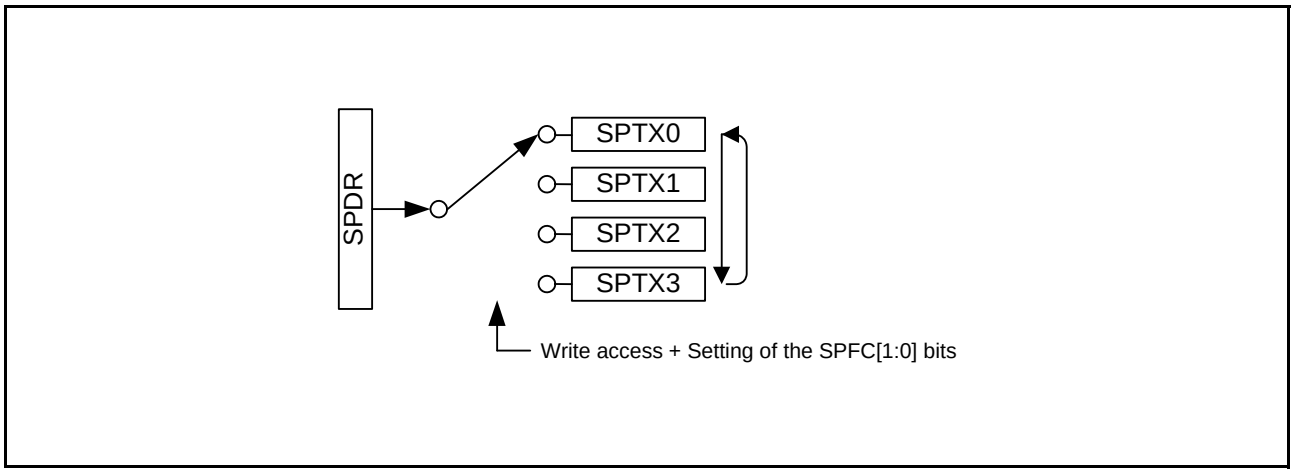


Figure 34.3 Configuration of SPDR (Writing)

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, 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.

(b) Reading

SPDR 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 34.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

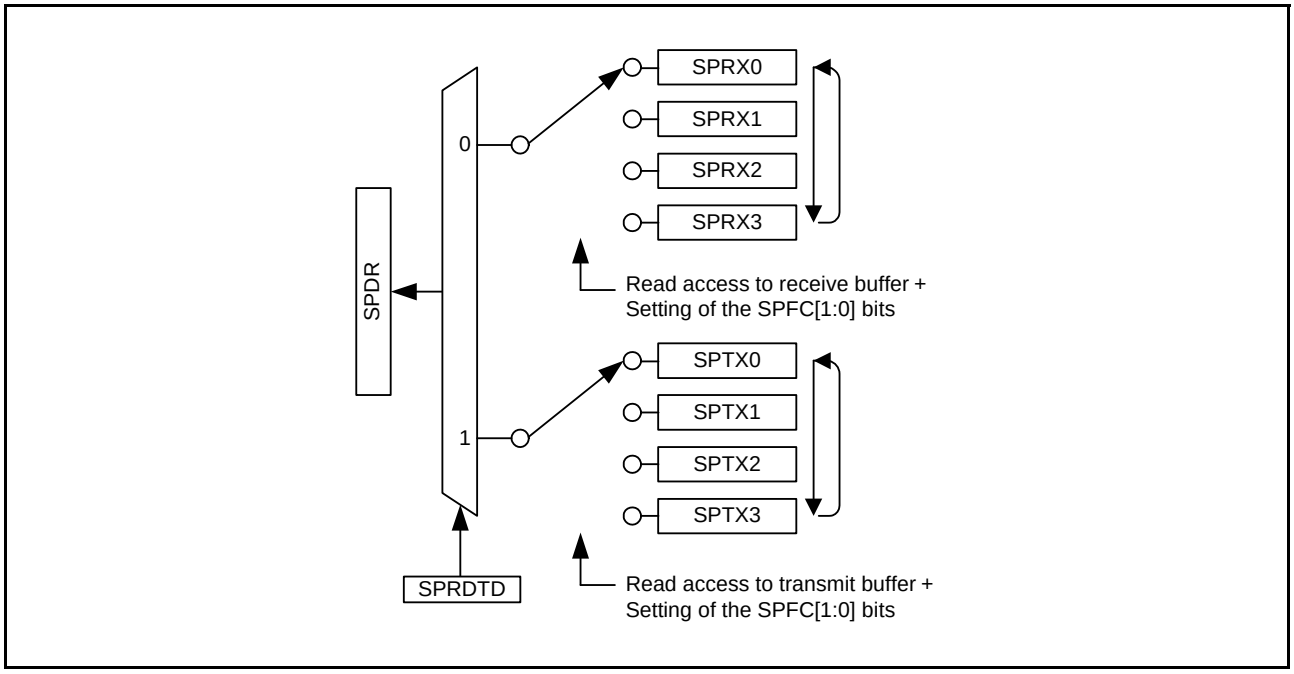


Figure 34.4 Configuration of SPDR (Reading)

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. However, when 1 is written to the RSRI 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 SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR 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.

34.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h, RSPI1.SPSCR 0008 83A8h

	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 (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 SPCMD07 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 RSPI is shown above. However, the RSPI in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR 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 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, SPCMD0 is referred.

34.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h, RSPI1.SPSSR 0008 83A9h

	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

SPSSR indicates the sequence control status when the RSPI operates in master mode.
Any writing to SPSSR is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMDm that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, refer to section 34.3.10.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm 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 34.3.8, Error Detection. For the RSPI's sequence control, refer to section 34.3.10.1, Master Mode Operation.

34.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah, RSPI1.SPBR 0008 83AAh

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

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

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 SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1) \times 2^{N}}$$

Table 34.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 34.3 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
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps

34.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh, RSPI1.SPDCR 0008 83ABh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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: SPDR values are read from the receive buffer 1: 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	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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 SPDR (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 RSPI receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts. Table 34.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations should not be performed.

Table 34.4 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 Receive Buffer Full Interrupt Occurs or Transmit Buffer Holding Data is Recognized
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 reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to 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.
For details, refer to section 34.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR is 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. When 20, 24, or 32 bits is specified, operations should not be performed.

34.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch, RSPI1.SPCKD 0008 83ACh

	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

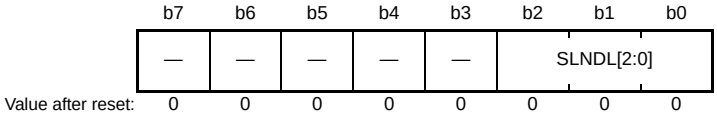
SPCKD sets a period from the beginning of SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

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.

34.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh, RSPI1.SSLND 0008 83ADh



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

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

34.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh, RSPI1.SPND 0008 83AEh

	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 SSLni signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

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.

34.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh, RSPI1.SPCR2 0008 83AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	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	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 (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI 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
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

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 (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI 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.

34.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh, RSPI1.SPCMD0 0008 83B0h, RSPI1.SPCMD1 0008 83B2h, RSPI1.SPCMD2 0008 83B4h, RSPI1.SPCMD3 0008 83B6h, RSPI1.SPCMD4 0008 83B8h, RSPI1.SPCMD5 0008 83BAh, RSPI1.SPCMD6 0008 83BCh, RSPI1.SPCMD7 0008 83BEh

	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 x: Don't care	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	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

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 SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI 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.
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 RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If the contents of SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, subsequent operations should not be performed.

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 SPBR (refer to section 34.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR 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 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 SSLni signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLni signal. When an SSLni 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 SSLn0 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 SSLni 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 34.3.10.1, 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.

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 SSLni signal inactive until the RSPI enables the SSLni 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 SSLni 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 SSLni 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.

34.3 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.

34.3.1 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 34.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 34.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
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
RSPCKn signal	Input	Output	Output/Hi-Z	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z	Input	Output
MISOn signal	Output/Hi-Z	Input	Input	Output	Input
SSLn0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	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 at generation of a transmit buffer empty interrupt request	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2				
Parity error detection	Supported*2,*3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.

34.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 34.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

Table 34.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi Bit for I/O Ports = 0	ODRn.Bi Bit for I/O Ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.
Note 3. When SSLn0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port 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 34.7.

Table 34.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIn Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

34.3.3 RSPI System Configuration Examples

34.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 34.5 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 SSLn0 to SSLn3 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 RSPCKn and MOSIn. 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 SSLni output of this MCU should be connected to the SSL input of the slave device.

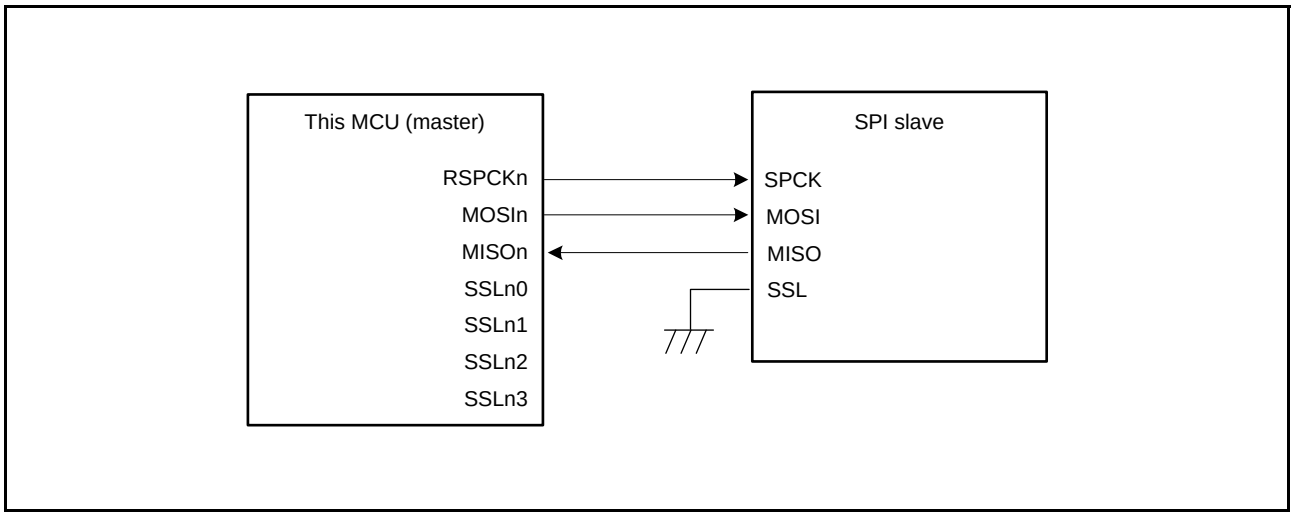


Figure 34.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

34.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 34.6 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 SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOn.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLn0 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 34.7).

Note 1. When SSLn0 is at the non-active level, the pin state is Hi-Z.

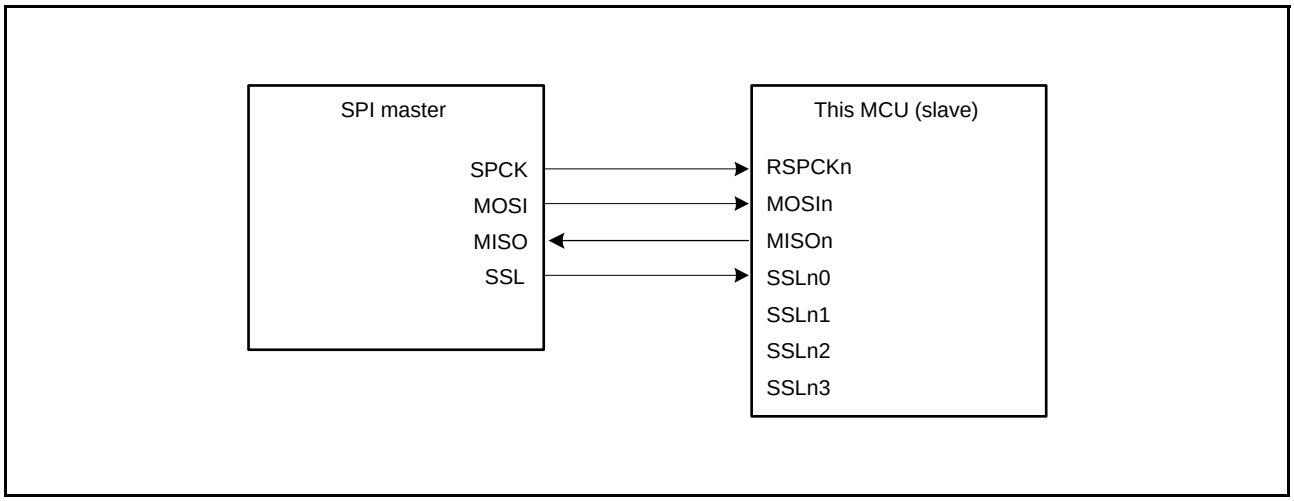


Figure 34.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

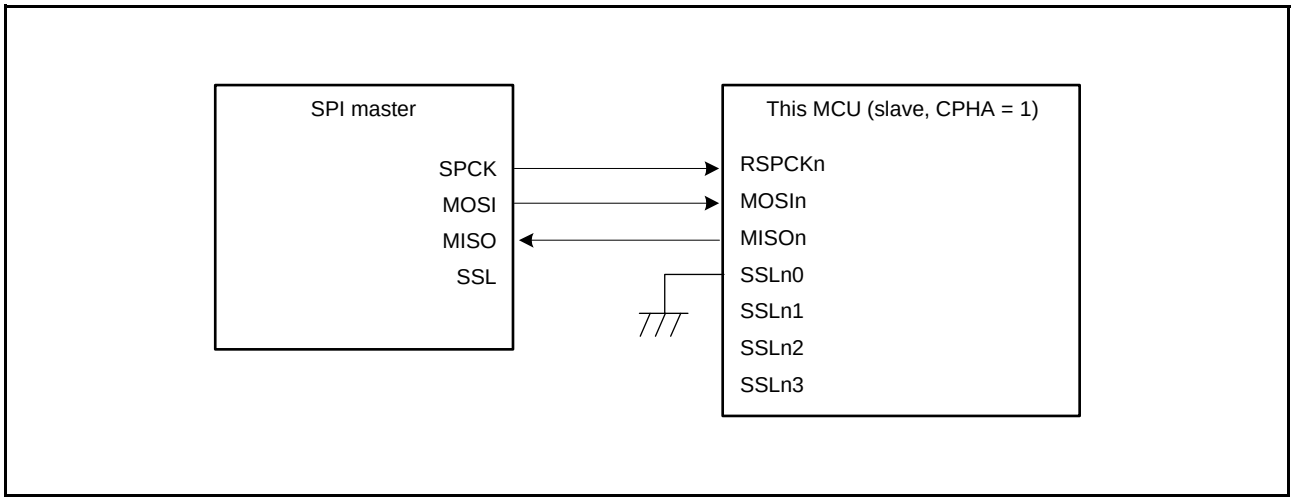


Figure 34.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

34.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 34.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 34.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn 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 MISOn input of this MCU (master). SSLn0 to SSLn3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCK, MOSI, and SSLn0 to SSLn3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

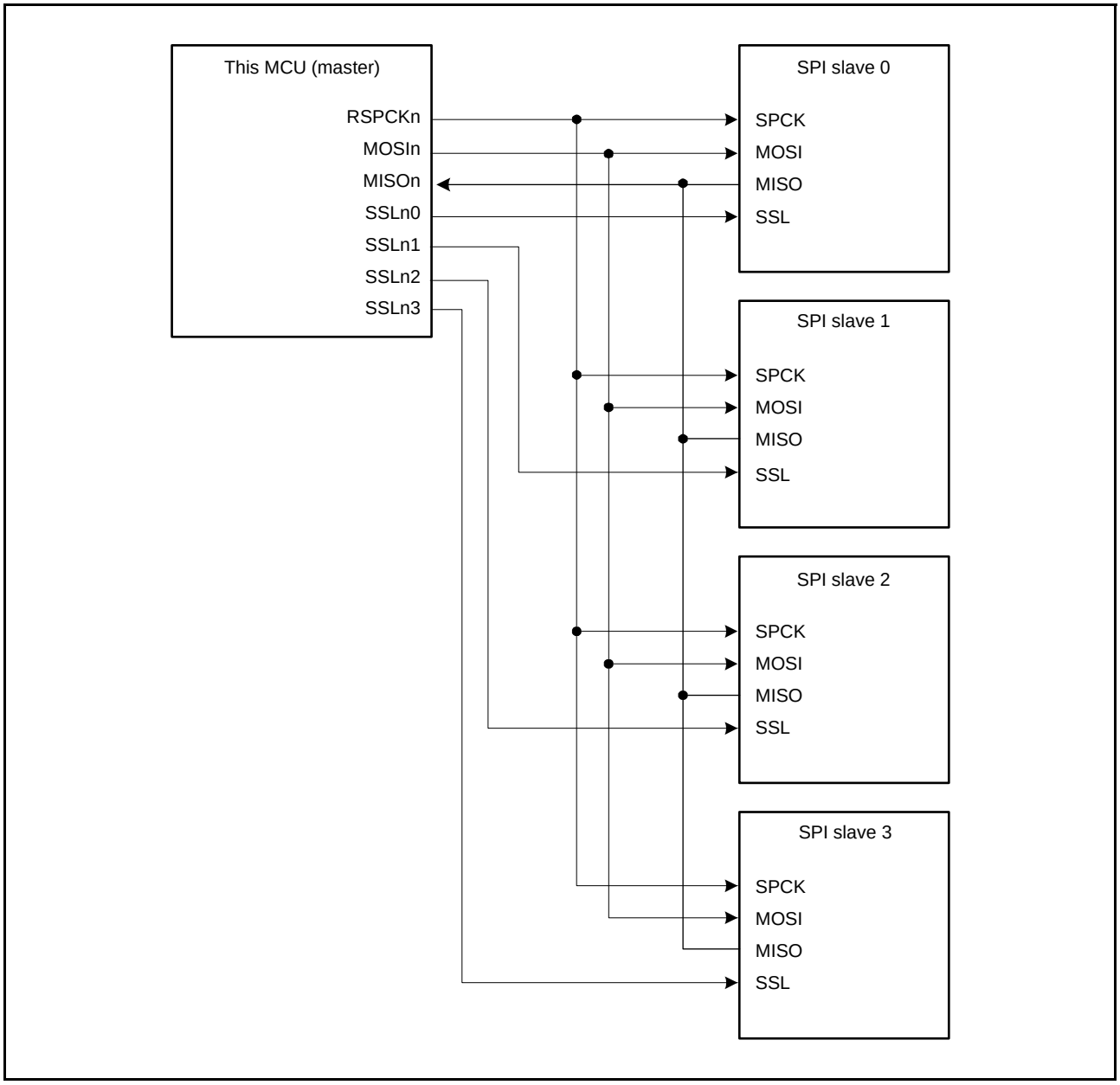


Figure 34.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

34.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 34.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 34.9, 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 RSPCKn and MOSIn inputs of the MCUs (slave X and slave Y). The MISOn 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 SSLn0 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 SSLn0 input drives MISOn.

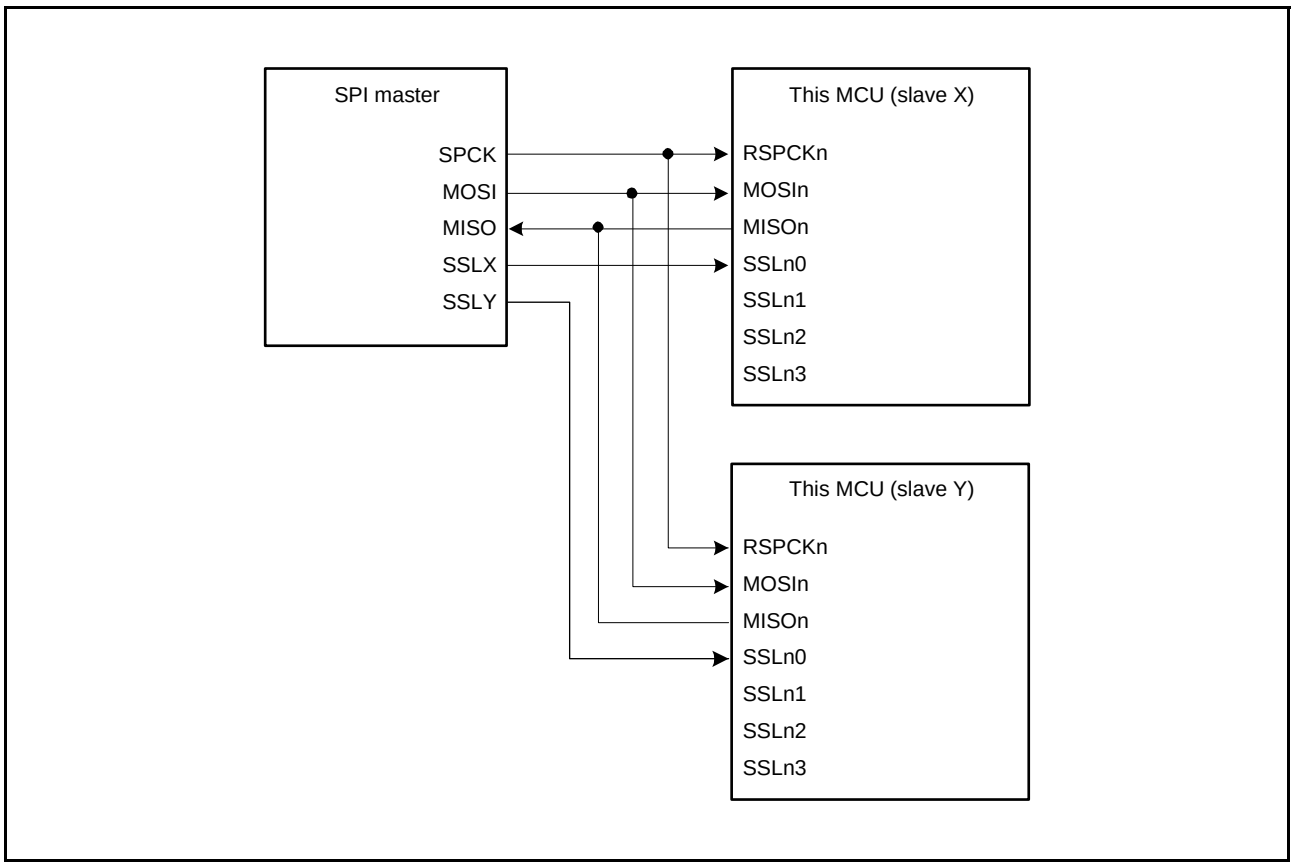


Figure 34.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

34.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 34.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 34.10, 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 RSPCKn and MOSIn 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 MISOn inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLn0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLn0 input of this MCU (master X). The SSLn1 and SSLn2 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 SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of this MCU is not required.

This MCU drives RSPCKn, MOSIn, SSLn1, and SSLn2 when the SSLn0 input level is high. When the SSLn0 input level is low, this MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 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.

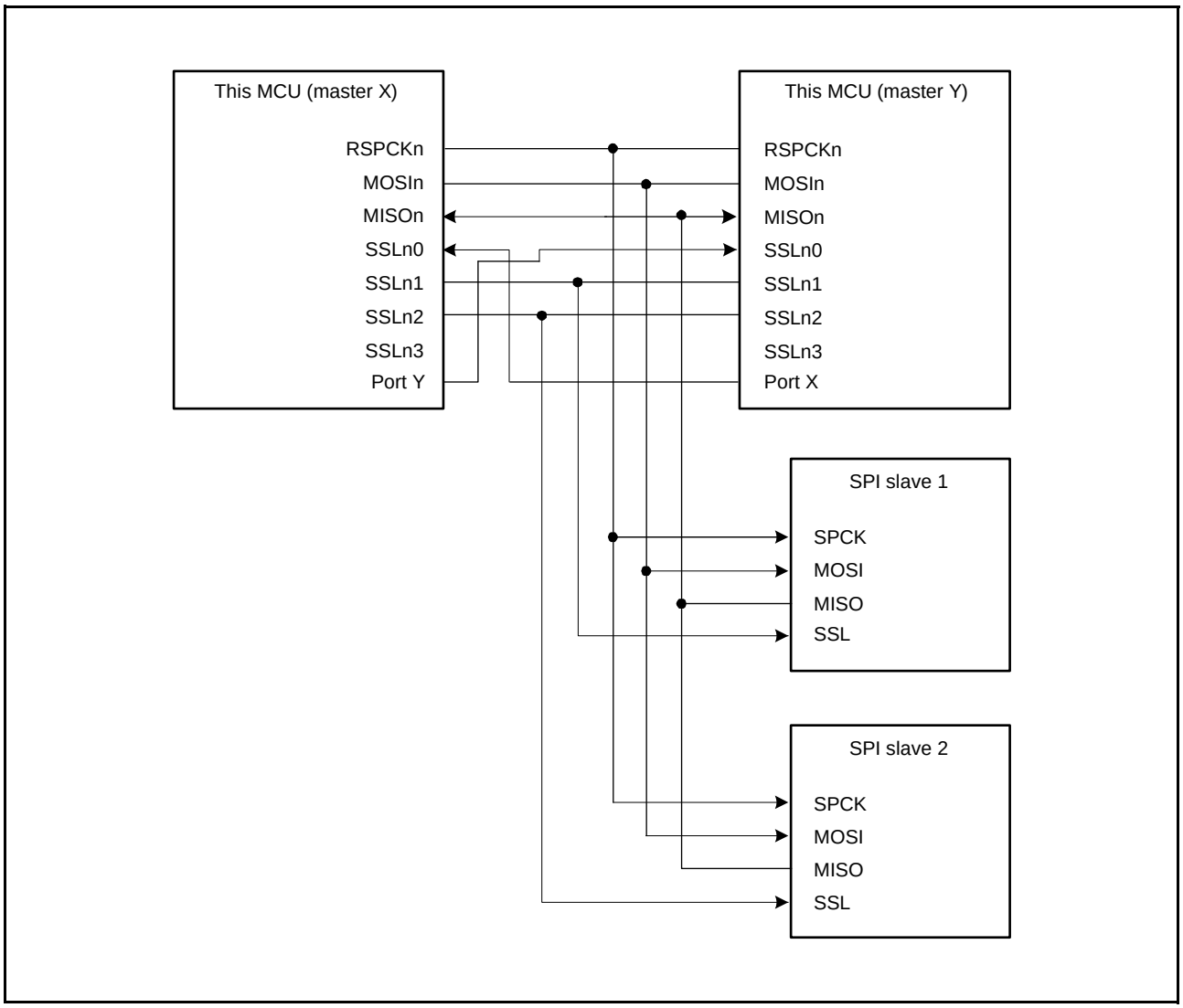


Figure 34.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

34.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 34.11 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, SSLn0 to SSLn3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKn and MOSIn. The SPI slave drives the MISO.

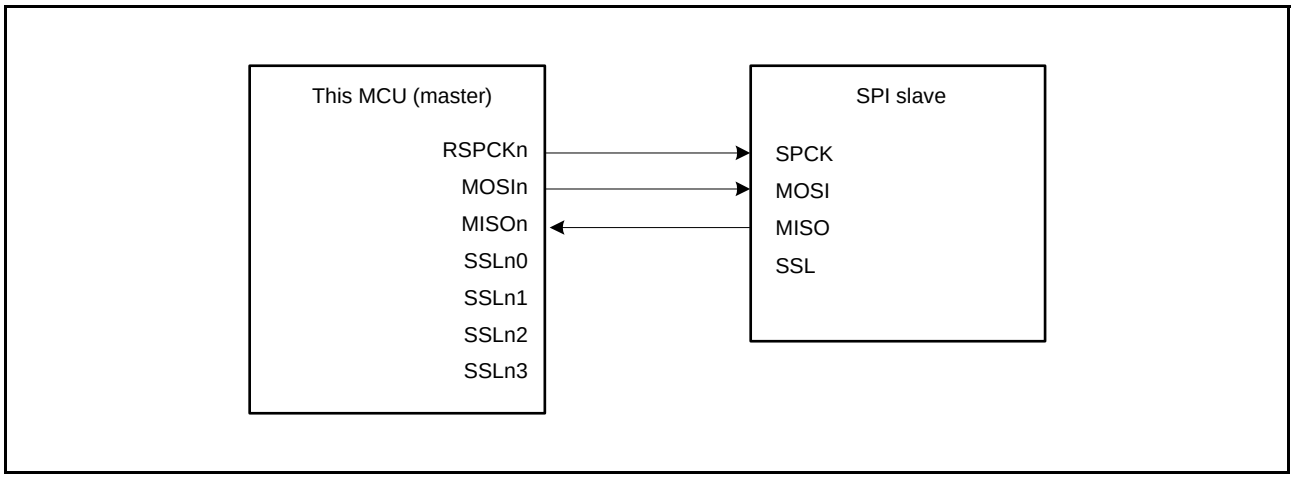


Figure 34.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

34.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 34.12 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 MISOn and the SPI master drives the SPCK and MOSI. In addition, SSLn0 to SSLn3 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.

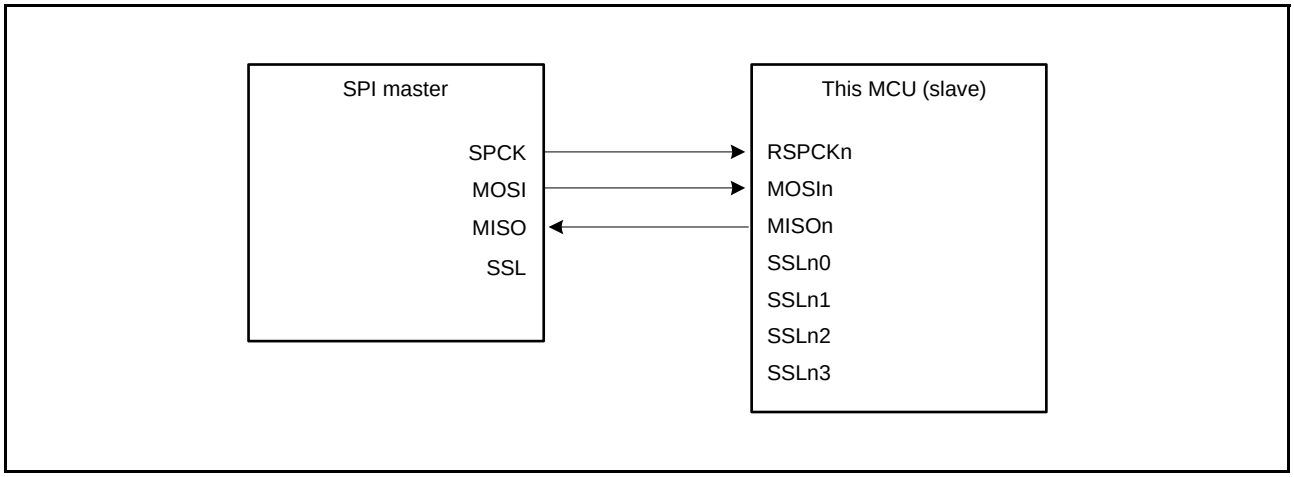


Figure 34.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

34.3.4 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). 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.

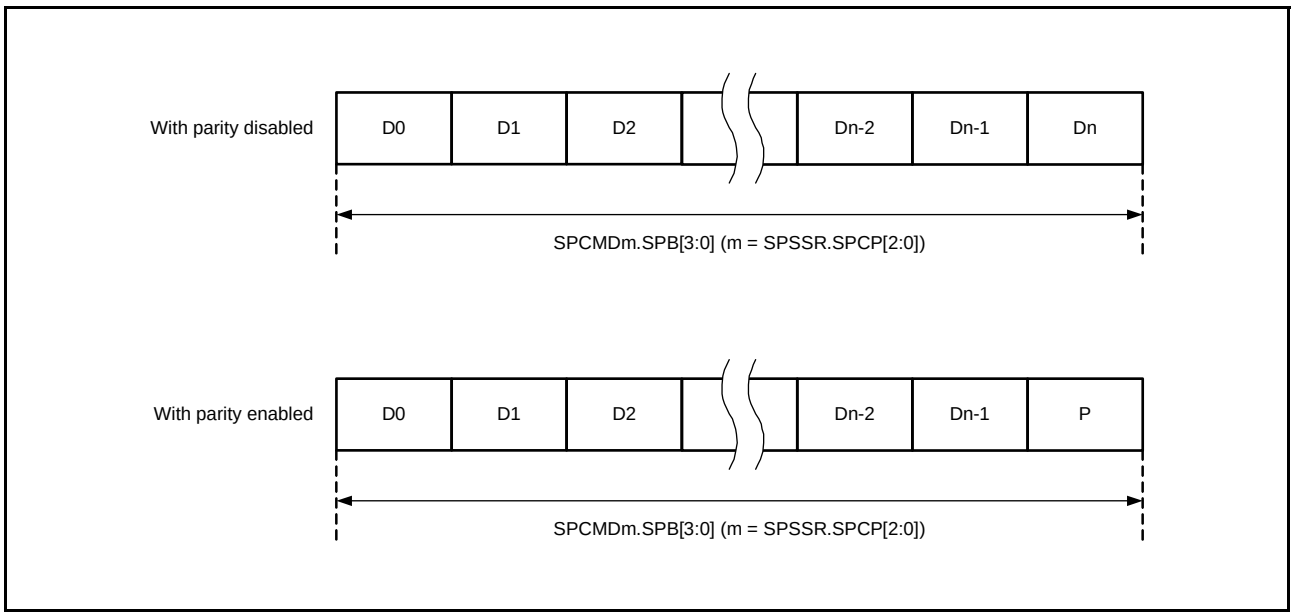


Figure 34.13 Outline of the Data Format (with Parity Disabled/Enabled)

34.3.4.1 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 34.14 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.

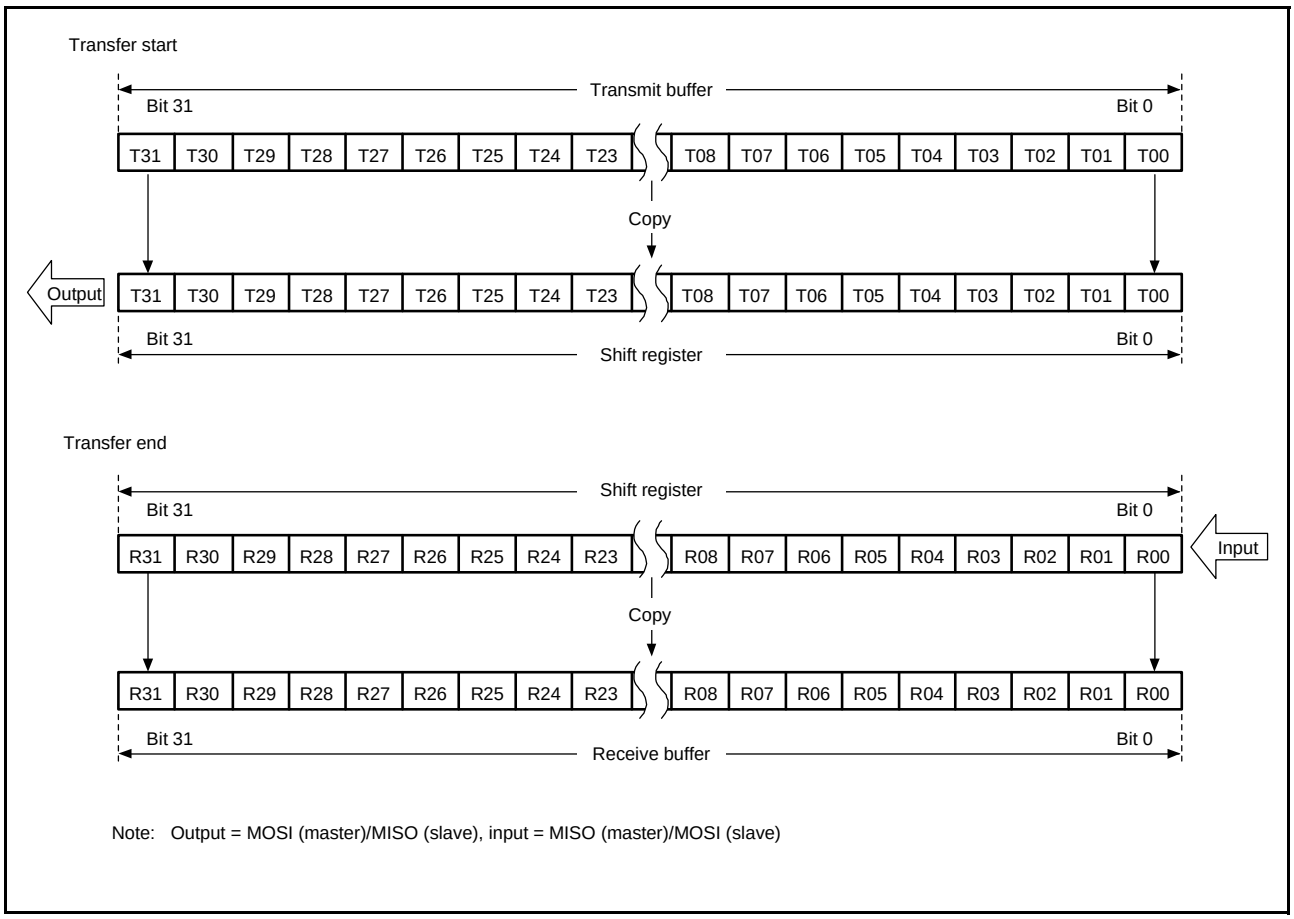


Figure 34.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 34.15 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-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. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

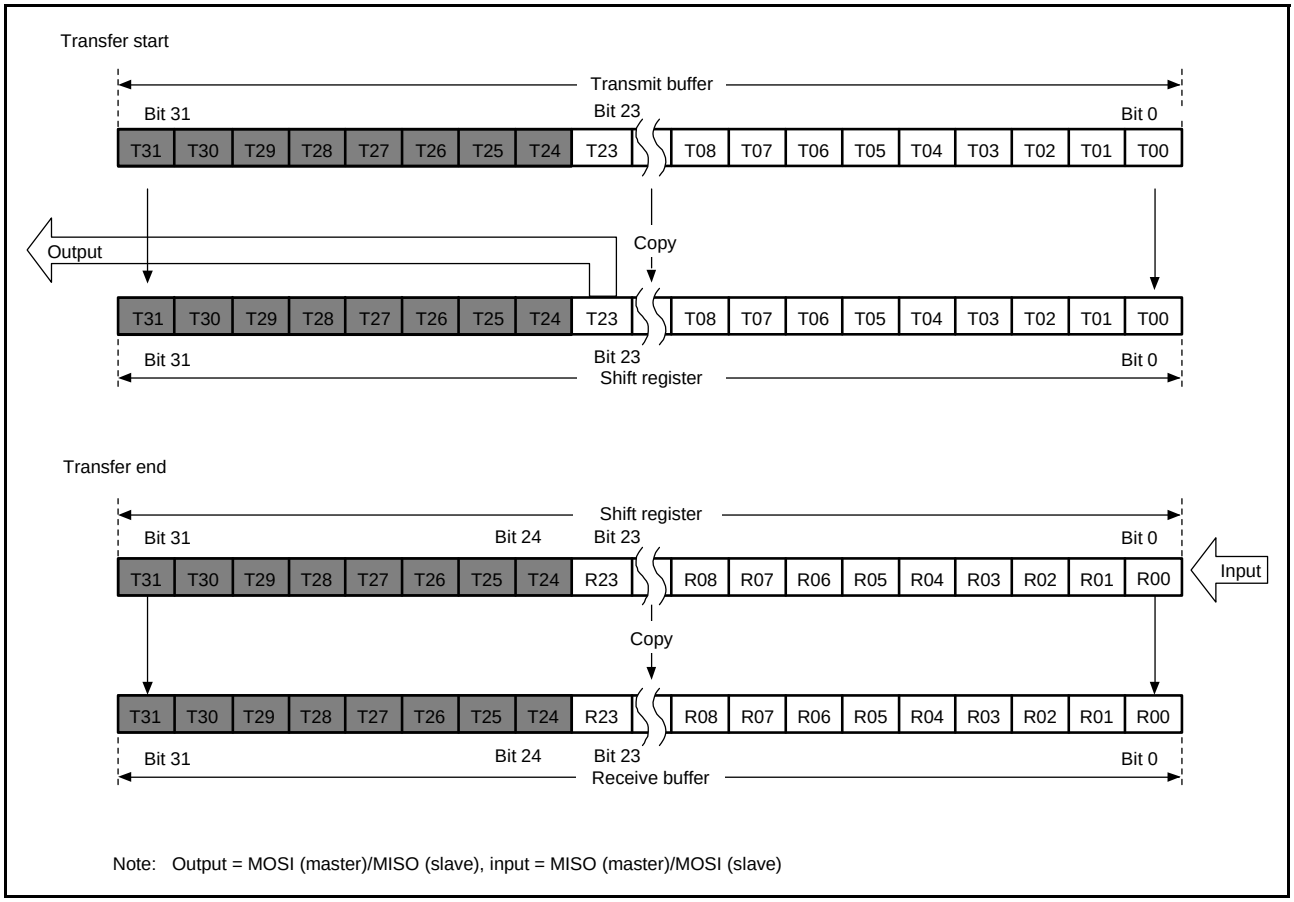


Figure 34.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.16 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.

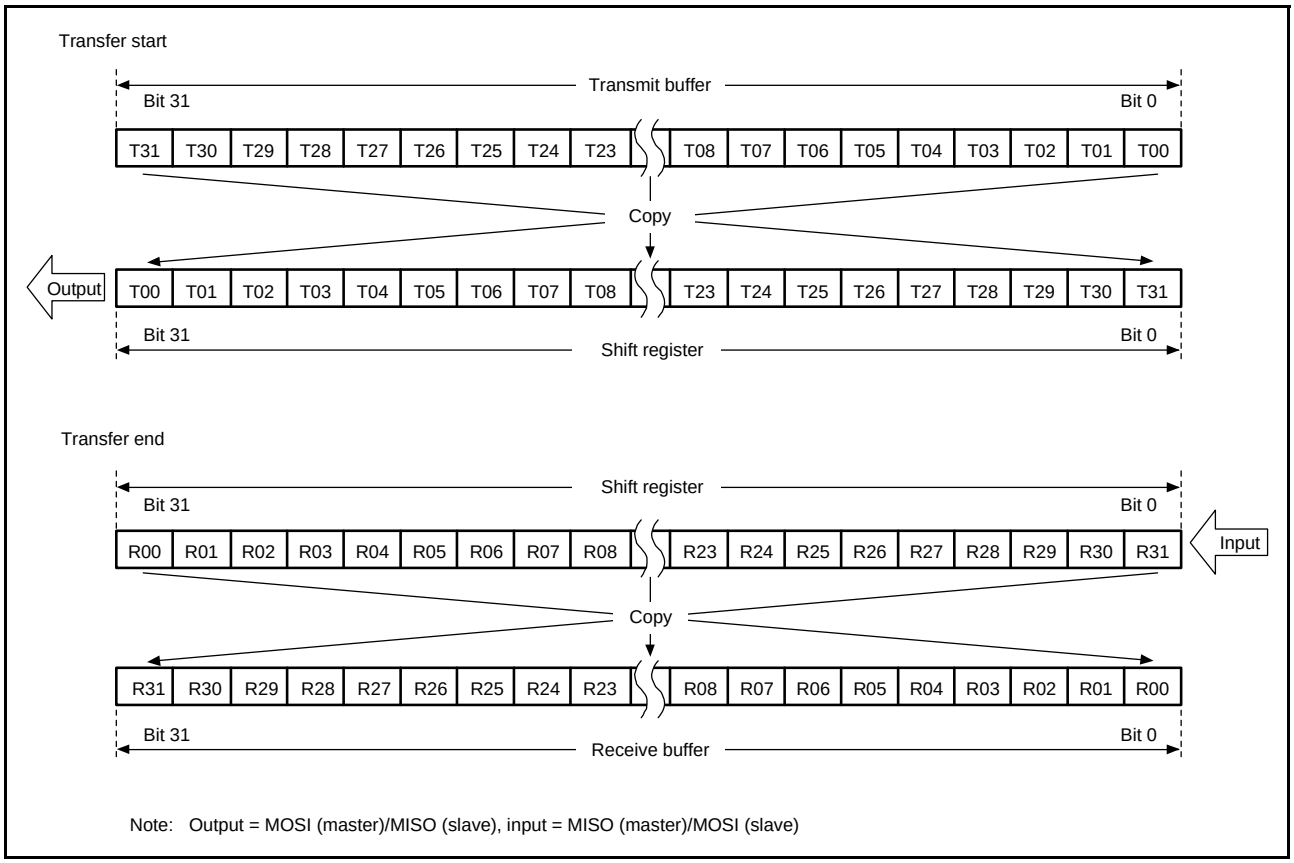


Figure 34.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 34.17 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-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 value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

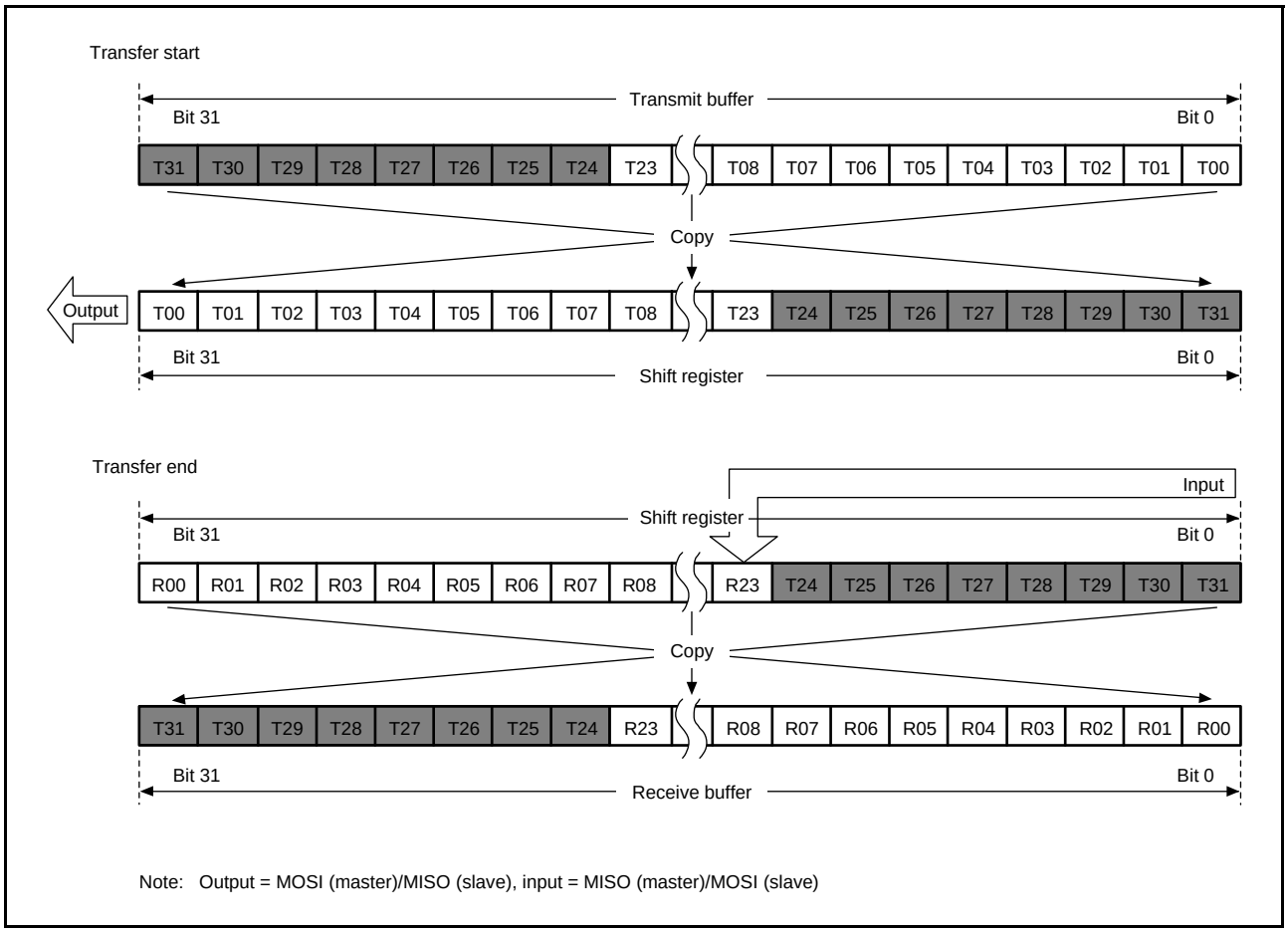


Figure 34.17 LSB First Transfer (24-Bit Data, Parity Disabled)

34.3.4.2 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 34.18 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.

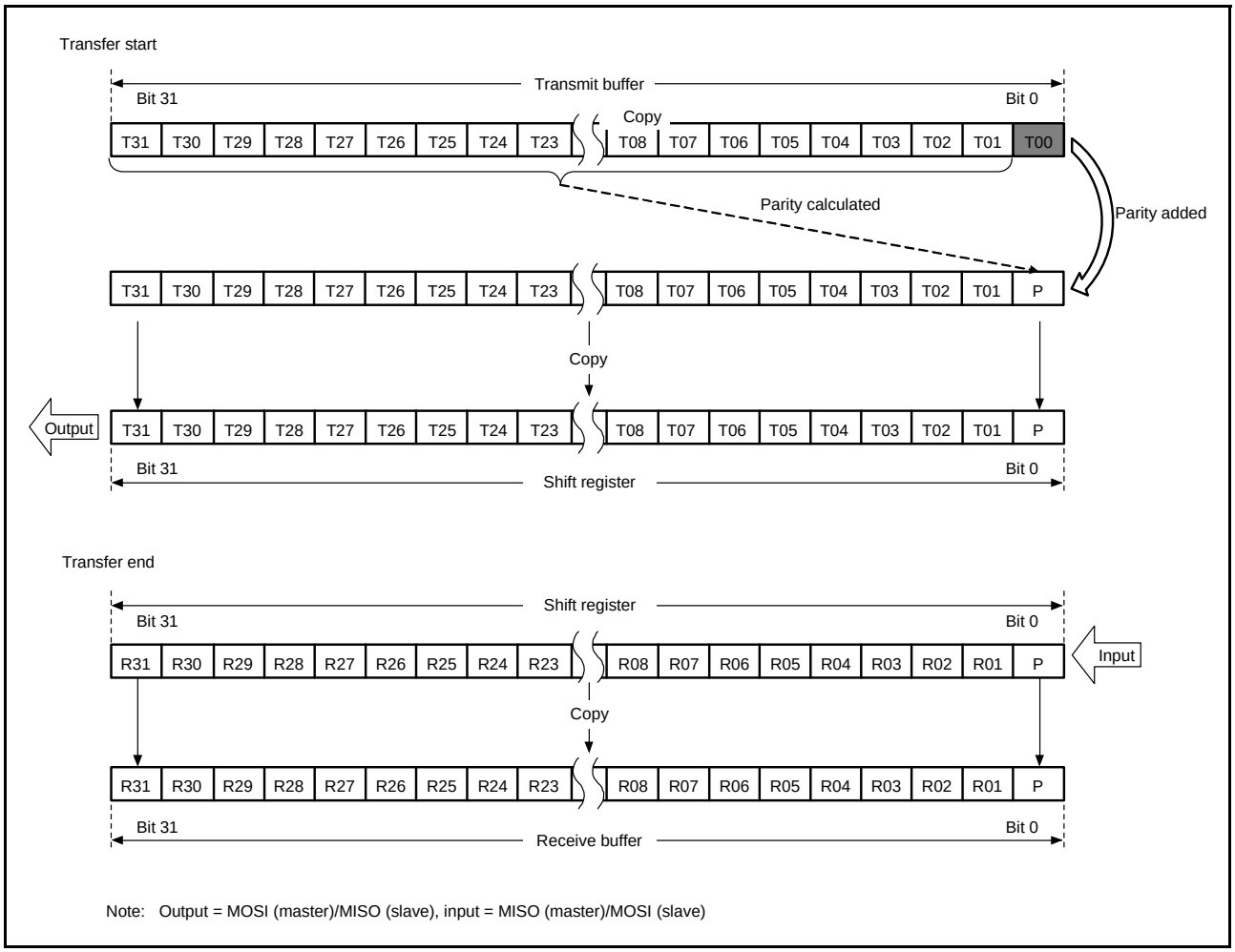


Figure 34.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 34.19 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, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

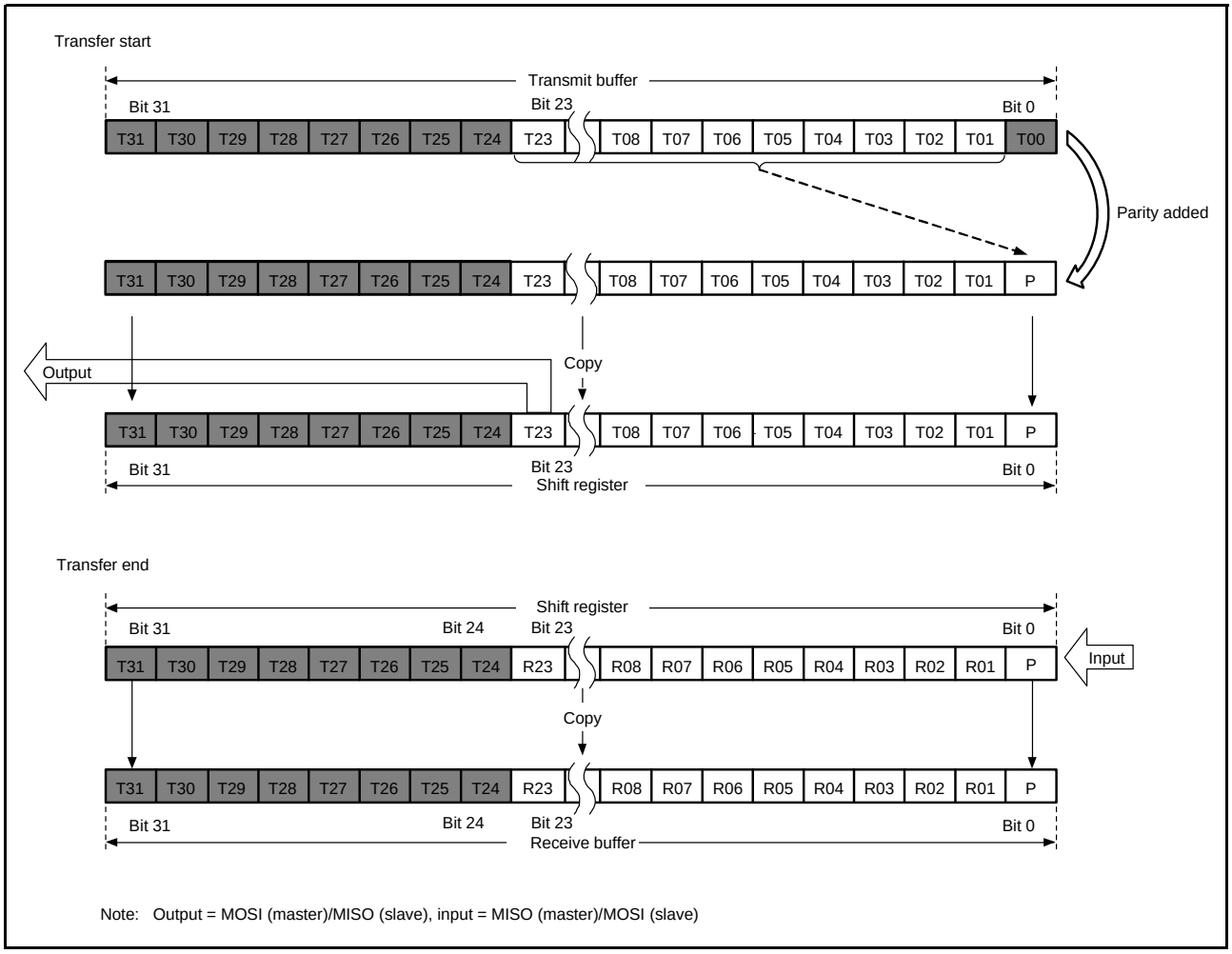


Figure 34.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.20 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.

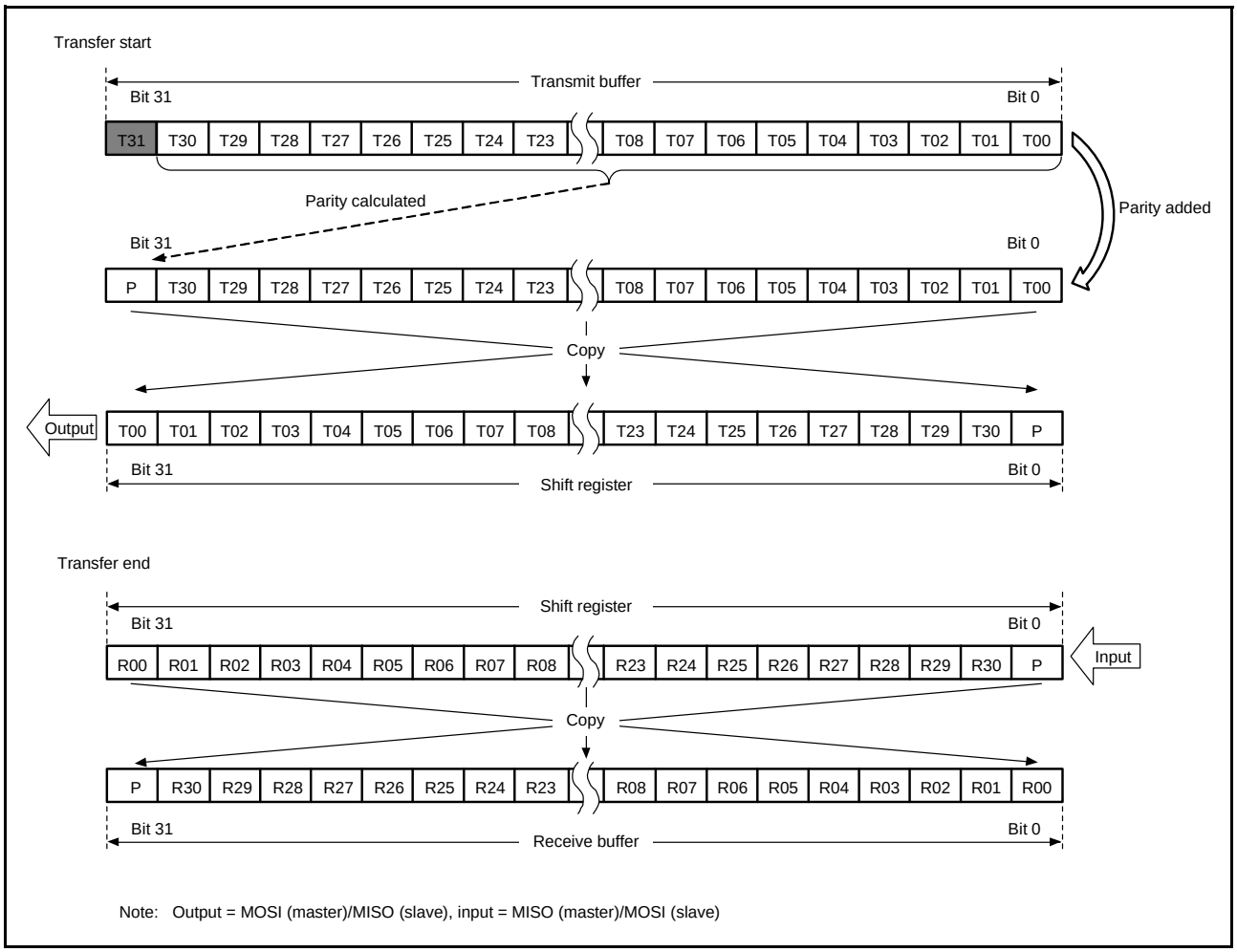


Figure 34.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 34.21 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, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

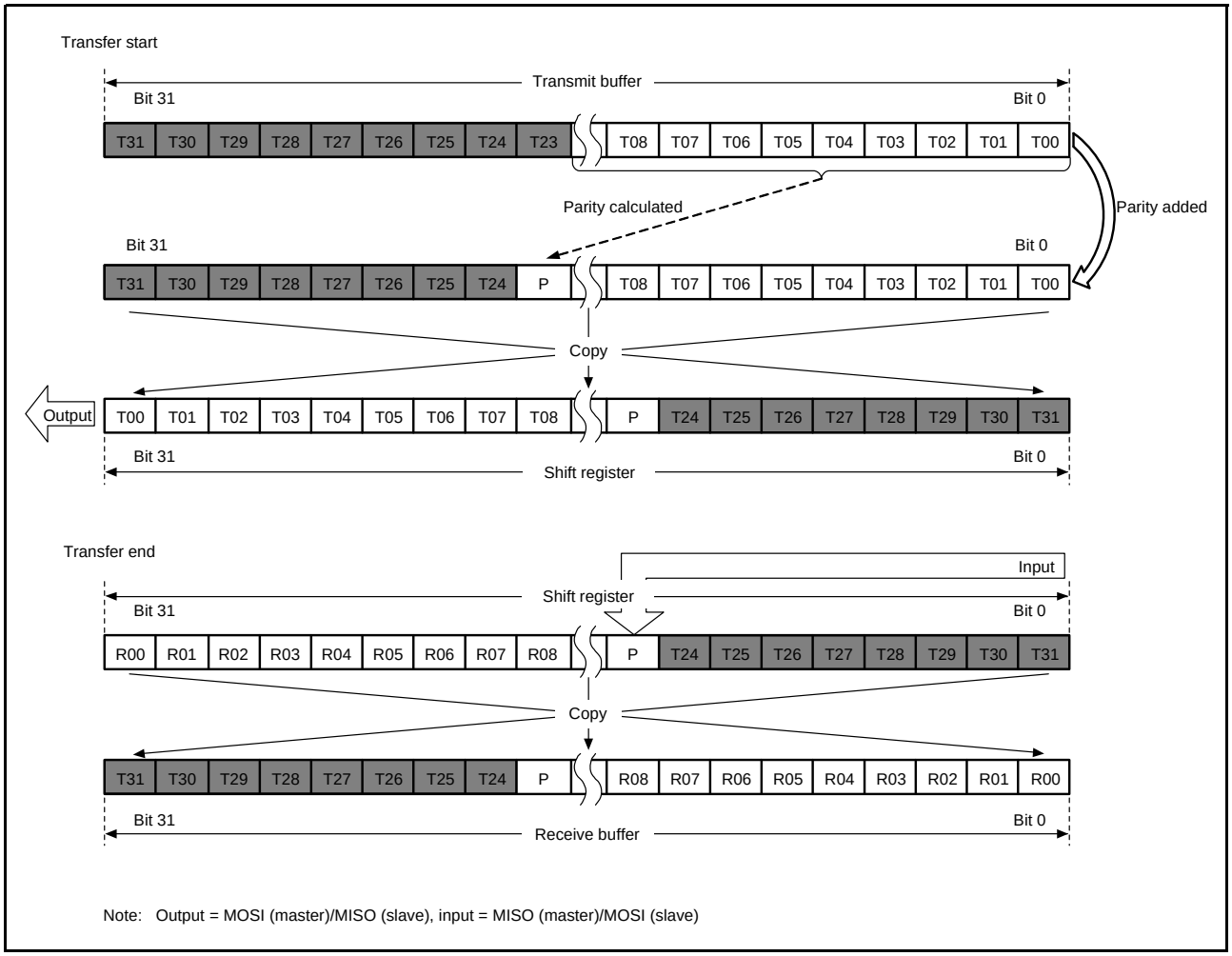


Figure 34.21 LSB First Transfer (24-Bit Data, Parity Enabled)

34.3.5 Transfer Format

34.3.5.1 CPHA = 0

Figure 34.22 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 34.22, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCKn (CPOL = 1) indicates the RSPCKn 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 34.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals commences at an SSLni signal assertion timing. The first RSPCKn signal change timing that occurs after the SSLni signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLni signal assertion to RSPCKn oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKn oscillation to an SSLni signal negation (SSL negation delay). t3 denotes a period in which SSLni 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 34.3.10.1, Master Mode Operation.

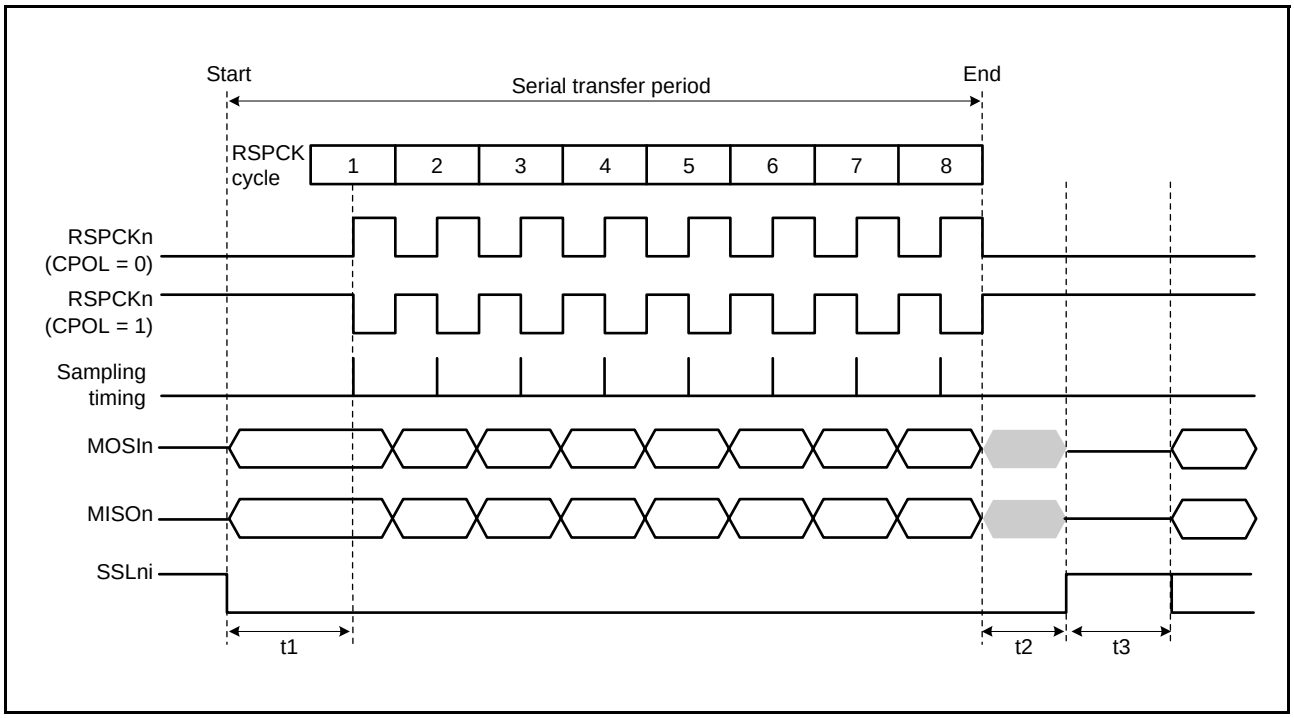


Figure 34.22 RSPI Transfer Format (CPHA = 0)

34.3.5.2 CPHA = 1

Figure 34.23 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 SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 34.23, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKn 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 34.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal commences at an SSLni signal assertion timing. The output of valid data to the MOSIn and MISOn signals commences at the first RSPCKn signal change timing that occurs after the SSLni 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 RSPCKn 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 34.3.10.1, Master Mode Operation.

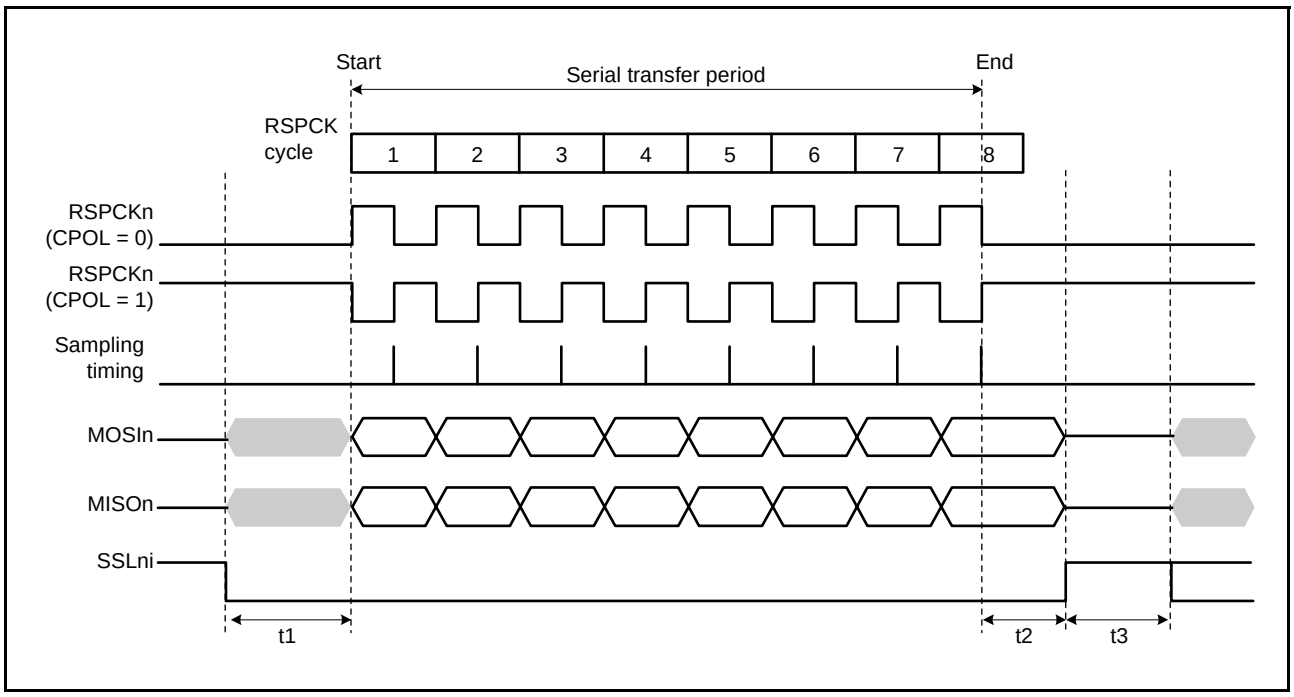


Figure 34.23 RSPI Transfer Format (CPHA = 1)

34.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 34.24 and Figure 34.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

34.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 34.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 34.24, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

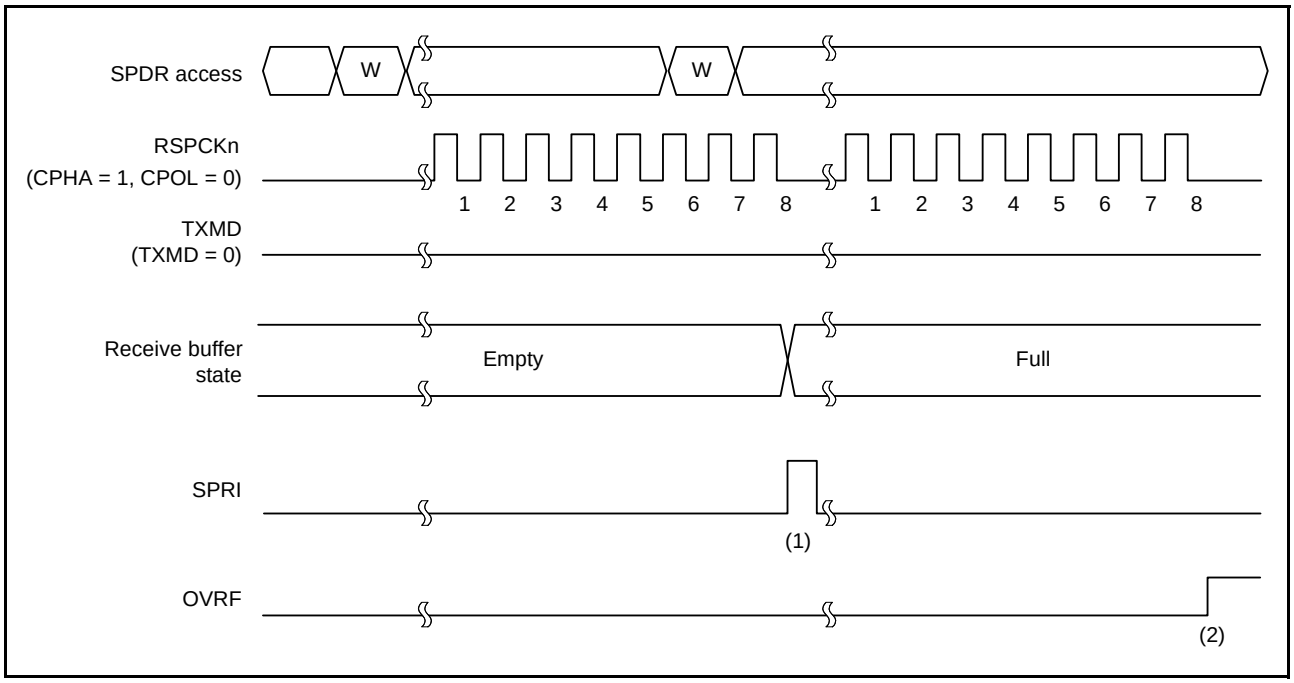


Figure 34.24 Operation Example of SPCR.TXMD = 0

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 SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of SPDR 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.

34.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 34.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 34.25, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

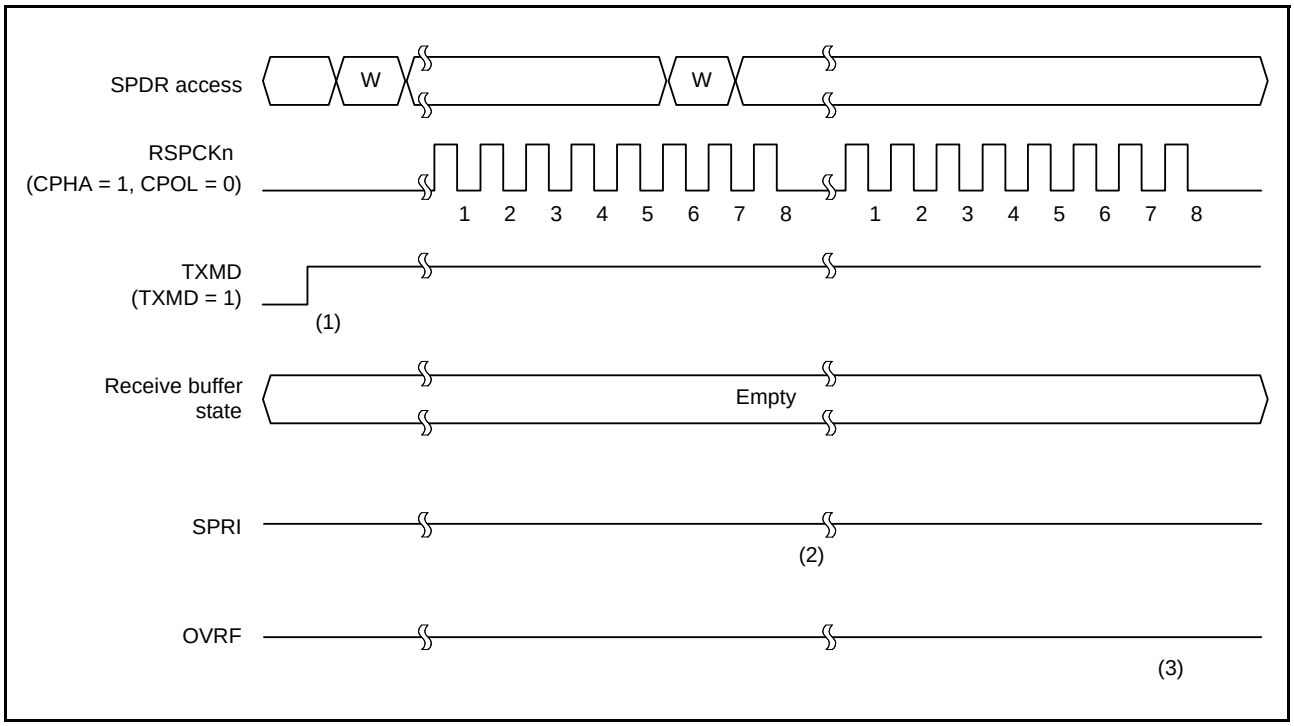


Figure 34.25 Operation Example of SPCR.TXMD = 1

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.OVRF flag is 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the RSPI does not copy the data in 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 performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits transmit data but does not receive received data. Therefore, the SPSR.OVRF flag remains 0 at the timings of (1) to (3).

34.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 34.26 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 34.26 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 34.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

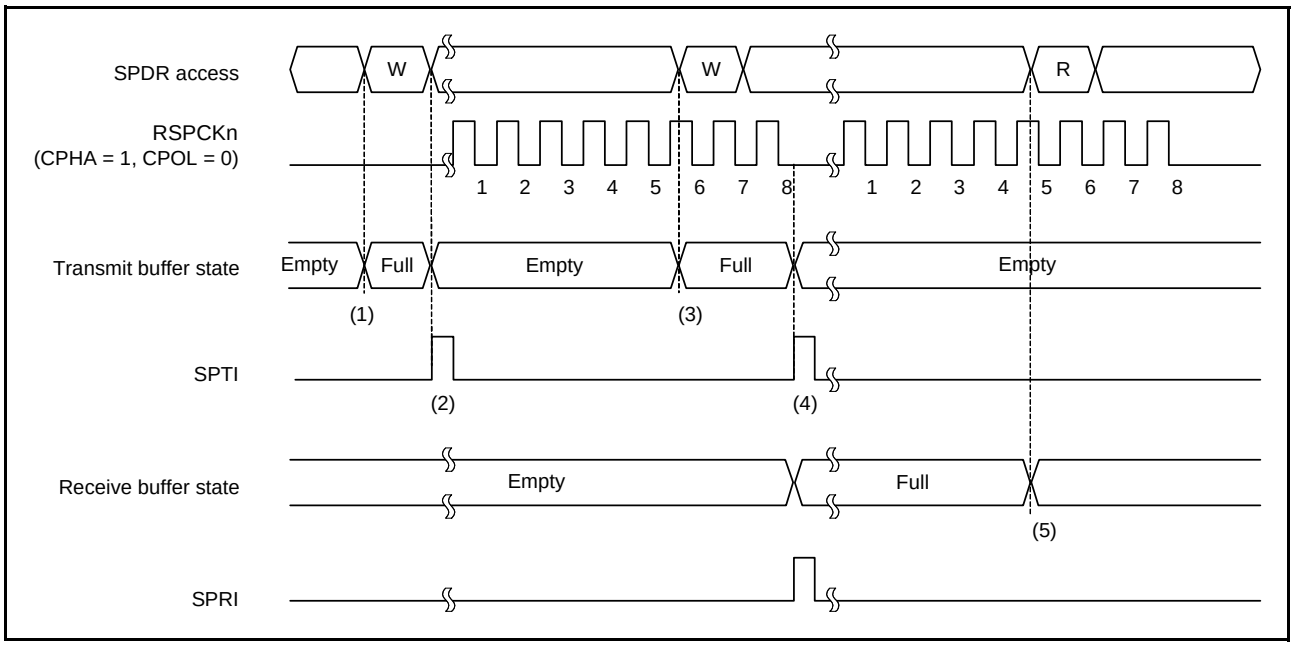


Figure 34.26 Operation Example of SPTI and SPRI Interrupts

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 SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer.
2. If the shift register is empty, the RSPI copies the data in the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI). How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 34.3.10, SPI Operation, and section 34.3.11, Clock Synchronous Operation.
3. When transmit data is written to SPDR by the transmit buffer empty interrupt routine, the data is transferred to the transmit buffer. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data in the shift register to the receive buffer and generates a receive buffer full interrupt request (SPRI). 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 copies the data in 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 SPDR is read by the receive buffer full interrupt routine, the receive data can be read.

If SPDR is written to when the transmit buffer holds data that has not yet been transmitted, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure to use a transmit buffer empty interrupt request. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
If the RSPI function is disabled (the SPCR.SPE bit being 0), set the SPTIE bit to 0.
When serial transfer ends with the receive buffer being full, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 34.3.8, 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 RSPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmission and reception 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 transmission and reception buffers. Refer to section 14, Interrupt Controller (ICUb), for the interrupt vector numbers.

34.3.8 Error Detection

In the normal RSPI serial transfer, 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 RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 34.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 34.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is transmitted.	None
3	SPDR is read when the receive buffer is empty.	Previously received data is output.	None
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 synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOn output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 34.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, write operations to SPDR should be executed using a transmit buffer empty interrupt request.

Likewise, the RSPI does not detect an error on operation 2. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 2 as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

Similarly, the RSPI does not detect an error on operation 3. To prevent extraneous data from being read, SPDR read operation should be executed using an RSPI receive buffer full interrupt request.

An overrun error shown in 4 is described in section 34.3.8.1, Overrun Error. A parity error shown in 5 is described in section 34.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 34.3.8.3, Mode Fault Error. For the transmit and receive interrupts, refer to section 34.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

34.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR 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 SPSR with the OVRF flag set to 1.
Figure 34.27 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 34.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 34.27, 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

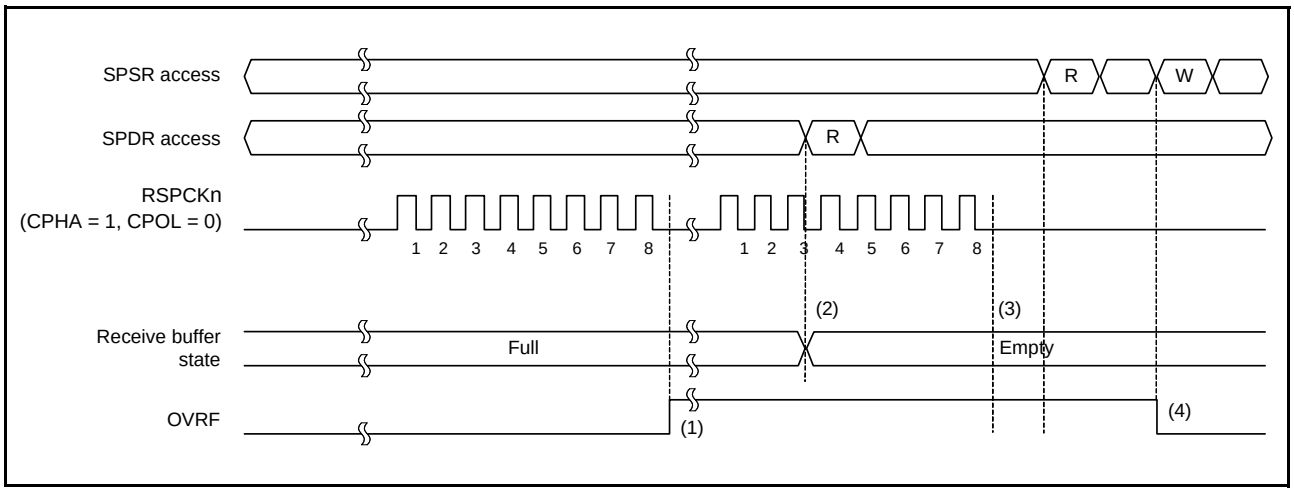


Figure 34.27 Operation Example of OVRF Flag

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 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 SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. When SPDR is read, the RSPI outputs the data in the receive buffer can be read. The receive buffer becoming empty does not set the OVRF flag to 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. 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 SPSR 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 SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is 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.
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.

34.3.8.2 Parity Error

If full-duplex synchronous serial communications 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 SPSR register is read with the PERF flag set to 1.
Figure 34.28 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 34.28 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 34.28, full-duplex synchronous serial communications 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 RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

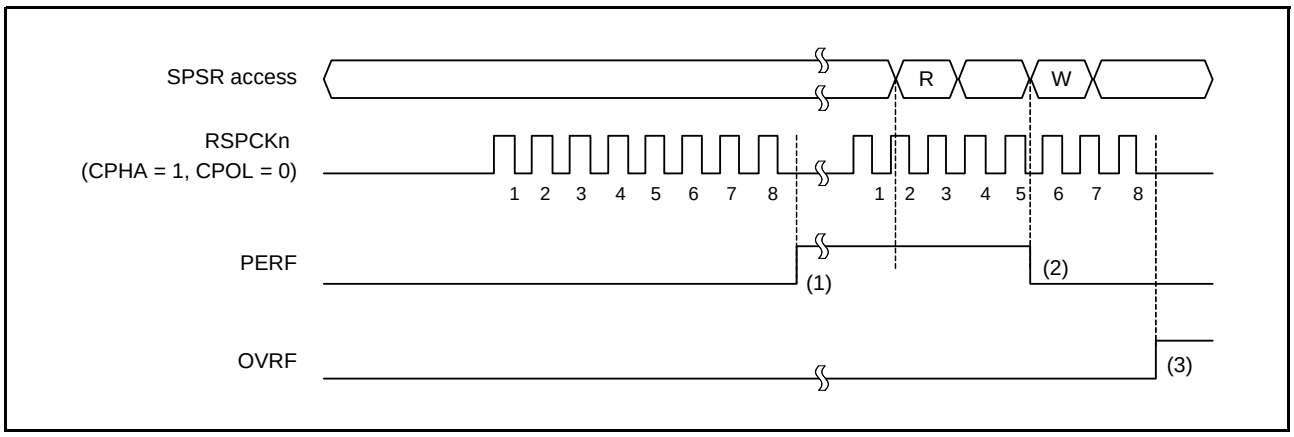


Figure 34.28 Operation Example of PERF Flag

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 SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. If 0 is written to the PERF flag after 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 RSPI 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 SPSR. When the RSPI 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.

34.3.8.3 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 SSLn0 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 SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLn0 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 SSLn0 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 34.3.9, 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 SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode-fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI 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 RSPI. To enable the RSPI function after the detection of a mode fault error, the MODF flag must be set to 0.

34.3.9 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, 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.

34.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI 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 RSPI
- Initializing the transmit buffer of the RSPI

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.OVRF and SPSR.MODF 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 in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state. 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. To disable any transmit buffer empty interrupt after a mode fault error is detected, use an error handling routine to write 0 to the SPTIE bit.

34.3.9.2 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 34.3.9.1, Initialization by Clearing the SPE Bit.

34.3.10 SPI Operation

34.3.10.1 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 34.3.8, 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 (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, 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 34.3.5, Transfer Format. The polarity of the SSLni 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 RSPCKn edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX), 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 SSLni output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR 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 SPCMDm register: SSLni 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. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of 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 SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

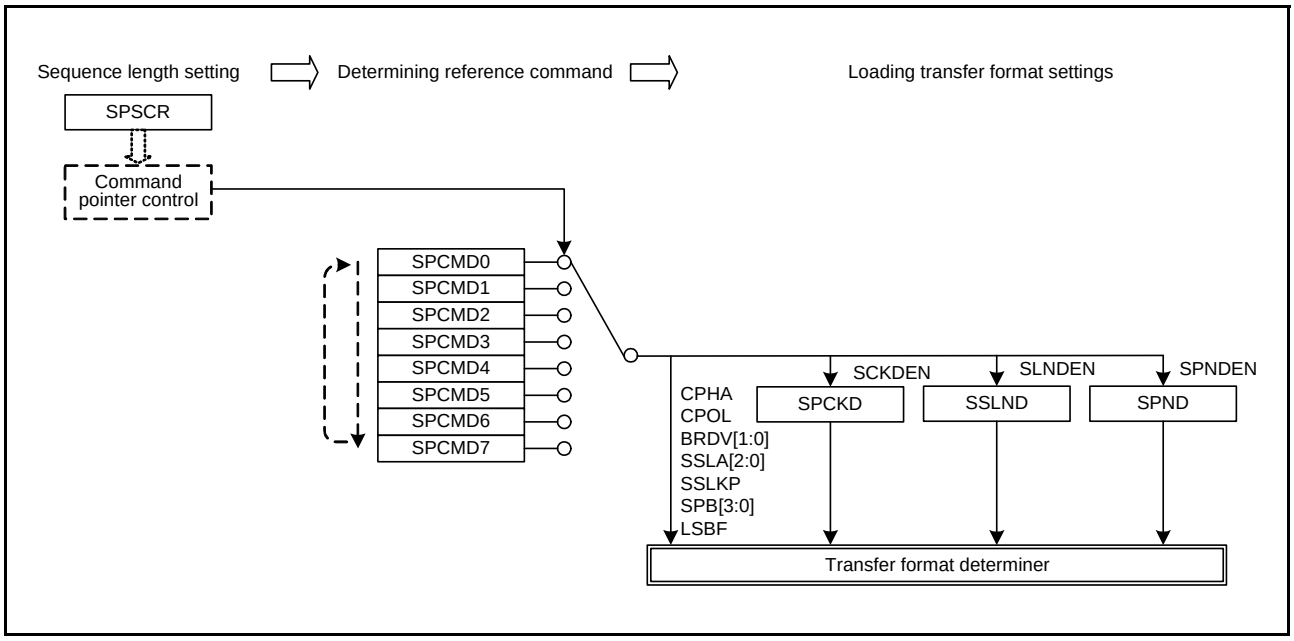


Figure 34.29 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

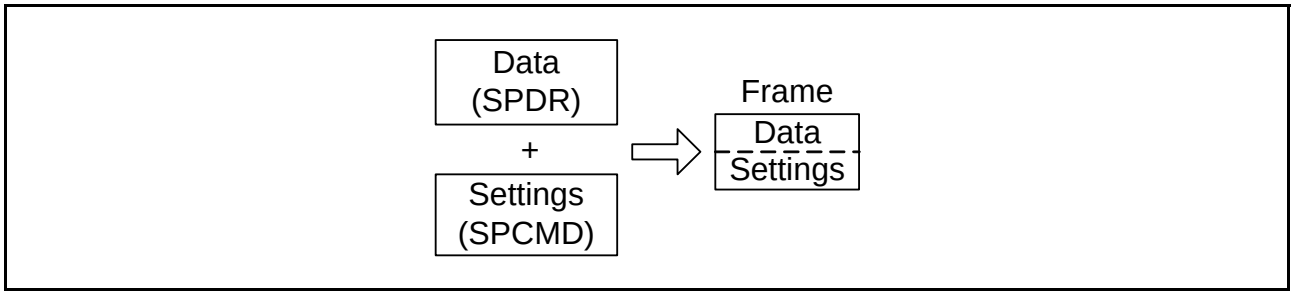


Figure 34.30 Concept of a Frame

Figure 34.31 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 34.4.

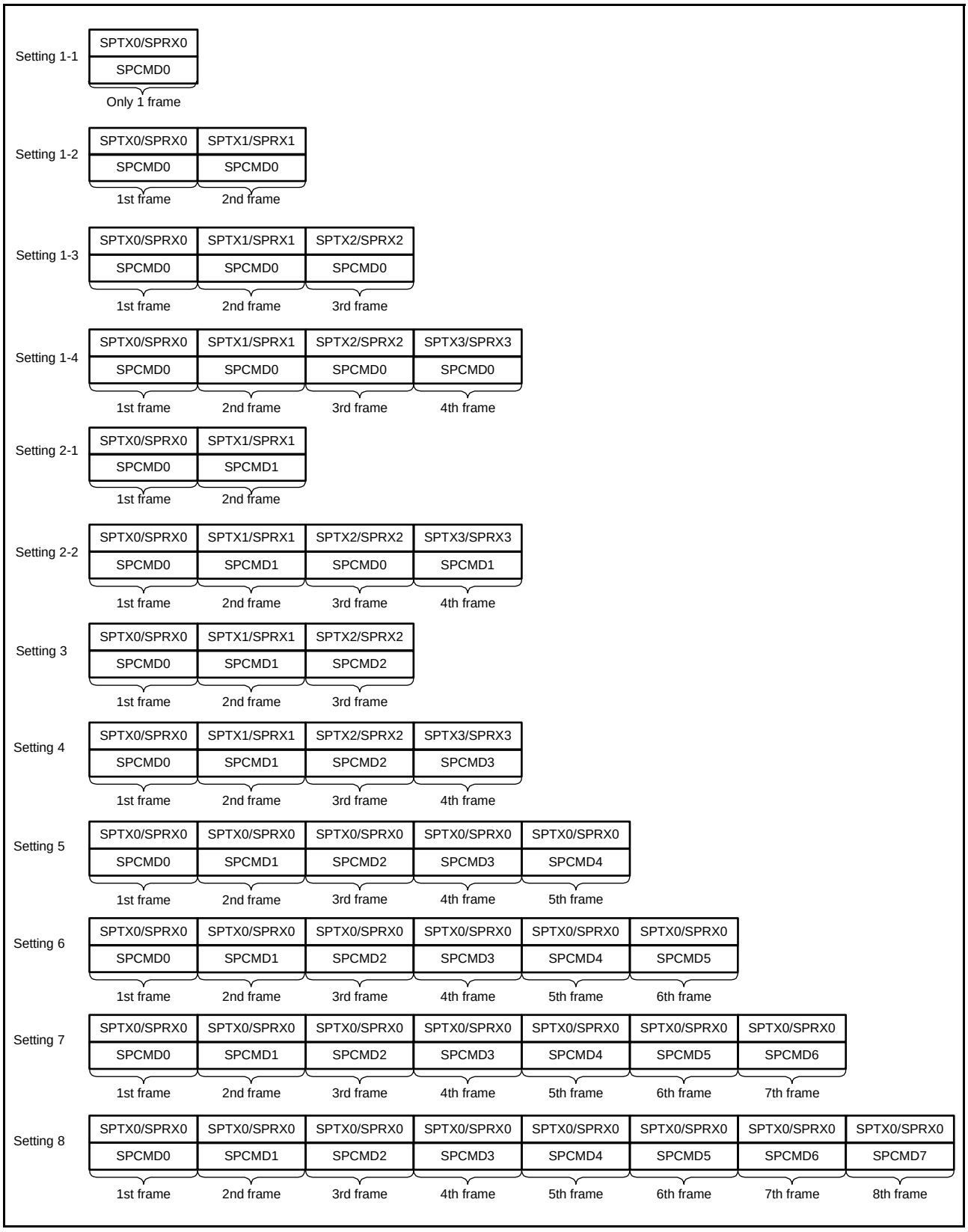


Figure 34.31 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).
Figure 34.32 shows an example of an SSLni 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 34.32. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.

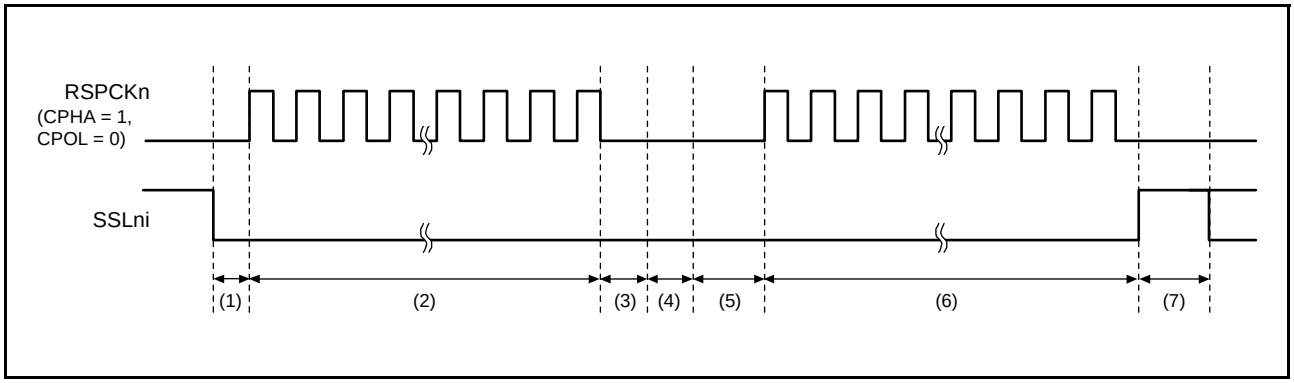


Figure 34.32 Example of Burst Transfer Operation Using SSLKP Bit

(1) Based on SPCMD0, the RSPI asserts the SSLni signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to SPCMD0.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLni signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on SPCMD1, the RSPI asserts the SSLni signal and inserts RSPCK delays.
(6) The RSPI executes serial transfers according to SPCMD1.
(7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLni signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLni signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLni signal status to SSLni signal assertion ((5) in Figure 34.32) corresponding to the command for the next transfer. Note that if such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLni 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 SSLni signal assertion for the next transfer that is detected internally.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI 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 SPCKD, as listed in Table 34.9. For a definition of RSPCK delay, refer to section 34.3.5, Transfer Format.

Table 34.9 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 of the RSPI 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 SSLND, as listed in Table 34.10. For a definition of SSL negation delay, refer to section 34.3.5, Transfer Format.

Table 34.10 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 of the RSPI 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 34.11. For a definition of next-access delay, refer to section 34.3.5, Transfer Format.

Table 34.11 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 34.33 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.

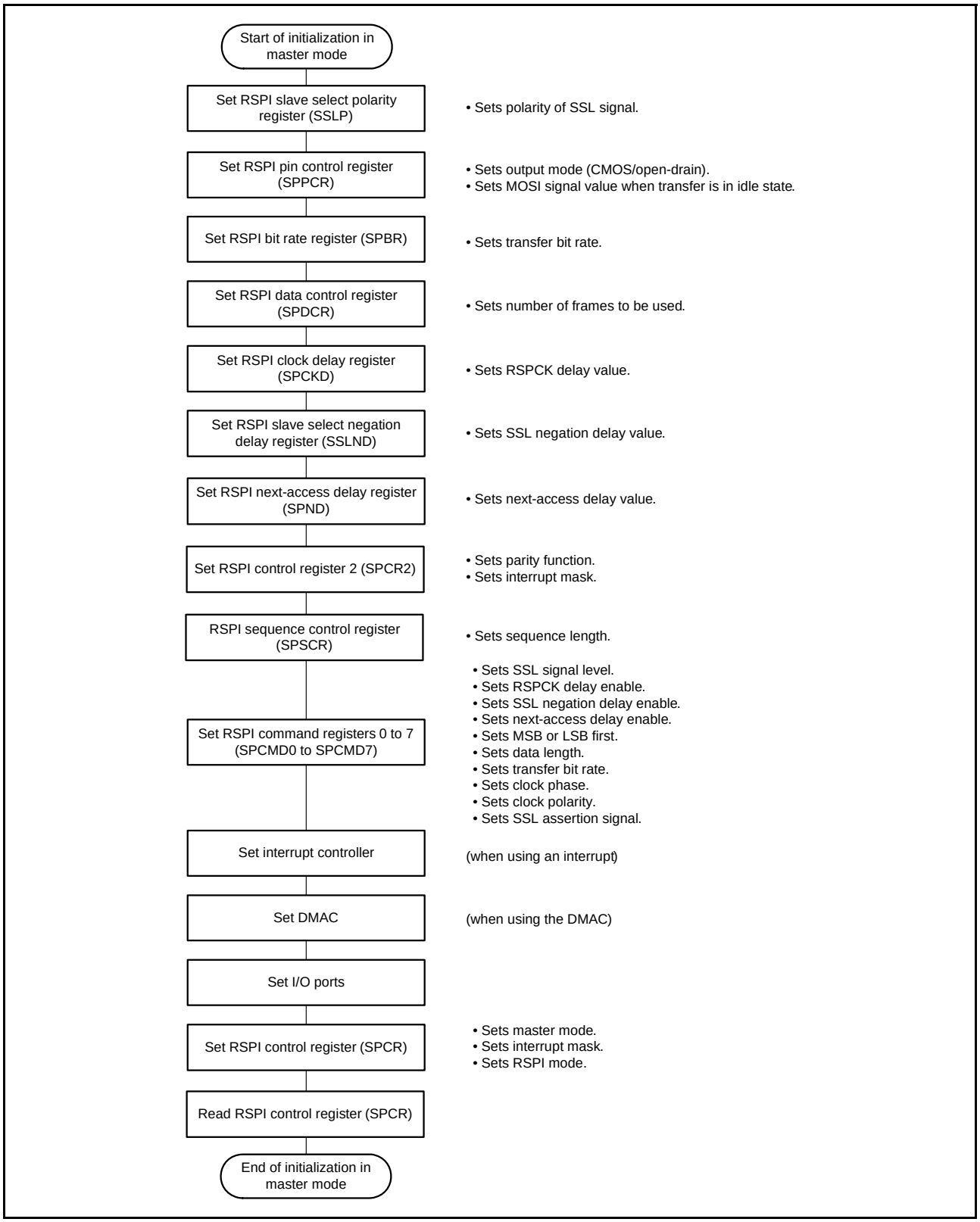


Figure 34.33 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 34.34 to Figure 34.36 show examples of the flow of software processing.

(a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission after the last writing of data for transmission if the SPII interrupt is enabled.

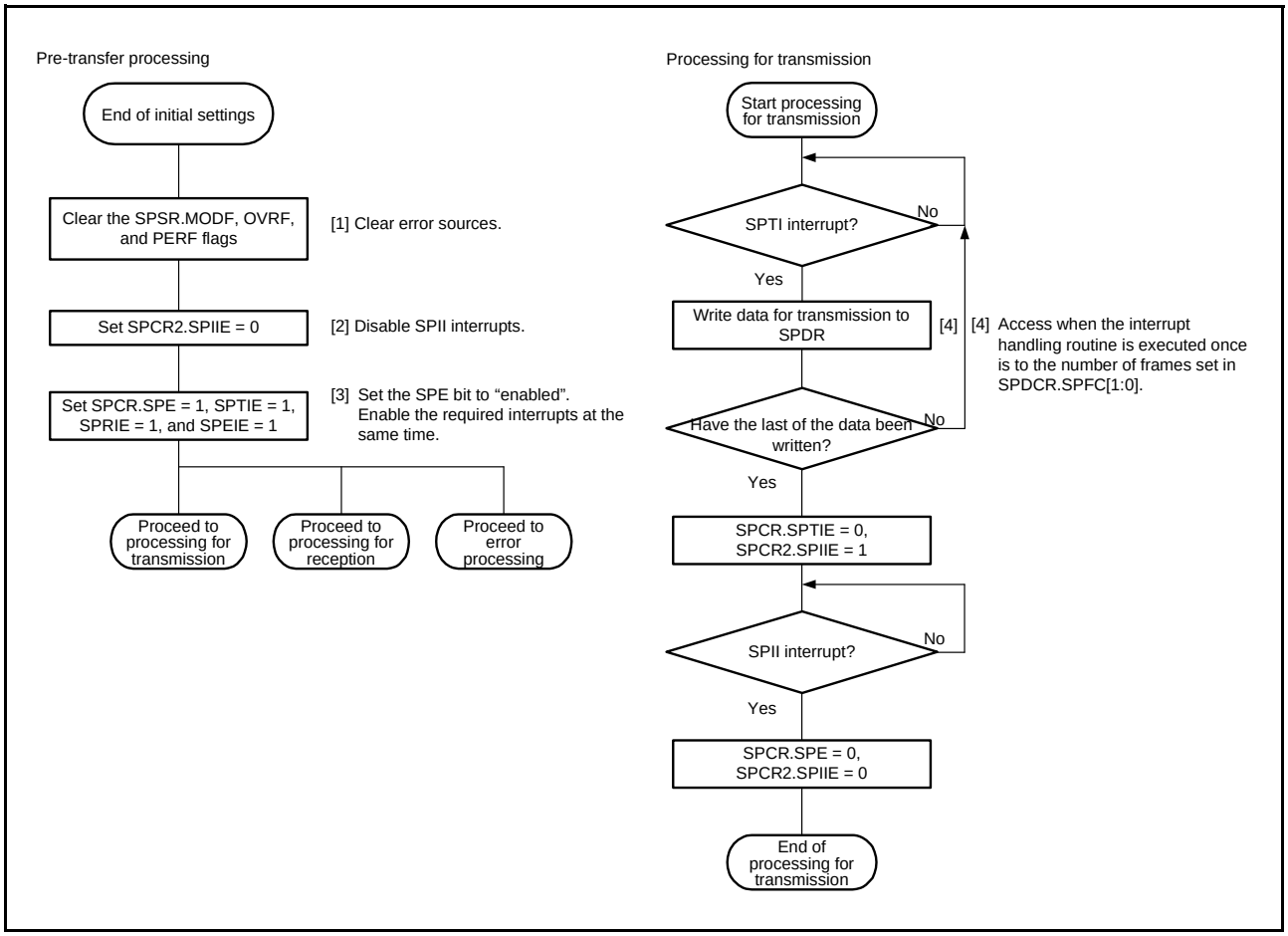


Figure 34.34 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

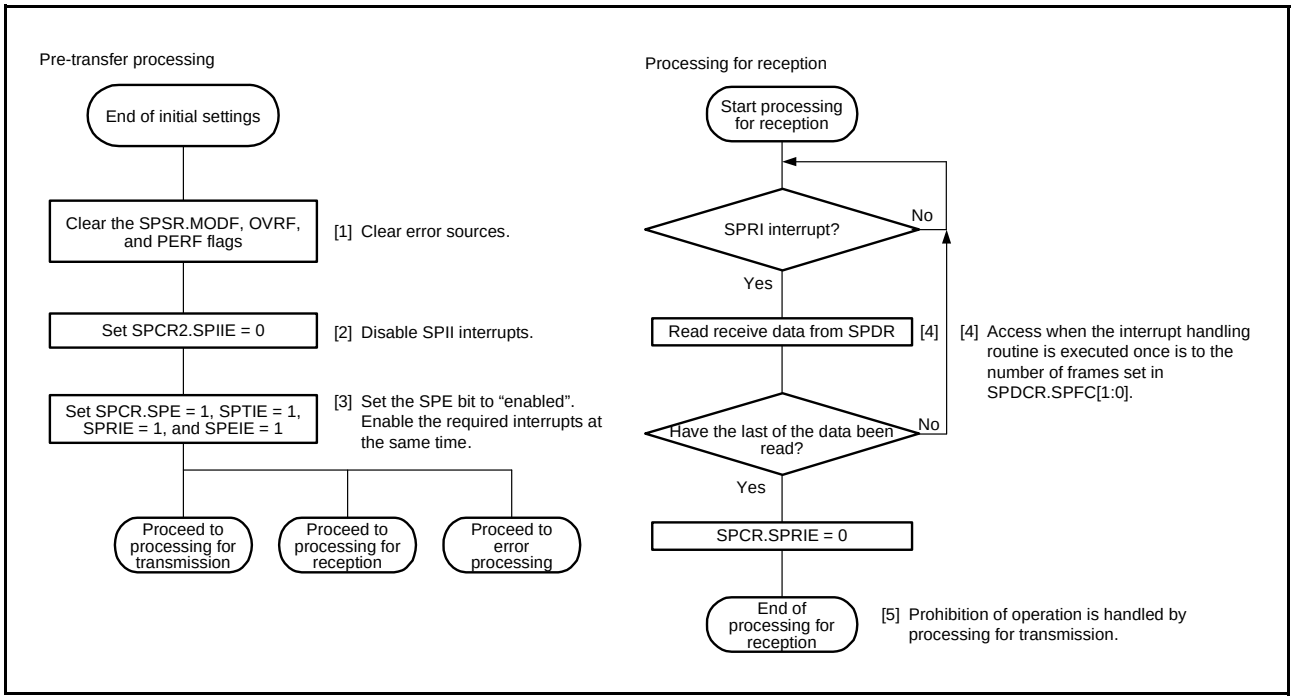


Figure 34.35 Flowchart in Master Mode (Reception)

(c) Flow of Error Processing

The RSPI has three types of error. 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 an error occurs, clear the ICU.IRn.IR flag in the error processing routine. If this is not done, the ICU.IRn.IR flag may continue to indicate the SPTI interrupt or SPRI interrupt request. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

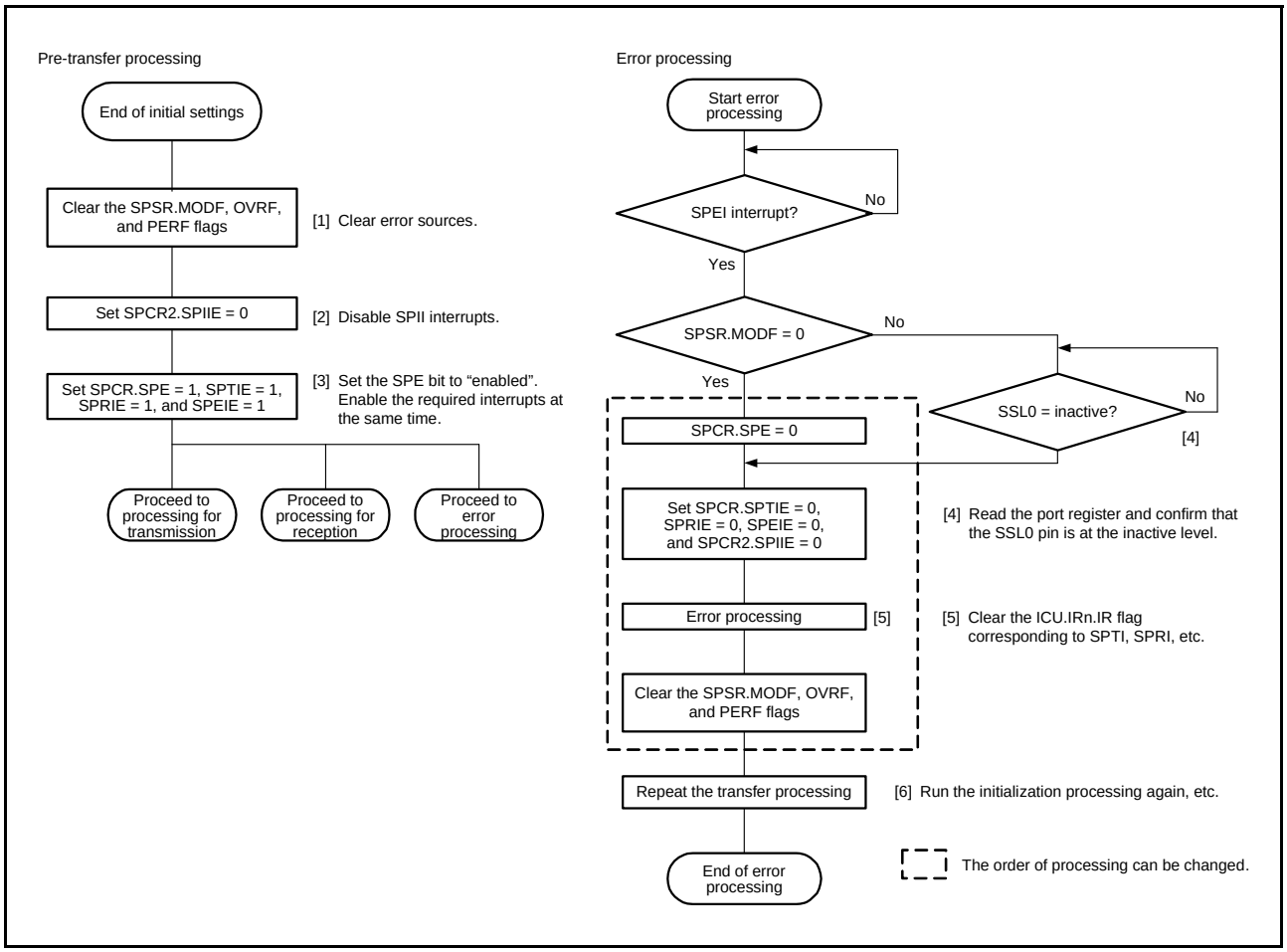


Figure 34.36 Flowchart for Master Mode (Error Processing)

34.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion, the RSPI needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCKn edge in an SSLn0 signal asserted condition, the RSPI needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.

Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOn output signal is the SSLn0 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 34.3.5, Transfer Format. The polarity of the SSLn0 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 RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the 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 SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 34.3.8, 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 SSLn0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, refer to section 34.3.5, 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 SSLn0 input signal. In the type of configuration shown in Figure 34.7 as an example, if the RSPI is used in single-slave mode, the SSLn0 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 SSLn0 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 SSLn0 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 SSLn0 input signal. If the CPHA bit is 1, the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state corresponds to a serial transfer period. Even when the SSLn0 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 34.37 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.

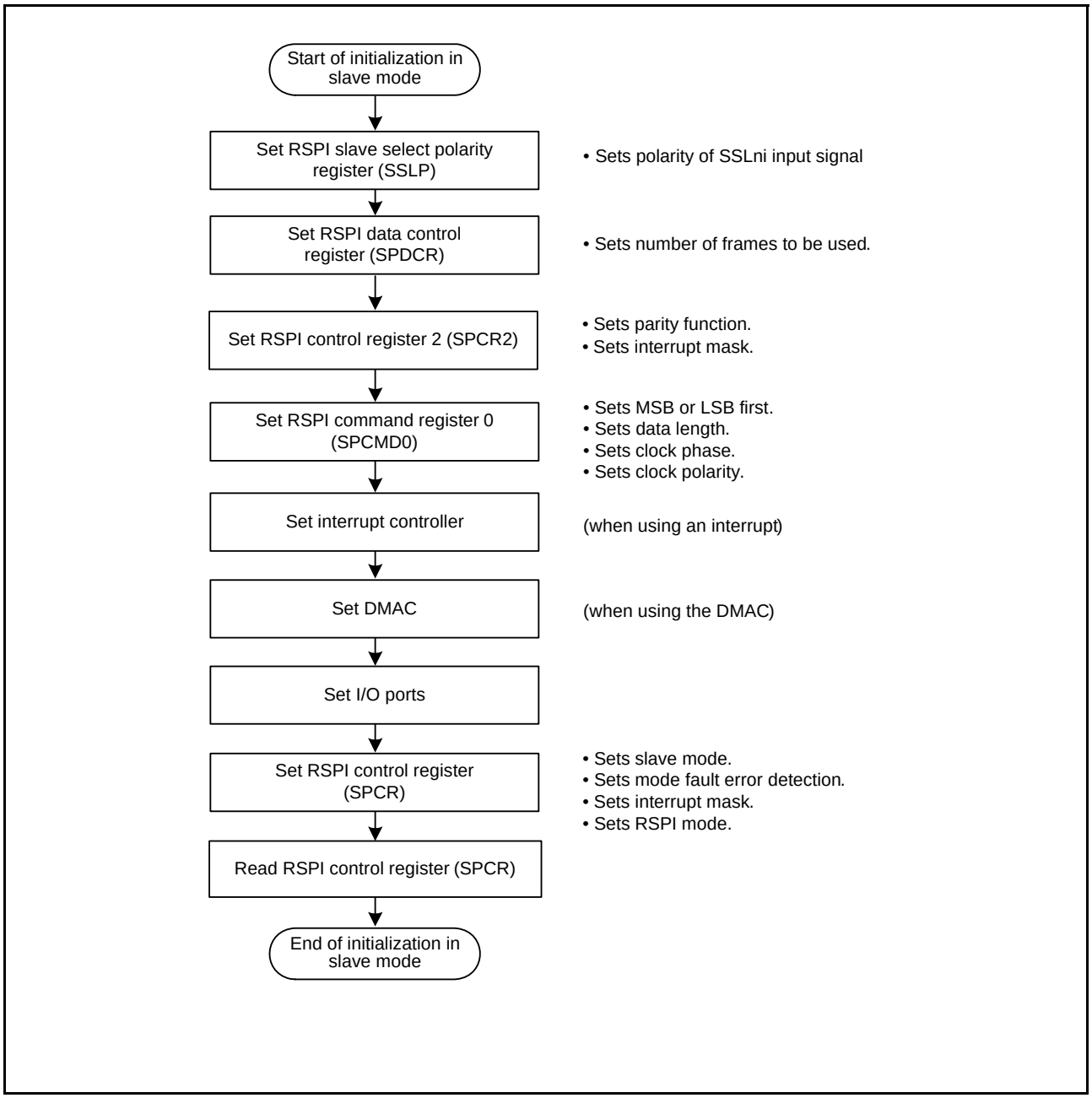


Figure 34.37 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 34.38 to Figure 34.40 show examples of the flow of software processing.

(a) Transmit Processing Flow

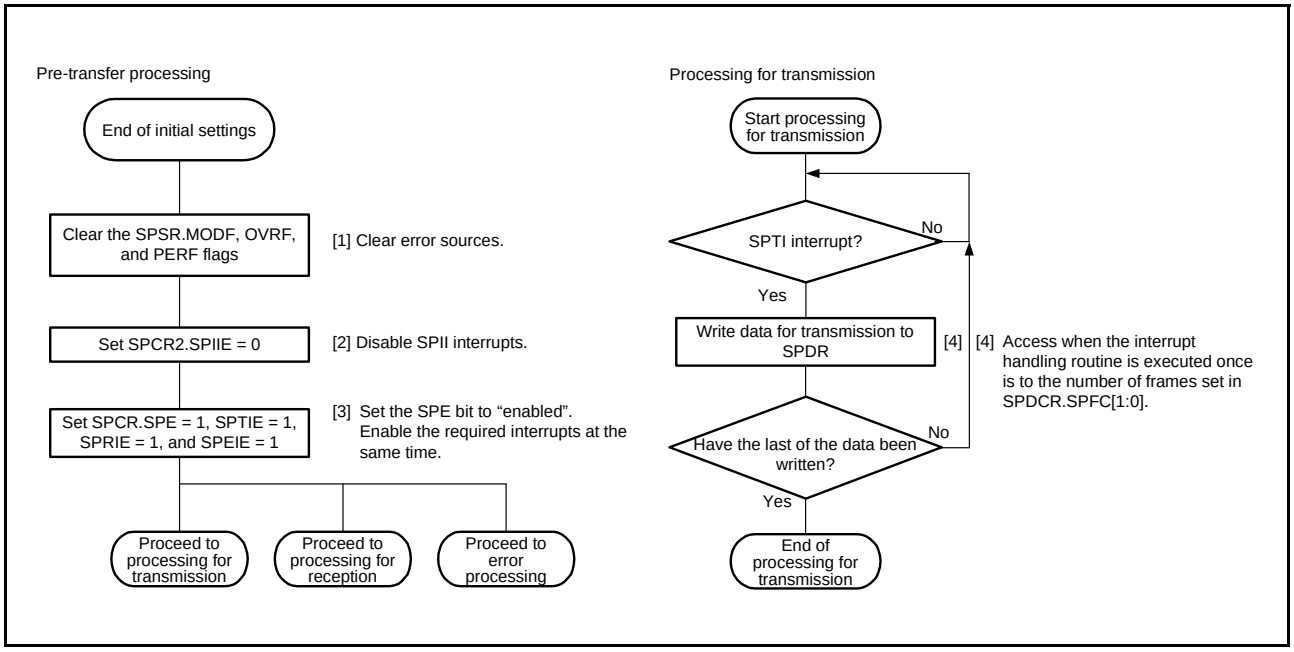


Figure 34.38 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

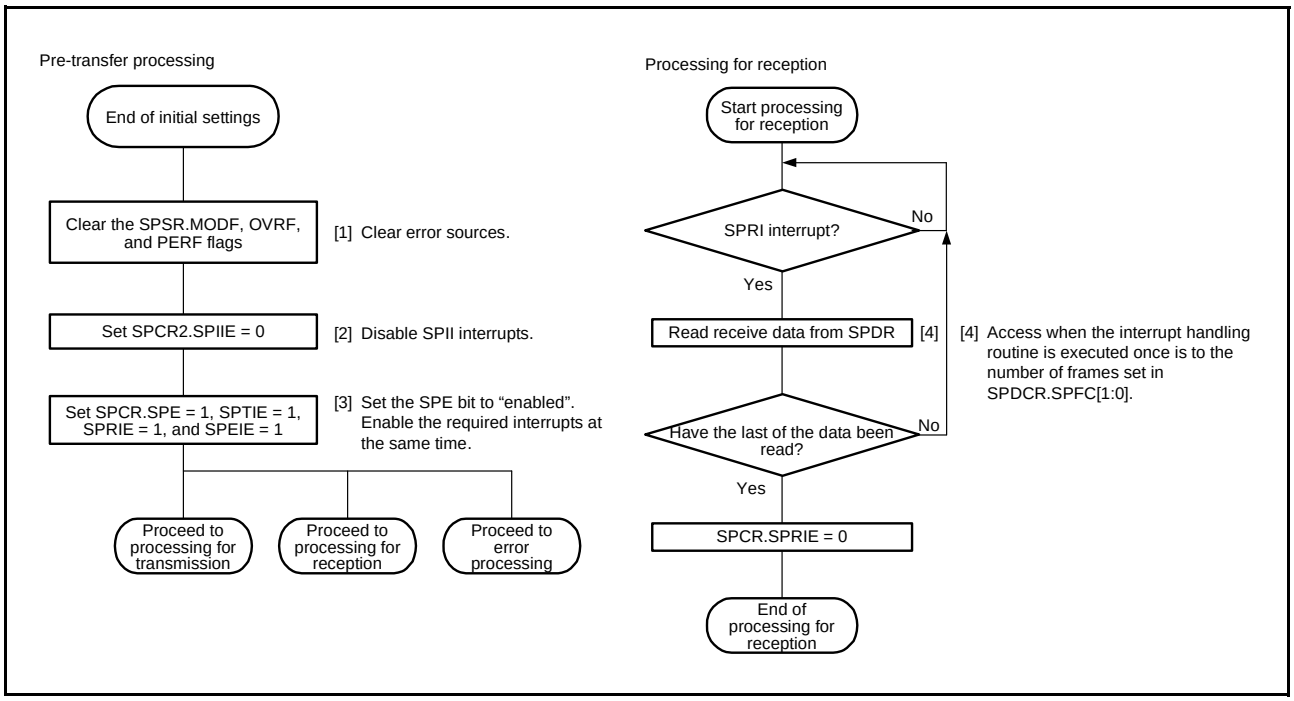


Figure 34.39 Flowchart in Slave Mode (Reception)

(c) Flow of Error Processing

In slave operation, even when a mode-fault error is generated, the SPSR.MODF flag can be cleared without de-asserting the pin.

When an error occurs, clear the ICU.IRn.IR flag in the error processing routine. If this is not done, the ICU.IRn.IR flag may continue to indicate the SPTI interrupt or SPRI interrupt request. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

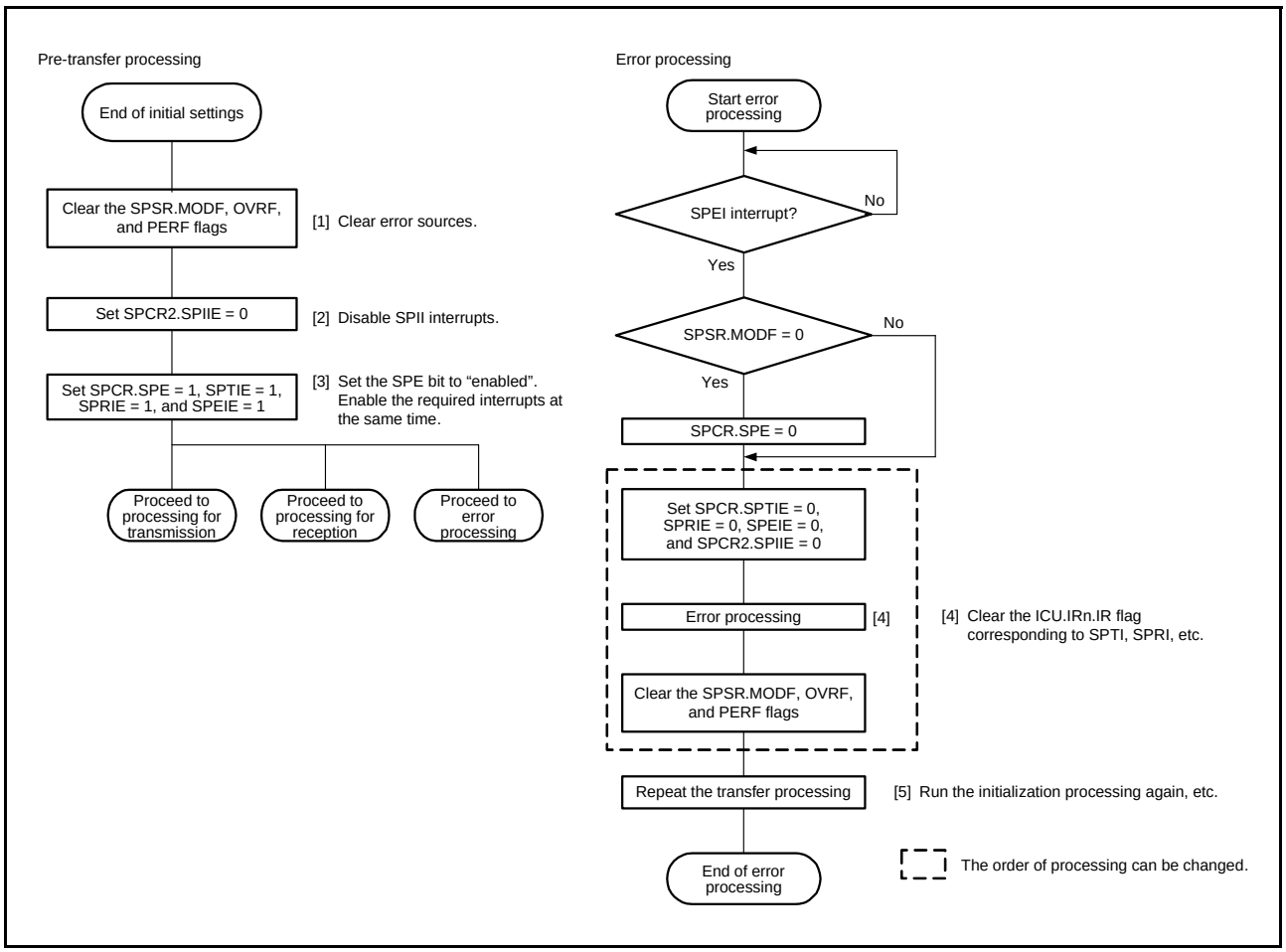


Figure 34.40 Flowchart for Slave Mode (Error Processing)

34.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLni pin is not used, and the three pins of RSPCKn, MOSIn, and MISOn handle communications. The SSLni pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLni pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLni pin is not used.
Furthermore, operation should not be performed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

34.3.11.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR 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 after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmission 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 34.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLn0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer, upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLn0 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 SSLni signals are not output in clock synchronous operation, these settings are valid.
SPSCR 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 SPCMDm register: SSLni output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKn polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of 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 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 SPCMD0 register, and in this manner the sequence is executed repeatedly.

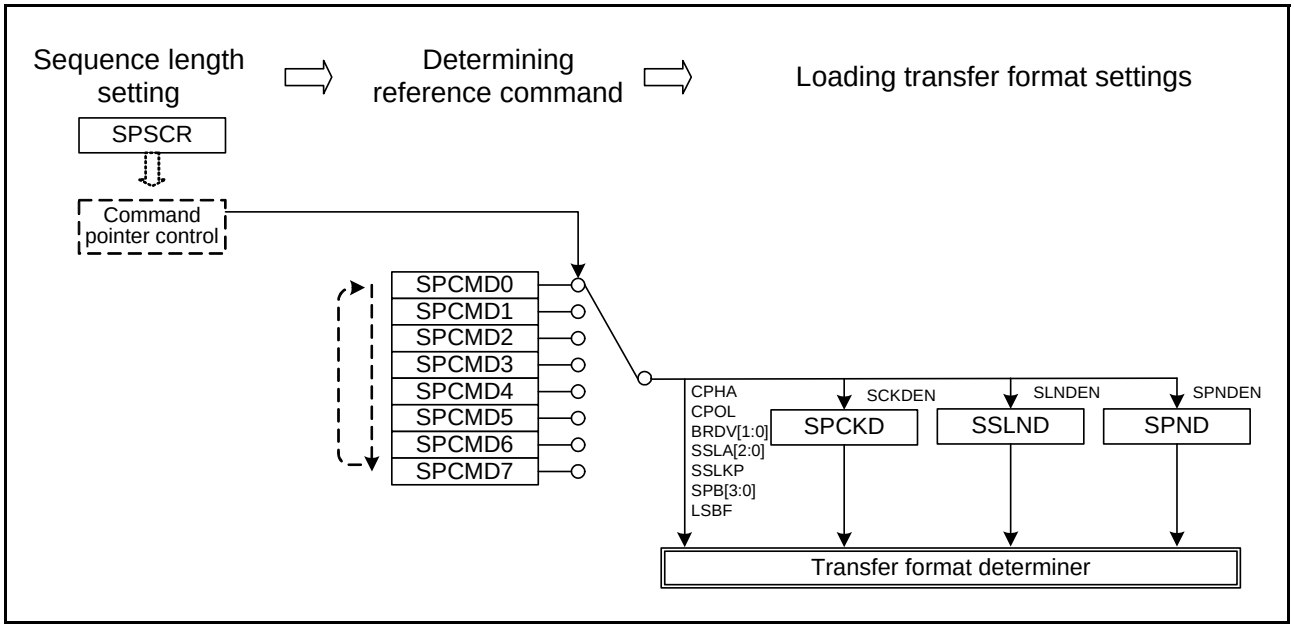


Figure 34.41 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

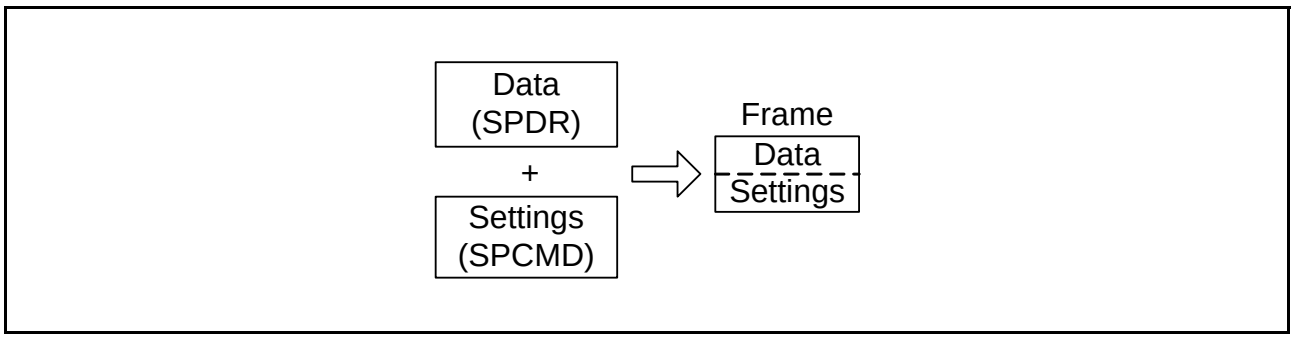


Figure 34.42 Concept of a Frame

Figure 34.43 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 34.4.

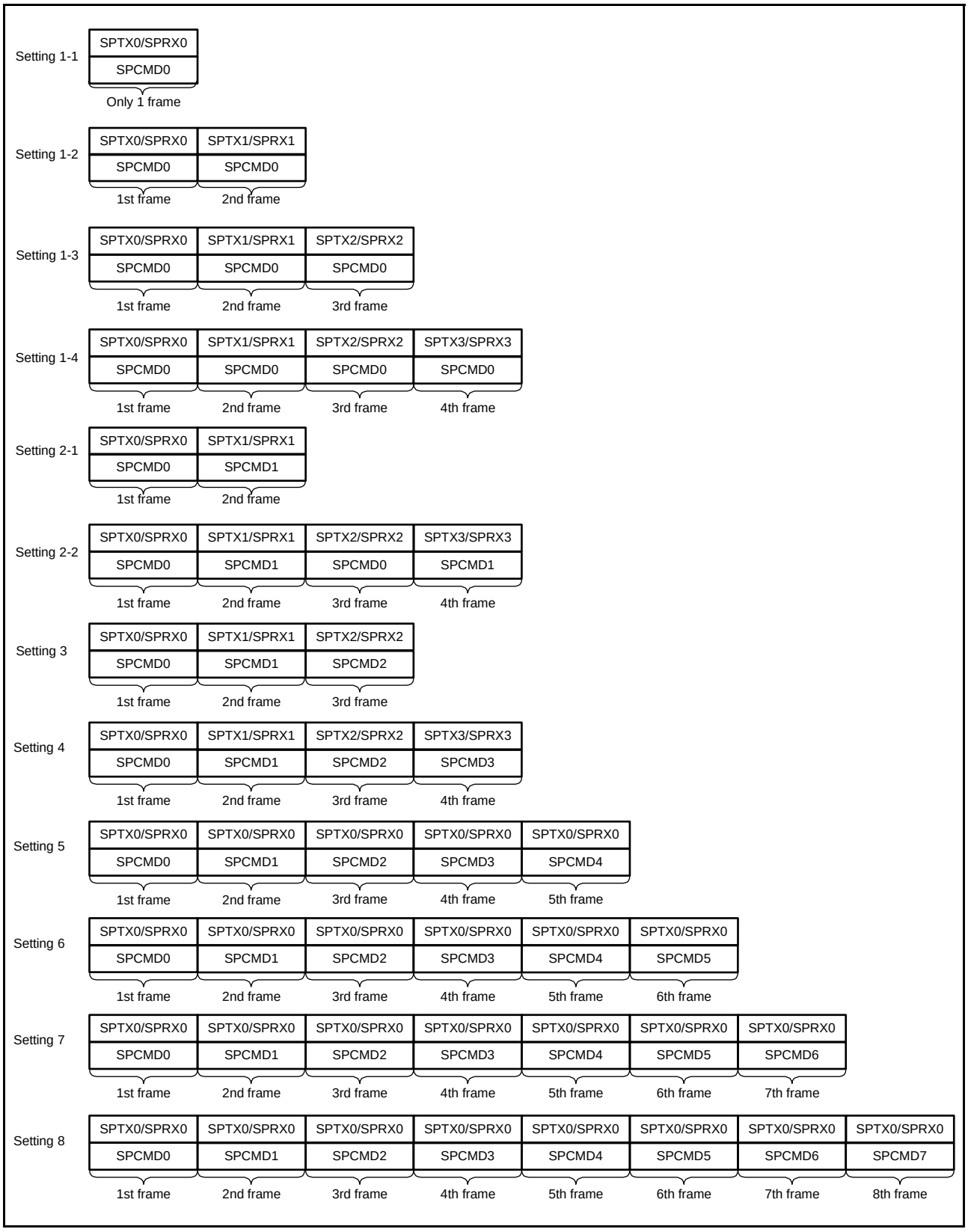


Figure 34.43 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 34.44 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.

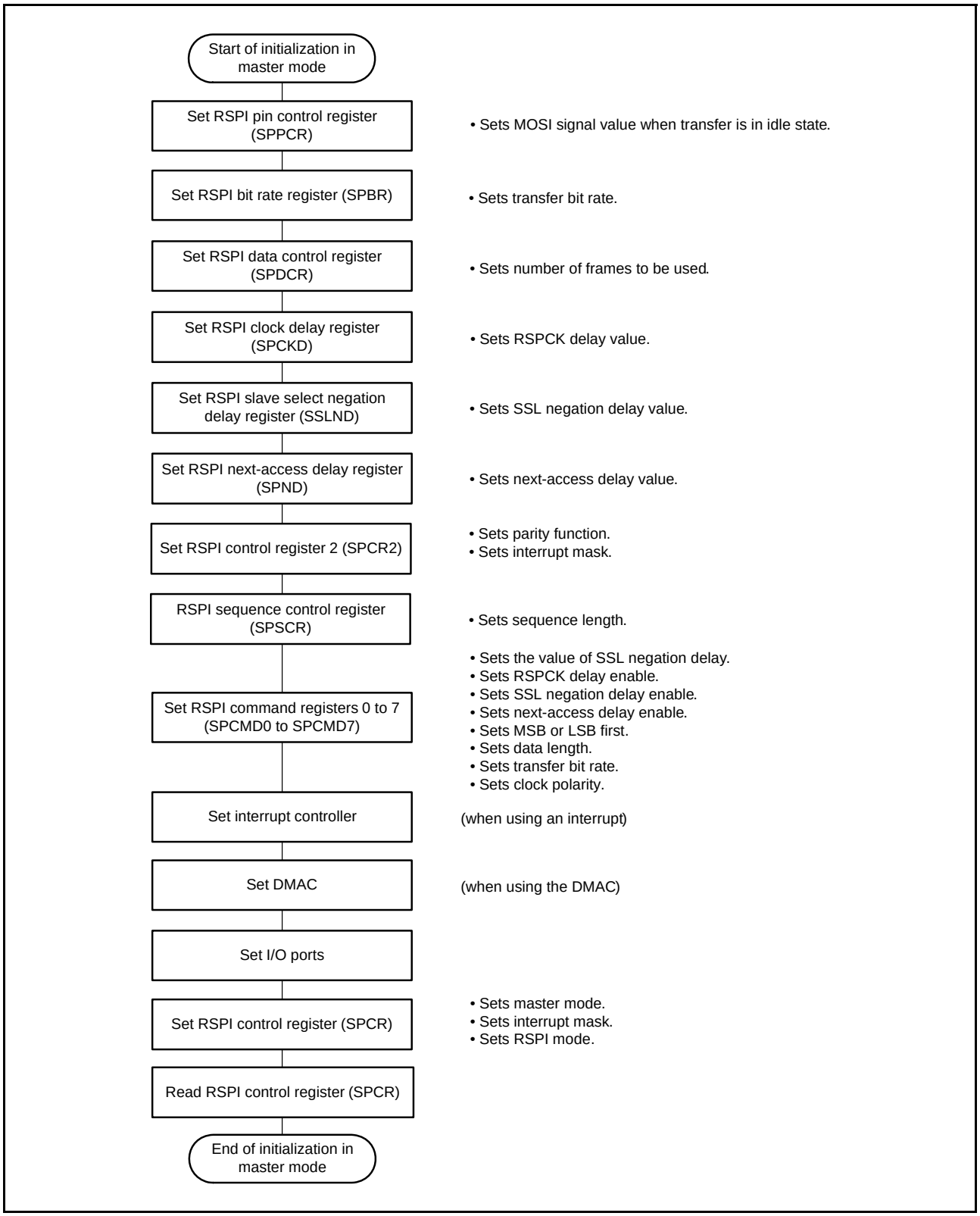


Figure 34.44 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 34.3.10.1, (9) Software Processing Flow. Note that mode-fault errors will not occur.

34.3.11.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI drives the MISOn output signal.
For details on the RSPI transfer format, refer to section 34.3.5, Transfer Format.
It should be noted that the SSL0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing.
When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”. 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 34.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 34.45 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.

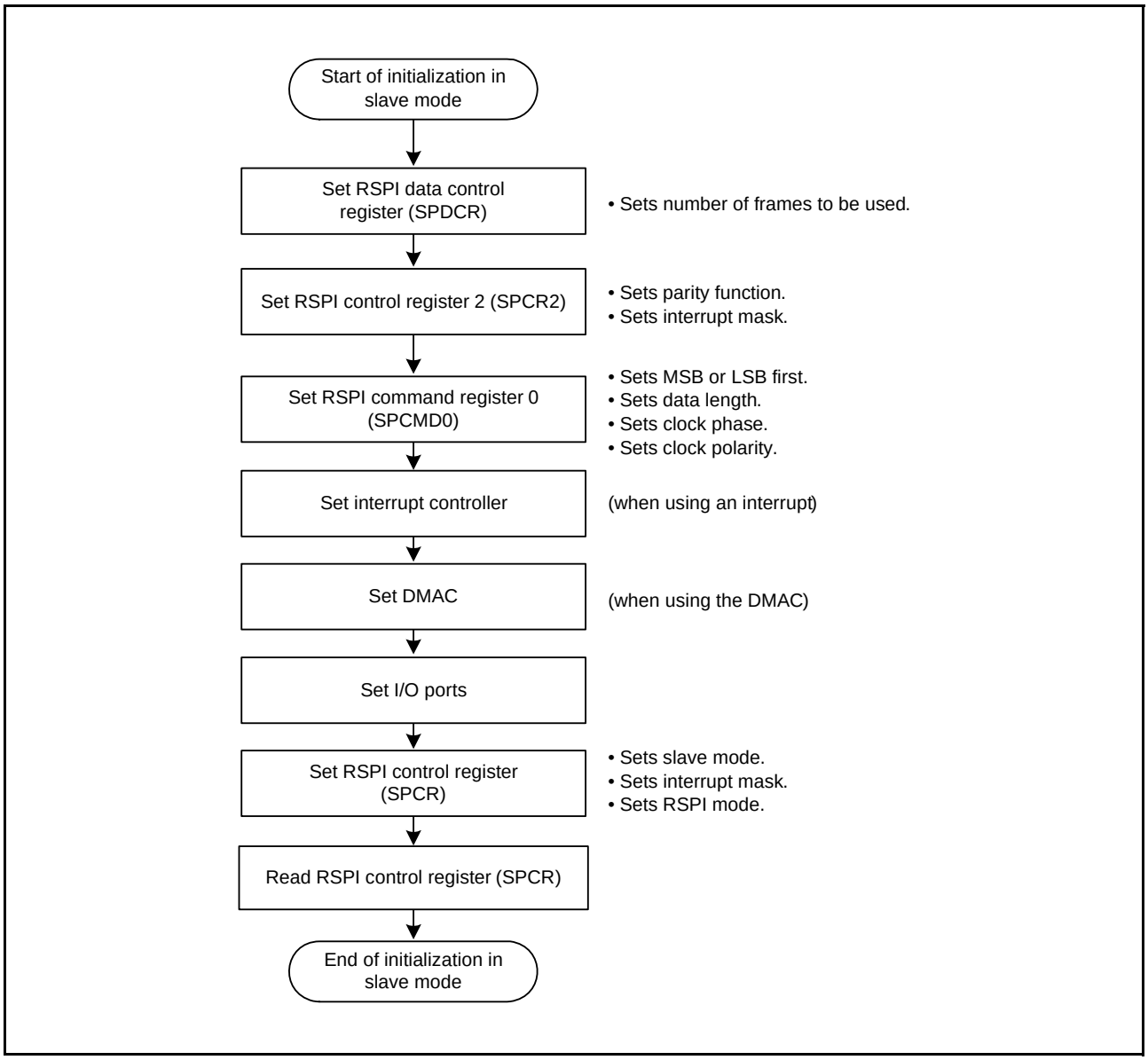


Figure 34.45 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 34.3.10.2, (6) Software Processing Flow. Note that mode-fault errors will not occur.

34.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn 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 MOSIn pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOn 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 34.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 34.46 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 34.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

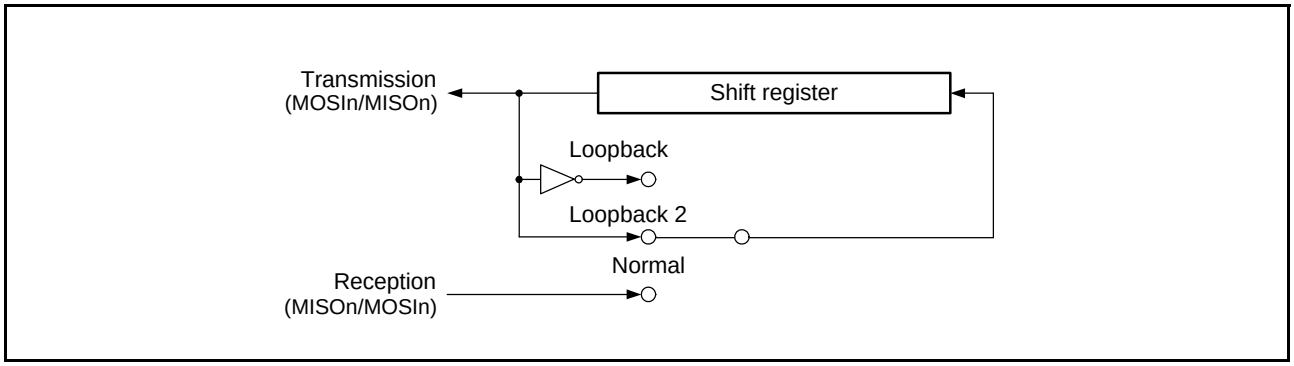


Figure 34.46 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

34.3.13 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 34.47.

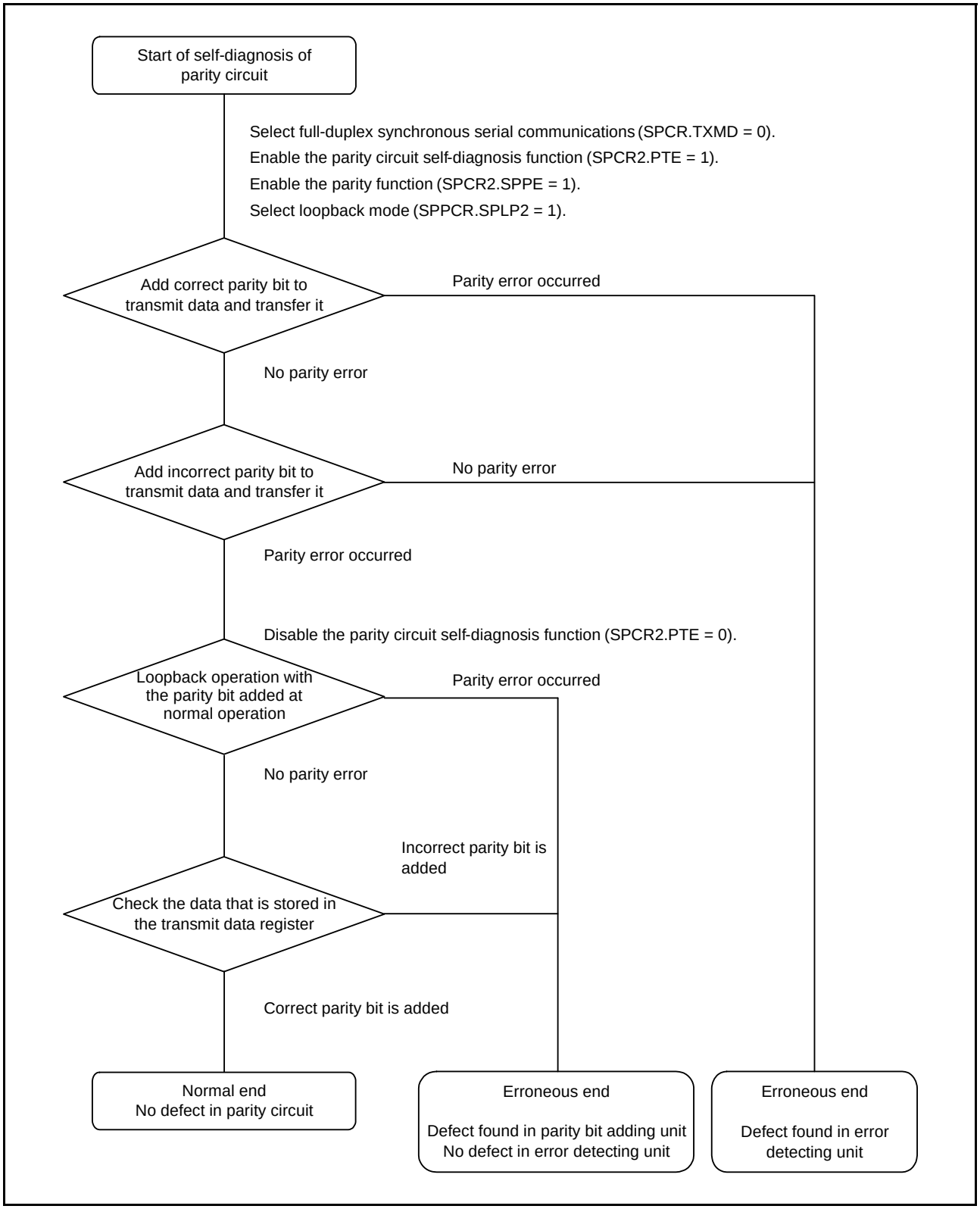


Figure 34.47 Flowchart for Self-Diagnosis of Parity Circuit

34.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. 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, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 34.13. An interrupt is generated on satisfaction of an interrupt condition in Table 34.13. 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 RSPI settings. For the method for setting the DTC or DMAC, refer to section 17, DMA Controller (DMACA), or section 18, Data Transfer Controller (DTCa).

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 34.13 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty while the SPCR.SPTIE bit is 1.	Possible
RSPI errors (mode fault, overrun and parity error)	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
RSPI idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1.	Impossible

34.4 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.

34.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR on completion of serial transfer.

34.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmission buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

34.4.3 Mode Fault, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 34.14 lists the occurrence conditions of a mode fault event.

Table 34.14 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLn0 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) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the reception 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.

(3) 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.

34.4.4 RSPI Idle Event Output

(1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (RSPI 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).

34.4.5 Transmission-Completed 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 (RSPI idle flag) from 1 to 0.

Table 34.15 Conditions for Generation of a Transmission-Completed Event (Slave)

	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSL0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCK

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode fault error.

34.5 Usage Notes

34.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can used to enable or disable operation of the RSPI. The RSPI is stopped after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

34.5.2 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.

34.5.3 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.

35. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

35.1 Overview

Table 35.1 lists the specifications of the CRC calculator, and Figure 35.1 shows a block diagram of the CRC calculator.

Table 35.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	Operation executed on 8 bits in parallel
CRC generating polynomial	One of three generating polynomials selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have functionality to divide data for calculation into CRC calculation units. Write data in 8-bit units.

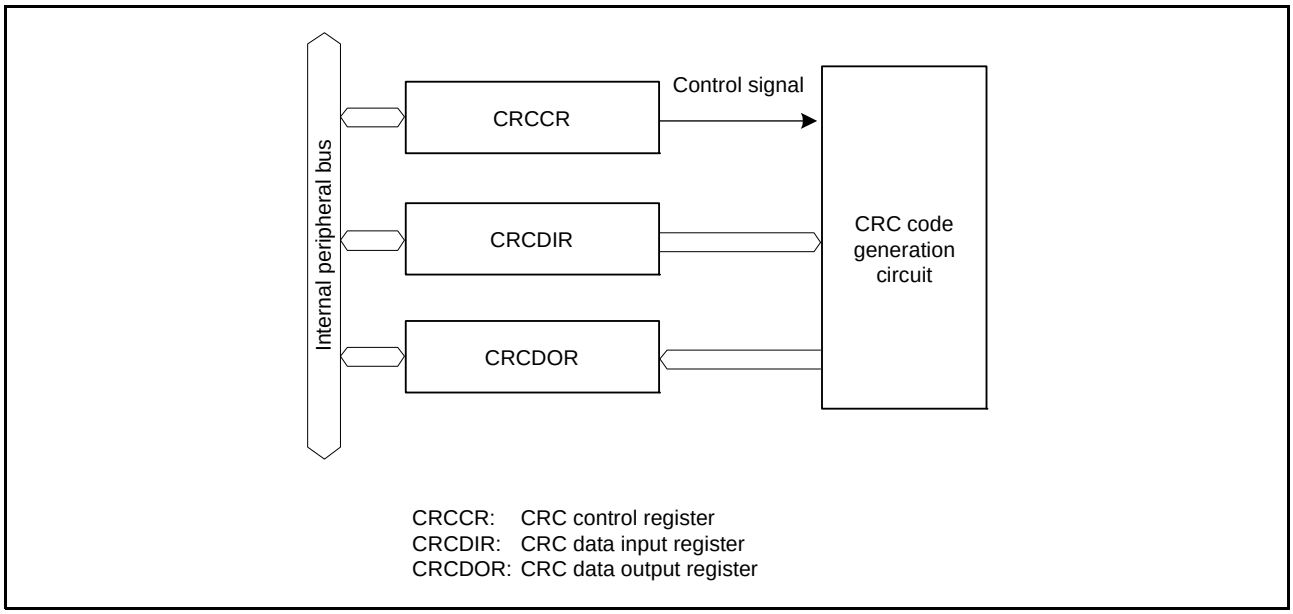


Figure 35.1 CRC Block Diagram

35.2 Register Descriptions

35.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

Set this bit to select the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on transmitting and receiving CRC code, refer to section 35.3, Operation.

35.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

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

CRCDIR is a readable/writable register. Write data for CRC calculation to this register.

35.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

	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

CRCDOR is a readable/writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

35.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
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 before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of CRCDOR.

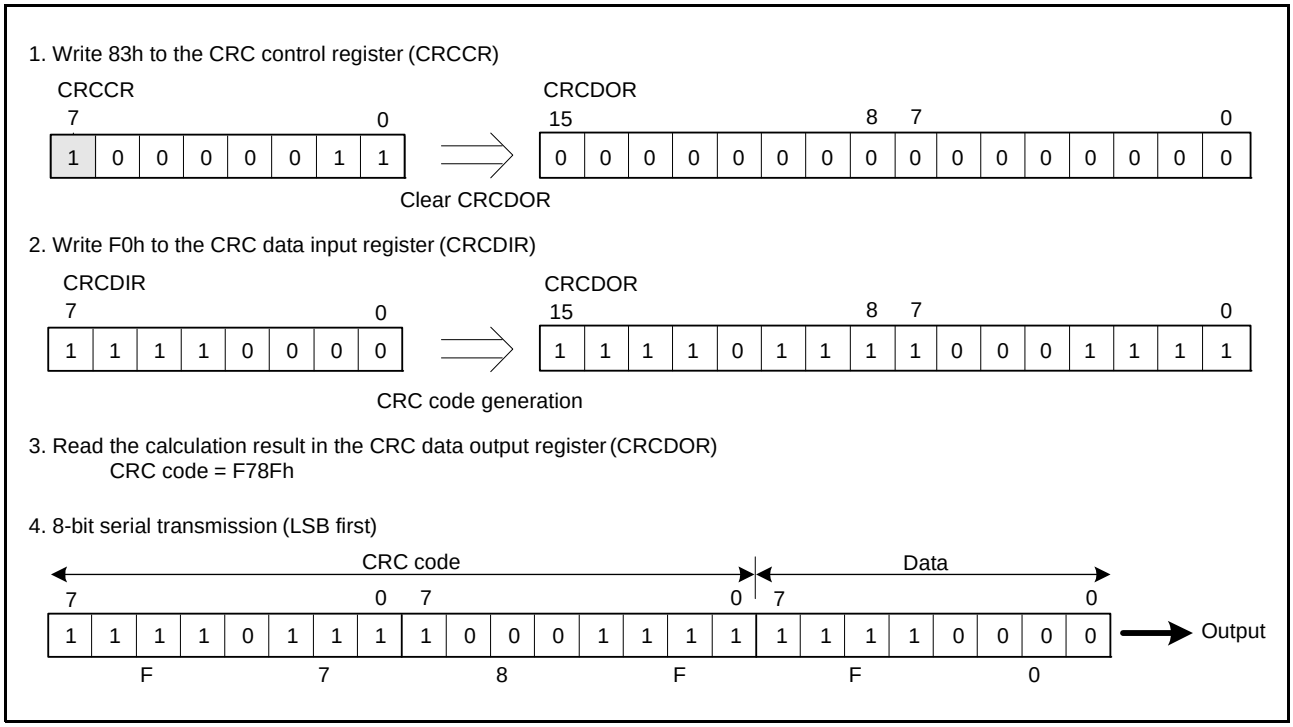


Figure 35.2 LSB First Data Transmission

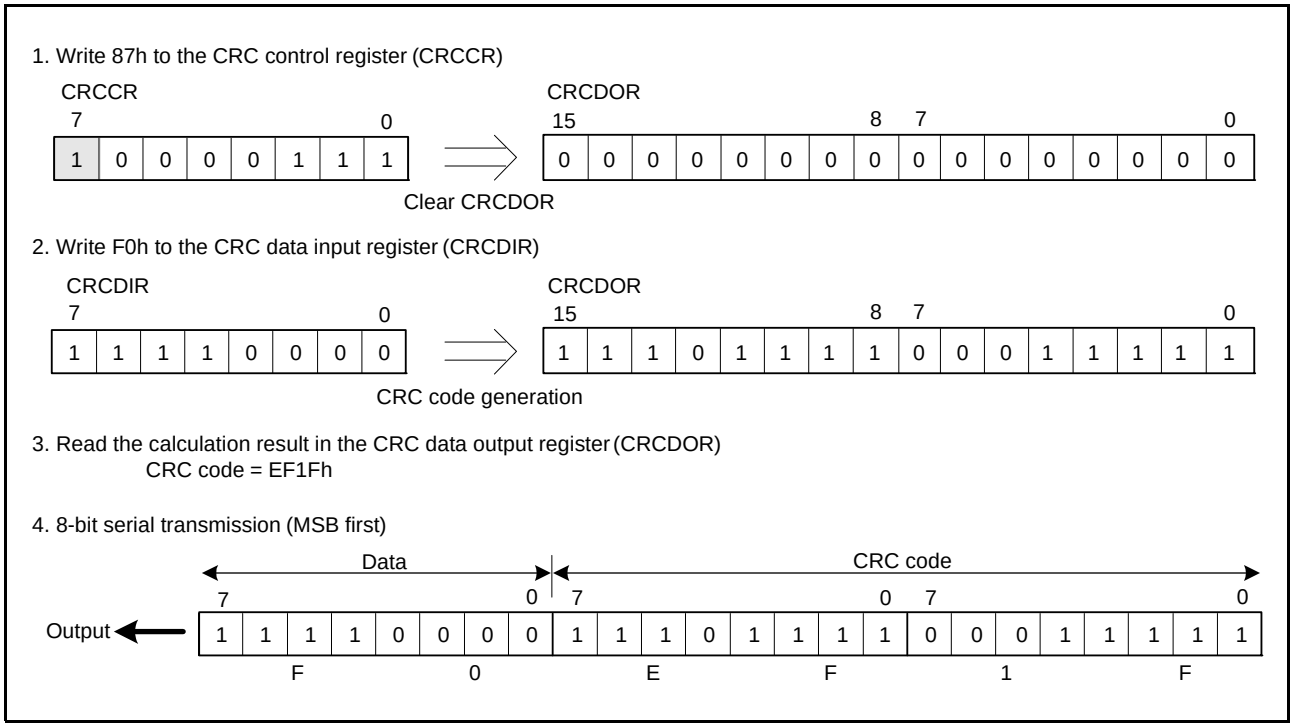


Figure 35.3 MSB First Data Transmission

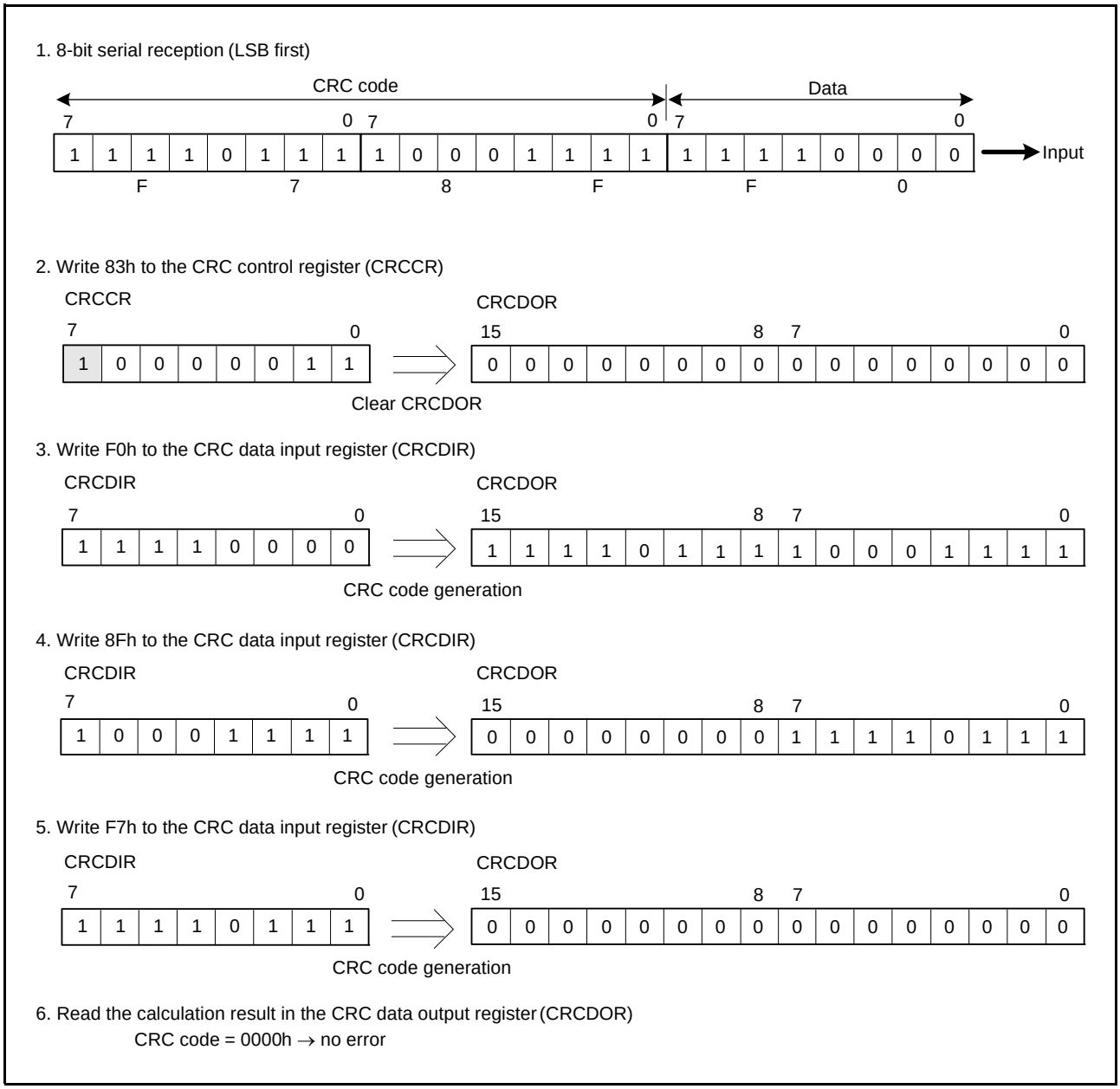
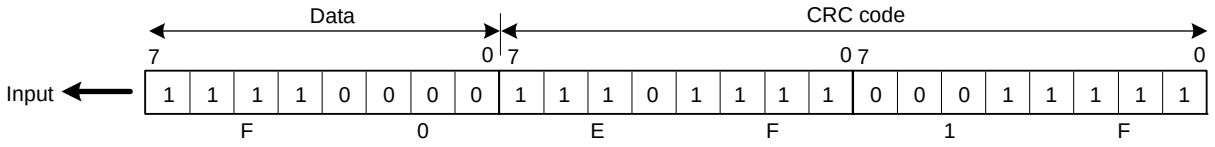
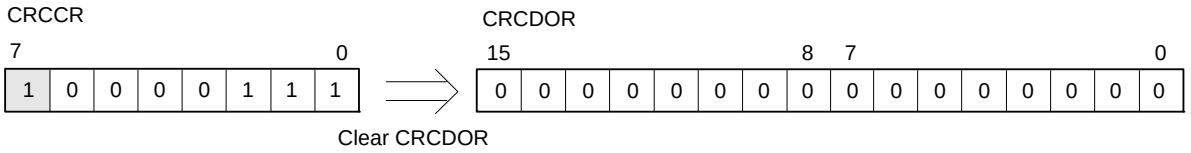


Figure 35.4 LSB First Data Reception

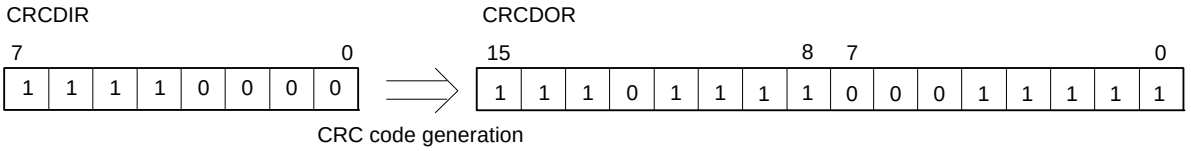
1. 8-bit serial reception (MSB first)



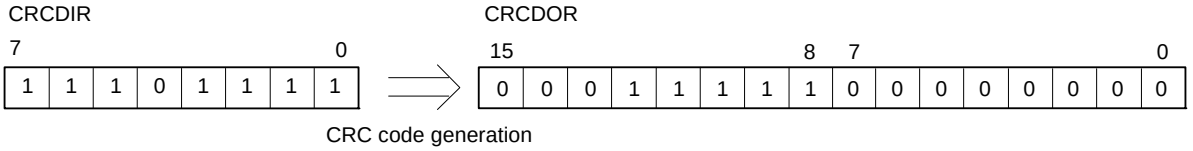
2. Write 87h to the CRC control register (CRCCR)



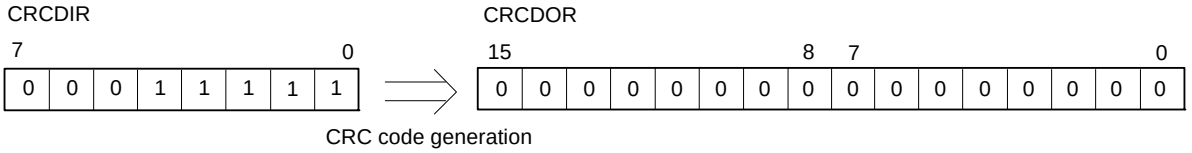
3. Write F0h to the CRC data input register (CRCDIR)



4. Write EFh to the CRC data input register (CRCDIR)



5. Write 1Fh to the CRC data input register (CRCDIR)



6. Read the calculation result in the CRC data output register (CRCDOR)

CRC code = 0000h → no error

Figure 35.5 MSB First Data Reception

35.4 Usage Notes

35.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using 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.

35.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

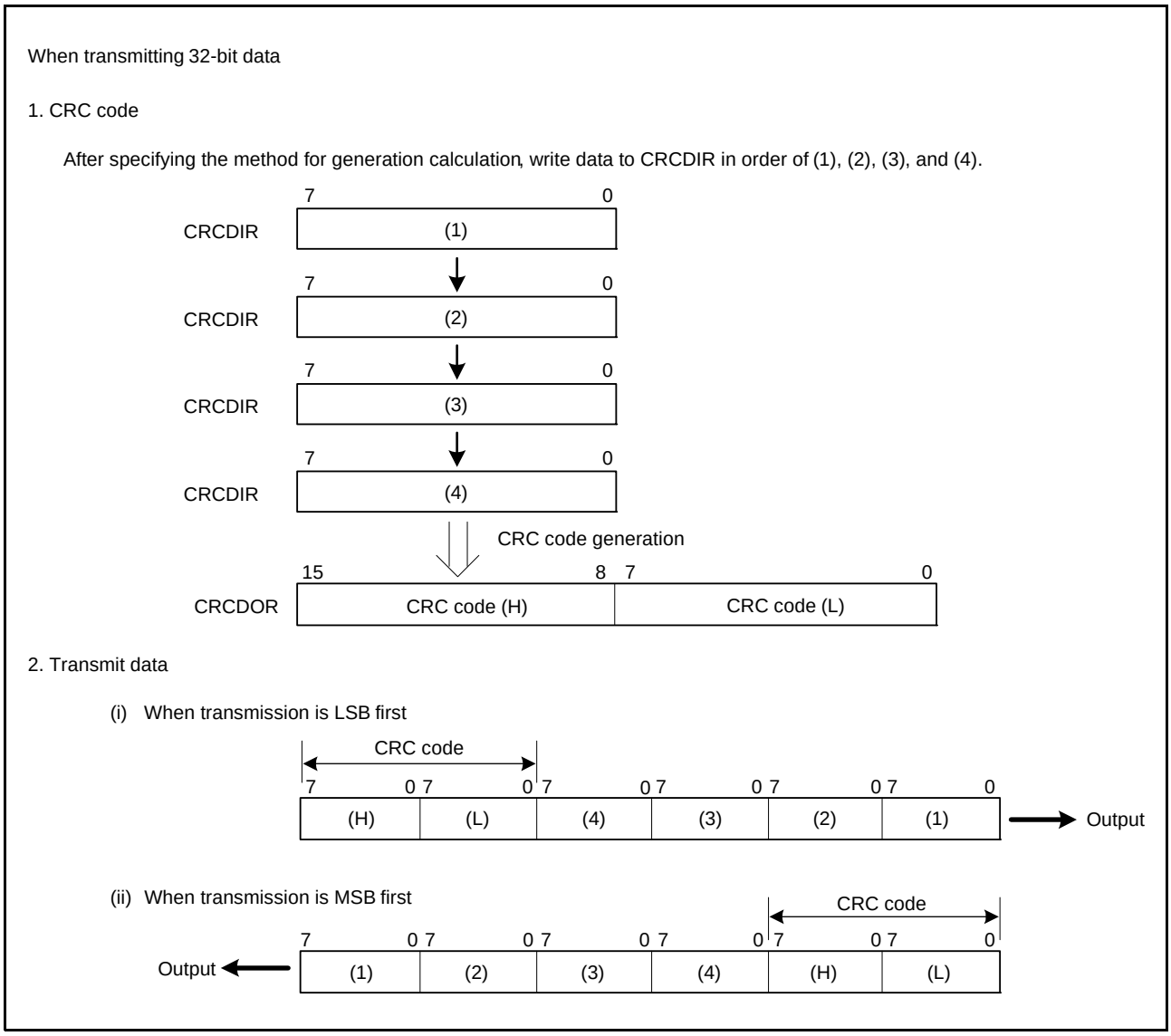


Figure 35.6 LSB First and MSB First Data Transmission

36. 12-Bit A/D Converter (S12ADB)

36.1 Overview

This MCU includes a 12-bit successive approximation A/D converter. Up to 16 channel analog inputs or internal reference voltages can be selected.

The 12-bit A/D converter converts a maximum of 16 selected channels of analog inputs or internal reference voltages into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 16 arbitrarily selected channels are converted for only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 16 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 16 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.

In group scan mode, the scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.

In double trigger mode, one arbitrarily selected analog input channel 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 triggers are stored into separate 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.

A/D conversion of the internal reference voltage is accomplished independently.

Table 36.1 lists the specifications of the 12-bit A/D converter and Table 36.2 indicates the functions of the 12-bit A/D converter. Figure 36.1 shows a block diagram of the 12-bit A/D converter.

Table 36.1 Specifications of 12-Bit A/D Converter (1 / 2)

Item	Description
Number of units	One unit
Input channels	Up to 16 channels
Extended analog inputs	Internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1.0 µs per channel (when A/D conversion clock ADCLK = 50 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock ADCLK*1 can be set so that the frequency division ratio should be one of the following. PCLKB to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8, 2:1, 4:1 ADCLK is set using the clock generation circuits.
Data registers	For analog input: 16 data registers For duplication of A/D conversion data in double trigger mode: One data register For internal reference voltage: One data register The A/D conversion result is stored in 12-bit A/D data registers. • 8, 10, and 12-bit accuracy output for the results of A/D conversion (selectable between 2 and 4-bit right shifts for output of conversion results). In addition mode, A/D conversion results are added and stored in A/D data registers as 14-bit data. Duplication of A/D conversion data • A/D conversion data of one selected analog input channel is stored into A/D data register y when conversion is started by the first trigger and into the duplication register when started by the second trigger. • Duplication is available only in double trigger mode in single scan mode or group scan mode.

Table 36.1 Specifications of 12-Bit A/D Converter (2 / 2)

Item	Description
Operating modes	• Single scan mode: A/D conversion is performed for only once on the analog inputs of up to 16 arbitrarily selected channels. A/D conversion is performed only once on the internal reference voltage. • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 16 arbitrarily selected channels.*2 • Group scan mode: Up to 16 channels of analog inputs are divided into group A and group B and A/D conversion is performed only once on all the selected channels on a group basis. The scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.
Conditions of A/D conversion start	• Software trigger • Synchronous trigger Trigger by MTU, TPU, or ELC • Asynchronous trigger A/D conversion can be triggered from the ADTRG0# pin.
Function	• Sample-and-hold function • Channel-dedicated sample-and-hold function • Variable sampling state count • Self-diagnosis of 12-bit A/D converter • A/D-converted value addition mode • Analog input disconnection detection assist • Double trigger mode (duplication of A/D conversion data)
Interrupt source	• In the modes except double trigger mode and group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of single scan. • In double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan. • In group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of group A scan, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan. • In group scan mode with double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan of group A, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan. • S12ADI0 or GBADI interrupt can activate DMA controller (DMAC) or data transfer controller (DTC).
Event linking	• An ELC event can be generated on completion of scans except for group B scan in group scan mode. • A/D conversion can be started by the trigger from ELC.
Low power consumption function	• Module stop state can be specified.*3

Note 1. Peripheral module clock PCLKB is set according to the setting of SCKCR.PCKB[3:0] and A/D conversion clock ADCLK is set according to the setting of SCKCR.PCKD[3:0].
Note 2. Do not use continuous scan mode or group scan mode when the internal reference voltage is selected.
Note 3. For details, refer to section 11, Low Power Consumption.

Table 36.2 Functions of 12-Bit A/D Converter

Item			Function
Analog input channel			AN000 to AN015, internal reference voltage
A/D conversion start conditions	Software	Software trigger	Enabled
	Asynchronous trigger	ADTRG0#	Enabled
	Synchronous trigger	TRGA compare match/input capture from MTU0	TRG0AN
		TRGB compare match/input capture from MTU0	TRG0BN
		TRGA compare match/input capture or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	TRGAN
		TRGE compare match from MTU0	TRG0EN
		TRGF compare match from MTU0	TRG0FN
		MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4AN
		MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4BN
		MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4ABN
		TRGA compare match/input capture from TPU0 to TPU4 or TRGA compare match/input capture from TPU0	TRGAN1 TRG4ABN1
		Trigger from ELC	Enabled
Interrupt			S12ADI0 interrupt, GBADI interrupt
Module stop function setting*1			MSTPCRA.MSTPA17 bit

Note 1. For details, see section 11, Low Power Consumption.

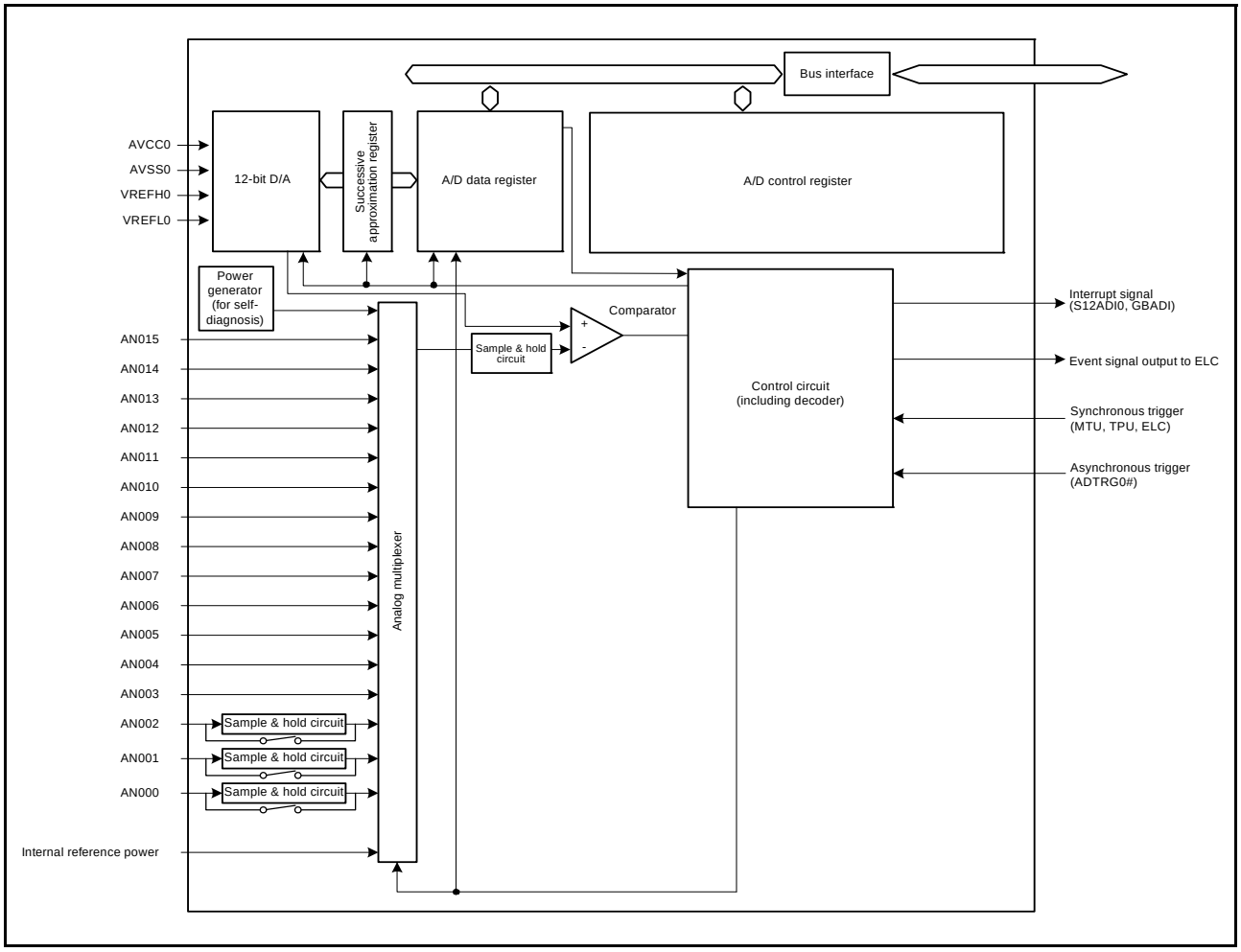


Figure 36.1 Block Diagram of 12-Bit A/D Converter

Table 36.3 indicates the input pins of the 12-bit A/D converter.

Table 36.3 Input Pins of 12-Bit A/D Converter

Pin Name	Input	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Reference power supply pin
VREFL0	Input	Reference power supply ground pin
AN000 to AN015	Input	Analog input pins (AN000, AN001, and AN002 have channel-dedicated sample-and-hold function)
ADTRG0#	Input	External trigger input pin for starting A/D conversion

36.2 Register Descriptions

36.2.1 A/D Data Registers y (ADDRy; y = 0 to 15), A/D Data-Doubling Register (ADDBLDR)

The ADDRy registers are 16-bit read-only registers for storing the results of A/D conversion. Register ADDBLDR is a 16-bit read-only register for storing the result of A/D conversion in response to the second trigger in double-trigger mode.

The formats for data in the ADDRy, ADDBLDR registers vary according to the following conditions.

- The setting of the A/D data register format selection bit (determining whether the data are flush-left or flush-right in the registers)
- The setting of the A/D data register bit-precision specification bits (for eight, 10, or 12 bits)
- The setting of the A/D-converted value cumulative mode selection register (determining whether A/D-converted value cumulative mode is or is not selected)

The conditions are given above each of the descriptions below.

- The settings are for flush-right data with 12-bit precision

Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
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
b11 to b0	AD[11:0]	Converted value 11 to 0	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings are for flush-right data with 10-bit precision

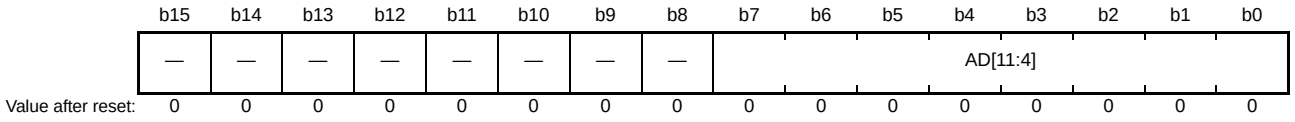
Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	AD[11:2]									
Value after 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	AD[11:2]	Converted value 11 to 2	10 higher-order bits of the 12-bit A/D converted value	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings are for flush-right data with eight-bit precision

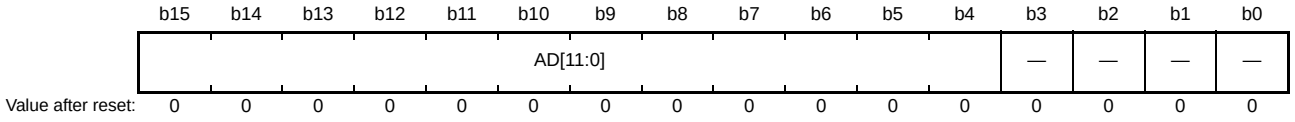
Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	AD[11:4]	Converted value 11 to 4	Eight higher-order bits of the 12-bit A/D converted value	R
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings are for flush-left data with 12-bit precision

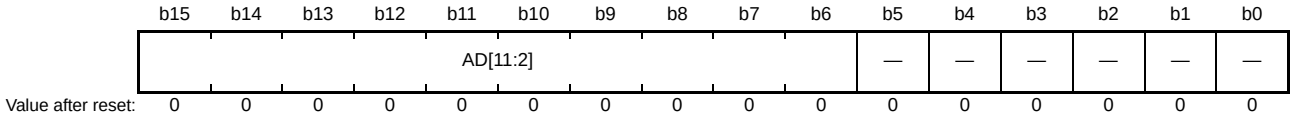
Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h



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
b15 to b4	AD[11:0]	Converted value 11 to 0	12-bit A/D-converted value	R

- The settings are for flush-left data with 10-bit precision

Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h



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
b15 to b6	AD[11:2]	Converted value 11 to 2	10 higher-order bits of the 12-bit A/D converted value	R

- The settings are for flush-left data with eight-bit precision

Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:4]								—	—	—	—	—	—	—	—
Value after 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
b15 to b8	AD[11:4]	Converted value 11 to 4	Eight higher-order bits of the 12-bit A/D converted value	R

- When A/D-converted value addition mode is selected

Address: ADDR0 0008 9020h to ADDR15 0008 903Eh, ADDBLDR 0008 9018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[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
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the AD[13:0] bits show the value that is obtained by adding up A/D-converted values on a specific channel. In A/D-converted value addition mode, the value obtained by adding up A/D conversion results is stored left-justified as a 14-bit value in the A/D data register. The settings of the ADCER.ADPRC[1:0] and ADCER.ADRFMT bits become invalid.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected.

First conversion: 0000h ≤ ADDRy (y = 0 to 15) ≤ 3FFCh
(ADDRy (y = 0 to 15): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)
Second conversion: 0000h ≤ ADDRy (y = 0 to 15) ≤ 7FF8h
(ADDRy (y = 0 to 15): Bit 15 = 0b, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)
Third conversion: 0000h ≤ ADDRy (y = 0 to 15) ≤ BFF4h
(ADDRy (y = 0 to 15): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)
Fourth conversion: 0000h ≤ ADDRy (y = 0 to 15) ≤ FFF0h
(ADDRy (y = 0 to 15): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

36.2.2 A/D Internal Reference Voltage Data Register (ADOCDR)

ADOCDR is a 16-bit read-only register that holds the A/D conversion results of the internal reference voltage.
The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

Address: 0008 901Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
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
b11 to b0	AD[11:0]	—	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address: 0008 901Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	—	—
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	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address: 0008 901Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[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
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADOCDR show the value added by the A/D-converted value of the internal reference voltage. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes left-aligned.

36.2.3 A/D Self-Diagnosis Data Register (ADRD)

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the A/D converter's self-diagnosis. The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER. ADRD cannot be set to A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

Address: 0008 901Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	AD[11:0]											
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
b11 to b0	AD[11:0]	—	12-bit A/D-converted value	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using 0 V has been executed. 1 0: Self-diagnosis using VREFH0 × 1/2 has been executed. 1 1: Self-diagnosis using VREFH0 has been executed. For details of self-diagnosis, see section 36.2.9, A/D Control Extended Register (ADCER).	R

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address: 0008 901Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	DIAGST[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	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using 0 V has been executed. 1 0: Self-diagnosis using VREFH0 × 1/2 has been executed. 1 1: Self-diagnosis using VREFH0 has been executed. For details of self-diagnosis, see section 36.2.9, A/D Control Extended Register (ADCER).	R
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	—	12-bit A/D-converted value	R

36.2.4 A/D Control Register (ADCSR)

Address: 0008 9000h

	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]	A/D Conversion Data Duplication Channel Select	Select one of 16 analog input channels for A/D conversion data duplication. These bits are valid only in double trigger mode.	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: Disables GBADI interrupt generation upon group B scan completion. 1: Enables GBADI interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: A/D conversion is started by the synchronous trigger (MTU, TPU, or ELC). 1: A/D conversion is started by the asynchronous trigger (ADTRG0#).	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by the synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger.	R/W
b10, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	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

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
If 1 is written to both the TRGE and EXTRG bits in ADCSR when a high-level signal is input to the external pin (ADTRG0#), and then if the ADTRG0# signal is driven low, the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 PCLKB clock cycles.

DBLANS[4:0] Bits (A/D Conversion Data Duplication 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 selected channel are stored into 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 36.4 shows the relationship between the DBLANS bit settings and selected duplication channel. A/D-converted value addition mode with double trigger mode can be set by selecting the channel selected by the DBLANS[4:0] bits using the ADADS register. If double trigger mode is selected, the channel selected by the ADANSA register is invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead. When converting analog inputs of channels, internal reference voltage should not be selected for A/D conversion.
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 36.4 Relationship between DBLANS Bit Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000	AN000	01000	AN008
00001	AN001	01001	AN009
00010	AN002	01010	AN010
00011	AN003	01011	AN011
00100	AN004	01100	AN012
00101	AN005	01101	AN013
00110	AN006	01110	AN014
00111	AN007	01111	AN015

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

DBLE Bit (Double Trigger Mode Select)

In double trigger mode, the following operation is performed after scanning is started by the MTU, TPU, or ELC trigger selected by the TRSA[3:0] bits in ADSTRGR.

1. When the ADIE bit is 1, a scan end interrupt is not output upon first scan completion but is output upon second scan completion.
2. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] are stored into A/D data register y for the first time, and into the A/D data duplication register for the second time.

Setting the DBLE bit to 1 invalidates the channel selected by the ADANSA register. In continuous scan mode, double trigger mode should not be selected. Internal reference voltage should not be selected for A/D conversion. In double trigger mode, software trigger should not be selected. The DBLE bit should be set while ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

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 (S12ADI0) in scans except for group B scan in group scan mode.

With double trigger mode deselected, the S12ADI0 interrupt is generated when the first scan is completed if the ADIE bit is set to 1.

When the internal reference voltage is selected, the S12ADI0 interrupt is also generated when A/D conversion is completed if the ADIE bit is set to 1.

With double trigger mode selected, the S12ADI0 interrupt is generated when the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the MTU, TPU, or ELC trigger selected by the TRSA[3:0] bits in ADSTRGR.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS bit selects the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 16 channels selected with the ADANSA register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, scan conversion is stopped.

In continuous scan mode, while the ADST bit in ADCSR is 1, A/D conversion is performed for the analog inputs of a maximum of 16 channels selected with the ADANSA register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated beginning at the first channel. A/D conversion is repeated until the ADST bit in ADCSR is set to 0.

In group scan mode, A/D conversion is performed for the analog inputs (group A) of a maximum of 16 channels selected with the ADANSA register in the ascending order of the channel number after scanning is started by the MTU, TPU, or ELC trigger selected by the TRSA[3:0] bits in ADSTRGR, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 16 channels selected with the ADANSB register in the ascending order of the channel number after scanning is started by the MTU, TPU, or ELC trigger selected by the TRSB[3:0] bits in ADSTRGR, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. In group scan mode, different channels and triggers should be selected for group A and group B.

When the internal reference voltage is selected, single scan mode should be selected and all the channels selected by the ADANSA register should be deselected, after which A/D conversion is to be started. The A/D conversion stops after completion of A/D conversion of the internal reference voltage selected.

The ADCS bit should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

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 (MTU, TPU, or ELC) selected by the ADSTRGR.TRSA[3:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- A synchronous trigger (MTU, TPU, or ELC) selected by the ADSTRGR.TRSB[3:0] 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[3:0] bits being set to 0000b.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the channels selected is completed in single scan mode.
- The A/D conversion of the internal reference voltage selected is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.

36.2.5 A/D Channel Select Register A (ADANSA)

Address: 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSA[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	ANSA[15:0]	A/D Conversion Channels Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

ADANSA selects analog input channels for A/D conversion from among AN000 to AN015. In group scan mode, group A channels are to be selected.

ANSA[15:0] Bits (A/D Conversion Channels Select)

The ANSA[15:0] bits select analog input channels for A/D conversion from among AN000 to AN015. The channels to be selected and the number of channels can be arbitrarily set. The ANSA[0] bit corresponds to AN000 and the ANSA[15] bit corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the internal reference voltage should not be performed.

When double trigger mode is selected, the channel selected by the ANSA[15:0] bits is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA[15:0] bits should be set while the ADCSR.ADST bit is 0.

36.2.6 A/D Channel Select Register B (ADANSB)

Address: 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSB[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	ANSB[15:0]	A/D Conversion Channels Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

ADANSB selects channels for A/D conversion in group B from among AN000 to AN015 in group scan mode. ADANSB is not used in any other scan mode. The channels for conversion can be selected from among the channels other than group A channels, which are selected by the ADANSA register or ADCSR.DBLANS[4:0] bits in double trigger mode. The ANSB[0] bit corresponds to AN000 and the ANSB[15] bit corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the internal reference voltage should not be performed.

The ANSB[15:0] bits should be set while the ADST bit is 0.

36.2.7 A/D-Converted Value Addition Mode Select Register (ADADS)

Address: 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADS[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	ADS[15:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode for AN000 to AN015 is not selected. 1: A/D-converted value addition mode for AN000 to AN015 is selected.	R/W

ADADS selects the channels 0 to 15 on which A/D conversion is performed successively two to four times and then converted values are added (integrated).

ADS[15:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADS[n] bit of the number that is the same as that of A/D converted channel selected by ANSA[n] bits (n = 0 to 15) in ADANSA or DBLANS[4:0] bits in ADCSR and ANSB[n] bits (n = 0 to 15) in ADANSB is set to 1, these bits perform A/D conversion of analog input of the selected channels successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.
The ADS[15:0] bits should be set while the ADCSR.ADST bit is 0.

Figure 36.2 shows a scanning operation sequence in which both the ADS[2] and ADS[6] bits are set to 1.
In continuous scan mode (ADCSR.ADCS = 10b), it is assumed that the addition count is set to 3 (ADADC.ADC[1:0] = 11b) and the channels AN000 to AN007 are selected (ADANSA.ANSA[15:0] = 00FFh). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to the A/D data register 2. After that the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is returned to the A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.
For the channel for which the addition mode is not selected, the A/D data register format is determined by the ADRFMT bit in ADCER (right-alignment or left-alignment).

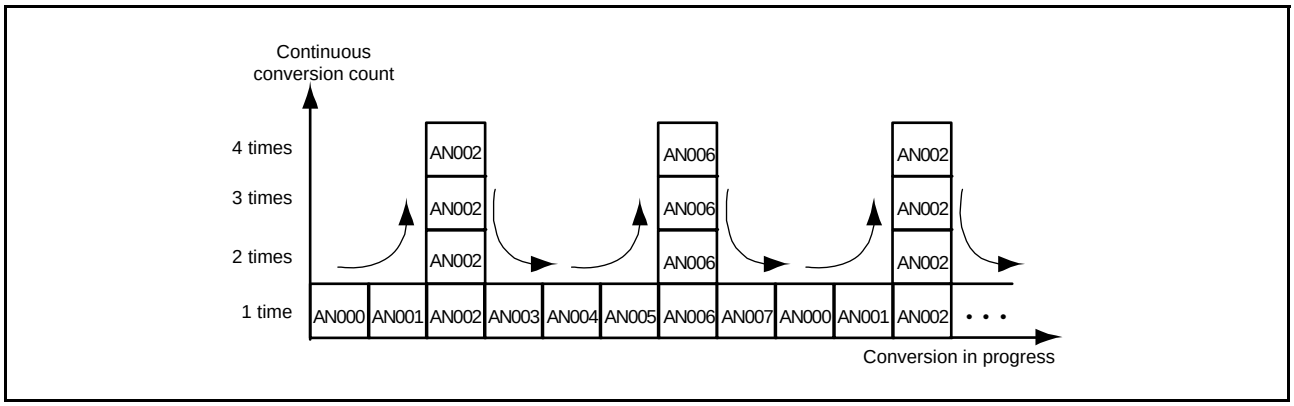


Figure 36.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS[2] = 1, and ADS[6] = 1

36.2.8 A/D-Converted Value Addition Count Select Register (ADADC)

Address: 0008 900Ch

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADC[1:0]	Addition Count Select	b1 b0 0 0: 1-time conversion (no addition; same as normal conversion) 0 1: 2-time conversion (addition once) 1 0: 3-time conversion (addition twice) 1 1: 4-time conversion (addition three times)	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADADC sets the addition count for the channels for which A/D-converted value addition mode is selected, and for A/D conversion of the internal reference voltage.

ADC[1:0] Bits (Addition Count Select)

The ADC[1:0] bits set the addition count common to the channels for which A/D conversion or A/D-converted value addition mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of the internal reference voltage.

The ADC[1:0] bits should be set while the ADCSR.ADST bit is 0.

36.2.9 A/D Control Extended Register (ADCER)

Address: 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[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
b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2, b1	ADPRC[1:0]	A/D Data Register Bit Precision Specification	b9 b8 0 0: Values are stored with 12-bit precision in the A/D data registers. 0 1: Values are stored with 10-bit precision in the A/D data registers. 1 0: Values are stored with 8-bit precision in the A/D data registers. 1 1: Do not make this setting.	R/W
b5	ACE	Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Conversion Voltage Select for Self-Diagnosis	b9 b8 0 0: Setting prohibited when self-diagnosis is enabled 0 1: Uses 0 V for self-diagnosis. 1 0: Uses VREFH0 × 1/2 for self-diagnosis. 1 1: Uses VREFH0 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-alignment is selected for the A/D data register format. 1: Left-alignment is selected for the A/D data register format.	R/W

ADPRC[1:0] Bit (A/D Data Register Bit Precision Specification)

These bits select storage of the results of A/D conversion with eight-, 10-, or 12-bit precision in the ADDRy and ADDBLDR registers.

ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (All 0) of ADDRy, ADRD, ADOCDR, and ADDBLDR after the register has been read by the CPU, DTC, or DMAC. This function enables update failures of ADDRy, ADRD, ADOCDR, and ADDBLDR to be detected.

DIAGVAL[1:0] Bits (Conversion Voltage Select for Self-Diagnosis)

For details, refer to the ADCER.DIAGLD bit description.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 with these bits 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 to 0 allows conversion of the voltages in rotation mode where 0 V, VREFH0 × 1/2, and VREFH0 are converted in this order. After reset, 0 V is first converted if rotation mode is selected whereas the fixed voltage specified

by the ADCER.DIAGVAL[1:0] bits is converted if fixed mode is selected. In rotation mode, the self-diagnosis voltage value does not return to 0 V when scan conversion is completed; when scan conversion is restarted, 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 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 of the internally generated voltage values 0 V, VREFH0 × 1/2, and VREFH0 is converted. When conversion is completed, information of the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read 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. The execution time of self-diagnosis equals to the A/D conversion time of one channel. To execute self-diagnosis, A/D conversion of the internal reference voltage should not be selected. If selected, self-diagnosis is not executed. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.
The DIAGM bit should be set while the ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies left-alignment or right-alignment for the data to be stored in ADDRy, ADRD, ADOCDR, and ADDBLDR.
When the A/D converted value addition mode is selected, the format of each data register is fixed to left-alignment, irrespective of the ADCER.ADRFMT bit value.

For details on the format of the data registers, see section 36.2.1, A/D Data Registers y (ADDRy; y = 0 to 15), A/D Data-Doubling Register (ADDBLDR), section 36.2.2, A/D Internal Reference Voltage Data Register (ADOCDR), and section 36.2.3, A/D Self-Diagnosis Data Register (ADRD).

36.2.10 A/D Start Trigger Select Register (ADSTRGR)

Address: 0008 9010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	TRSA[3:0]				—	—	—	—	TRSB[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	TRSB[3:0]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TRSA[3:0]	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 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRSB[3:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[3:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[3:0] 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 software trigger or asynchronous trigger is prohibited. Therefore, the TRSB[3:0] bits should be set to the value other than 0000 and the ADCSR.TRGE bit should be set to 1 in group scan mode.

Table 36.5 shows the A/D conversion startup sources selected by TRSB[3:0] bits.

TRSA[3:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[3:0] 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. For scan execution in group scan mode or double trigger mode, software trigger or asynchronous trigger cannot be used.

- When using the A/D conversion startup source of the synchronous trigger (MTU, TPU, or ELC), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 0.
- When using the asynchronous trigger (ADTRG0#), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 1.
- Software trigger (ADST bit in ADCSR) is enabled regardless of the set values of the TRGE, EXTRG in ADCSR, and the TRSA[3:0] bits.

Table 36.6 shows the A/D conversion startup sources selected by TRSA[3:0] bits

Table 36.5 List of A/D Conversion Startup Sources Selected by TRSB[3:0] Bits

Module	Source	Remarks	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
MTU	TRG0AN	TRGA input capture/compare match from MTU0	0	0	0	1
	TRG0BN	TRGB input capture/compare match B from MTU0	0	0	1	0
	TRGAN	TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	0	0	1	1
	TRG0EN	TRGE compare match from MTU0	0	1	0	0
	TRG0FN	TRGF compare match from MTU0	0	1	0	1
	TRG4AN	MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	0
	TRG4BN	MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	1
	TRG4ABN	MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	1	0	0	0
ELC	ELC	Trigger from ELC	1	0	0	1
TPU	TRGAN1	TRGA compare match/input capture from TPU0 to TPU4	1	0	1	1
	TRG4ABN1	TRGA compare match/input capture from TPU0	1	1	0	0

Table 36.6 List of A/D Conversion Startup Sources Selected by TRSA[3:0] Bits

Module	Source	Remarks	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
ADC	ADST	Software trigger	—	—	—	—
External input	ADTRG0#	A/D conversion start trigger pin	0	0	0	0
MTU	TRG0AN	TRGA input capture/compare match from MTU0	0	0	0	1
	TRG0BN	TRGB input capture/compare match B from MTU0	0	0	1	0
	TRGAN	TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	0	0	1	1
	TRG0EN	TRGE compare match from MTU0	0	1	0	0
	TRG0FN	TRGF compare match from MTU0	0	1	0	1
	TRG4AN	MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	0
	TRG4BN	MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	1
	TRG4ABN	MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	1	0	0	0
ELC	ELC	Trigger from ELC	1	0	0	1
TPU	TRGAN1	TRGA compare match/input capture from TPU0 to TPU4	1	0	1	1
	TRG4ABN1	TRGA compare match/input capture from TPU0	1	1	0	0

36.2.11 A/D Conversion Extended Input Control Register (ADEXICR)

Address: 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCS	—	—	—	—	—	—	—	OCSAD	—
Value after 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	OCSAD	Internal Reference Voltage A/D Converted Value Addition Mode Select	0: Internal reference voltage A/D converted value addition mode is not selected 1: Internal reference voltage A/D converted value addition mode is selected	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OCS	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed 1: A/D conversion of internal reference voltage is performed	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

OCSAD Bit (Internal Reference Voltage A/D Converted Value Addition Mode Select)

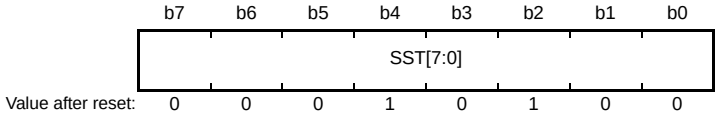
The OCSAD bit selects A/D conversion for the internal reference voltage. Setting the OCSAD bit to 1 performs A/D conversion of the internal reference voltage successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the integrated value to the A/D internal reference voltage data register (ADOCDR). The OCSAD bit should be set while the ADST bit in ADCSR is 0.

OCS Bit (Internal Reference Voltage A/D Conversion Select)

The OCS bit selects A/D conversion for the internal reference voltage. When A/D conversion of the internal reference voltage is to be performed, all the bits in ADANSA and the TSS bit should be set to 0 in single scan mode. The OCS bit should be set while the ADST bit is 0.

36.2.12 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, O)

Address: ADSSTR0: 0008 9060h, ADSSTR1: 0008 9073h, ADSSTR2: 0008 9074h, ADSSTR3: 0008 9075h, ADSSTR4: 0008 9076h, ADSSTR5: 0008 9077h, ADSSTR6: 0008 9078h, ADSSTR7: 0008 9079h, ADSSTRL: 0008 9061h, ADSSTRO: 0008 9071h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SST[7:0]	Sampling Time Setting	Sets the sampling time (12 to 255 states).	R/W

ADSSTRn sets the sampling time for analog input.
One state is one ADCLK (A/D conversion clock) cycle. When the ADCLK is 50 MHz, one state is 20 ns. The initial value is 20 states. If the impedance of the analog input signal source is too high to secure sufficient sampling time or if the ADCLK is slow, the sampling time can be adjusted. ADSSTRn should be set while the ADST bit in ADCSR is 0. The set value for sampling time should be 12 or more states and 255 or less states. The sampling time should be 0.4 µs or longer. Table 36.7 shows the A/D sampling state registers and corresponding channels.

Table 36.7 A/D Sampling State Registers and Corresponding Channels

Bit Name	Corresponding Channels
ADSSTR0.SST[7:0]	AN000/self-diagnosis
ADSSTR1.SST[7:0]	AN001
ADSSTR2.SST[7:0]	AN002
ADSSTR3.SST[7:0]	AN003
ADSSTR4.SST[7:0]	AN004
ADSSTR5.SST[7:0]	AN005
ADSSTR6.SST[7:0]	AN006
ADSSTR7.SST[7:0]	AN007
ADSSTRL.SST[7:0]	AN008 to AN015
ADSSTRO.SST[7:0]	Internal reference voltage

36.2.13 A/D Sample and Hold Circuit Control Register (ADSHCR)

Address: 0008 9066h

	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	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SSTSH[7:0]	Sampling Time Sample-and-Hold Circuit Setting	Set the sampling time (4 to 255 states).	R/W
b10 to b8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. 0: Bypass the channel-dedicated sample-and-hold circuits. 1: Use the channel-dedicated sample-and-hold circuits.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSHCR sets the parameters related to channel-dedicated sample-and-hold circuits.

SSTSH[7:0] Bits (Sampling Time Sample-and-Hold Circuit Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits. One state is one ADCLK (A/D conversion clock) cycle. When the ADCLK is 50 MHz, one state is 20 ns. The initial value is 20 states. If the impedance of the 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 SSTSH[7:0] bits should be set while the ADST bit in ADCSR is 0. The set value for sampling time should be 4 or more states and 255 or less states. The sampling time should be 0.4 µs or longer. For example, when the ADCLK is 25 MHz, the lower limit of the set value for sampling states is 10 states.

SHANS[2:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. The SHANS[0] bit selects AN000, SHANS[1] bit selects AN001, and SHANS[2] bit selects AN002. The SHANS[2:0] bits should be set while the ADST bit in ADCSR is 0.

36.2.14 A/D Disconnecting Detection Control Register (ADDISCR)

Address: 0008 907Ah

	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
b4 to b0	ADNDIS[4:0]	Disconnection Detection Assist Setting	Set the disconnection detection assist function.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADDISCR sets the disconnection detection assist function.

ADNDIS[4:0] Bits (Disconnection Detection Assist Setting)

The ADNDIS[4:0] bits select precharging or discharging of the disconnection detection assist function and set the precharging or discharging time. Setting the ADNDIS[4] bit to 1 selects precharging and setting ADNDIS[4] to 0 selects discharging. The ADNDIS[3:0] bits set the precharging or discharging time. When the ADNDIS[3:0] bits are set to 0000b, the disconnection detection assist function is disabled. When the ADNDIS[3:0] bits are set to any value other than 0000b, the set value is interpreted as the number of states, that is, the precharging or discharging time. However, setting the ADNDIS[3:0] bits to 00001b or 00010b is prohibited.

When the internal reference voltage is to be converted, the disconnection detection assist function is not available; the ADNDIS[4:0] bits should be set to 00000b in this case.

The disconnection detection assist function is enabled when 550 ns has elapsed after the ADNDIS[4:0] bits are set. When this function is to be used, make sure that 550 ns has elapsed after the ADNDIS[4:0] bits are set before starting A/D conversion.

36.3 Operation

36.3.1 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 ADST bit in ADCSR is cleared to 0 from 1 by software. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after starting to be scanned according to the respective triggers.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A and group B selected by the ADANSA and ADANSB registers, respectively, starting from the channel with the smallest number n.
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.
When the internal reference voltage is selected, single scan mode should be used for A/D conversion.
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 DBLANS[4:0] bits in ADCSR is duplicated only if the conversion is started by any of the MTU, TPU, or ELC triggers selected by TRSA[3:0] bits in ADSTRGR.
When any of AN000 to AN002 channels is set by the SHANS[2:0] bits in ADSHCR so that the channel uses a channel-dedicated sample-and-hold circuit, the set analog input is sampled and held before the first A/D conversion of each scan.

36.3.2 Single Scan Mode

36.3.2.1 Basic Operation (Channel-Dedicated Sample-and-Hold Circuits not Used)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below. In selected channel scanning, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).

(4) The ADST bit remains 1 (A/D conversion start) 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.

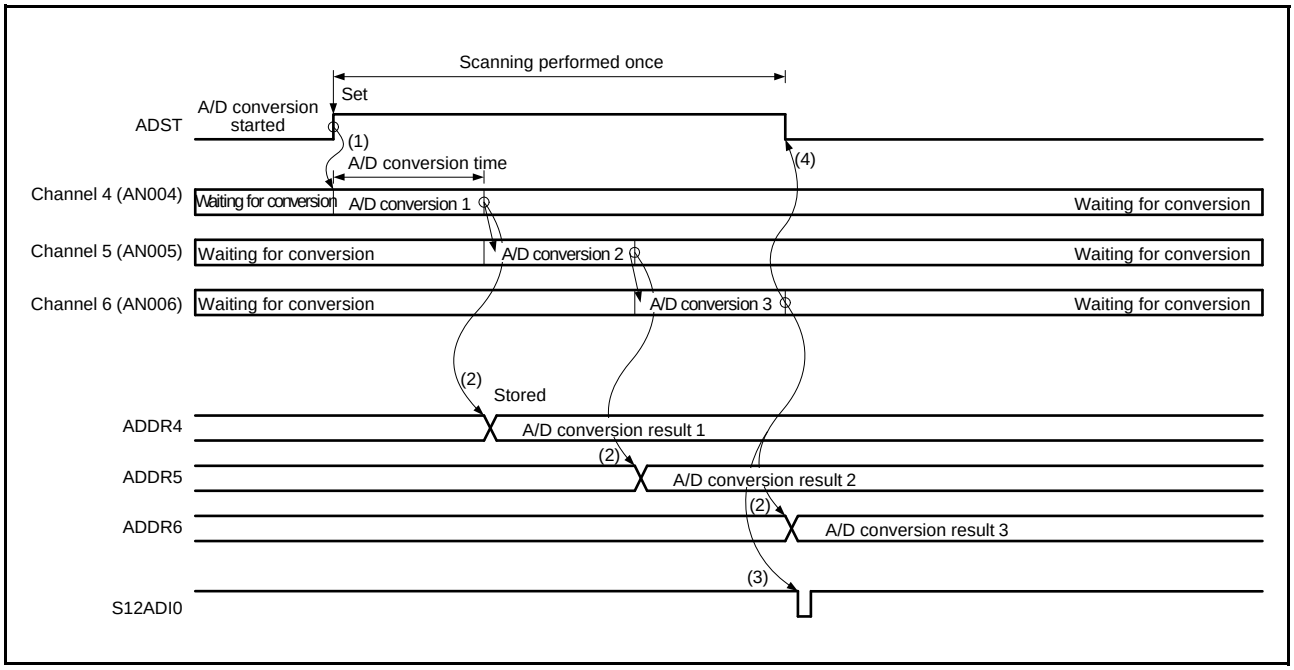


Figure 36.3 Example of Operation in Single Scan Mode (Basic Operation: AN004 to AN006 Selected)

36.3.2.2 Basic Operation (Channel-Dedicated Sample-and-Hold Circuits Used)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed once on the analog input of all the selected channels as below. The channels whose channel-dedicated sample-and-hold circuit is to be used can be selected by SHANS[2:0] bits in ADSHCR.
In selected channel scanning, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) Analog input sampling of all the channels whose channel-dedicated sample-and-hold circuit is to be used is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.
(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).
(5) The ADST bit remains 1 (A/D conversion start) 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.

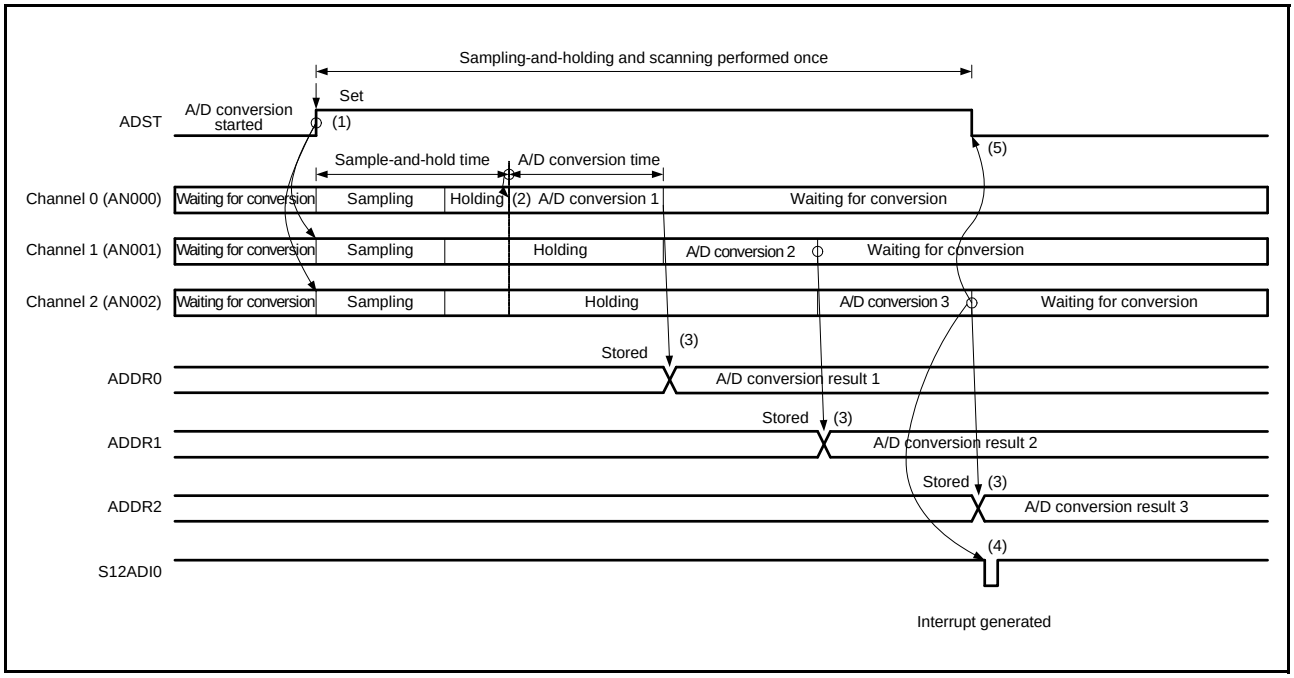


Figure 36.4 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)

36.3.2.3 Channel Selection and Self-Diagnosis (Channel-Dedicated Sample-and-Hold Circuits not Used)

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (×0, ×1/2, or ×1) supplied to the A/D converter, and then A/D conversion is performed once on the analog input of the selected channels as below. In selected channel scanning, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) A/D conversion for self-diagnosis is first started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input.

(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).

(5) The ADST bit remains 1 (A/D conversion start) 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.

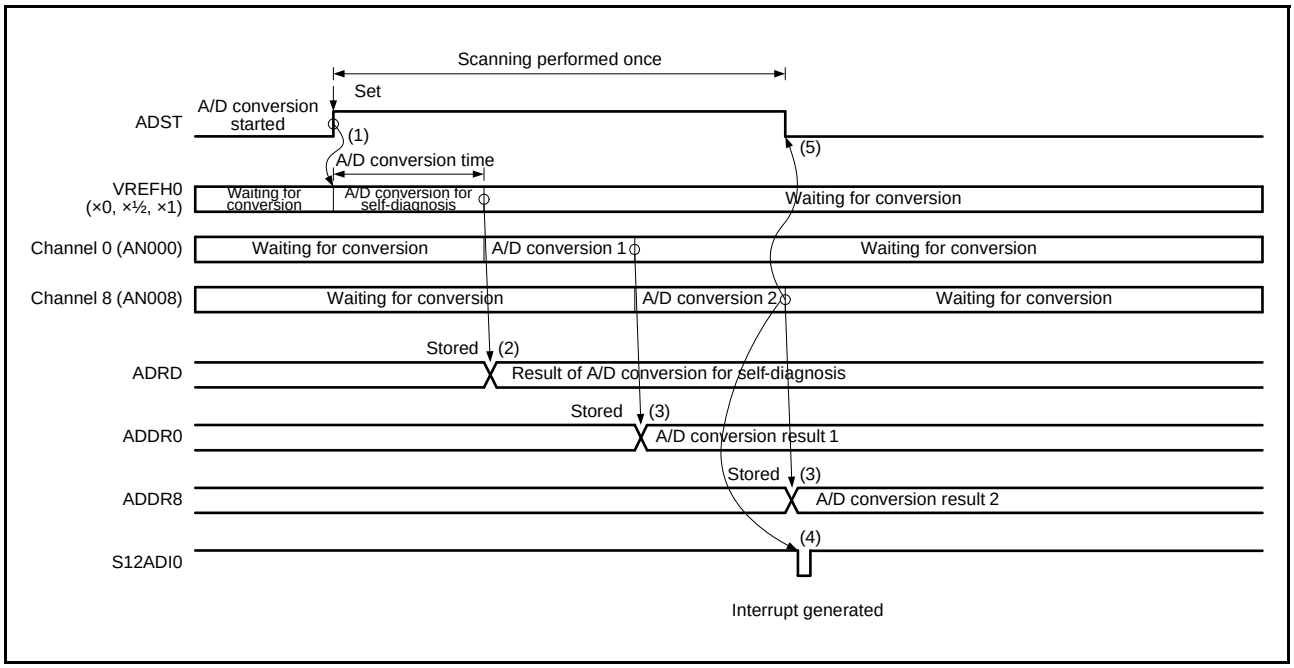


Figure 36.5 Example of Operation in Single Scan Mode (Basic Operation + Self-Diagnosis)

36.3.2.4 Channel Selection and Self-Diagnosis (Channel-Dedicated Sample-and-Hold Circuits Used)

When the channel-dedicated sample-and-hold circuit is used and channels and self-diagnosis are selected, sample-and-hold operation is first performed, and then A/D conversion is performed once for the reference voltage VREFH0 (×0, ×1/2, or ×1) 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.

In selected channel scanning, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) Analog input sampling of all the channels whose channel-dedicated sample-and-hold circuit is to be used is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input.

(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.

(3) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).

(6) The ADST bit remains 1 (A/D conversion start) 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.

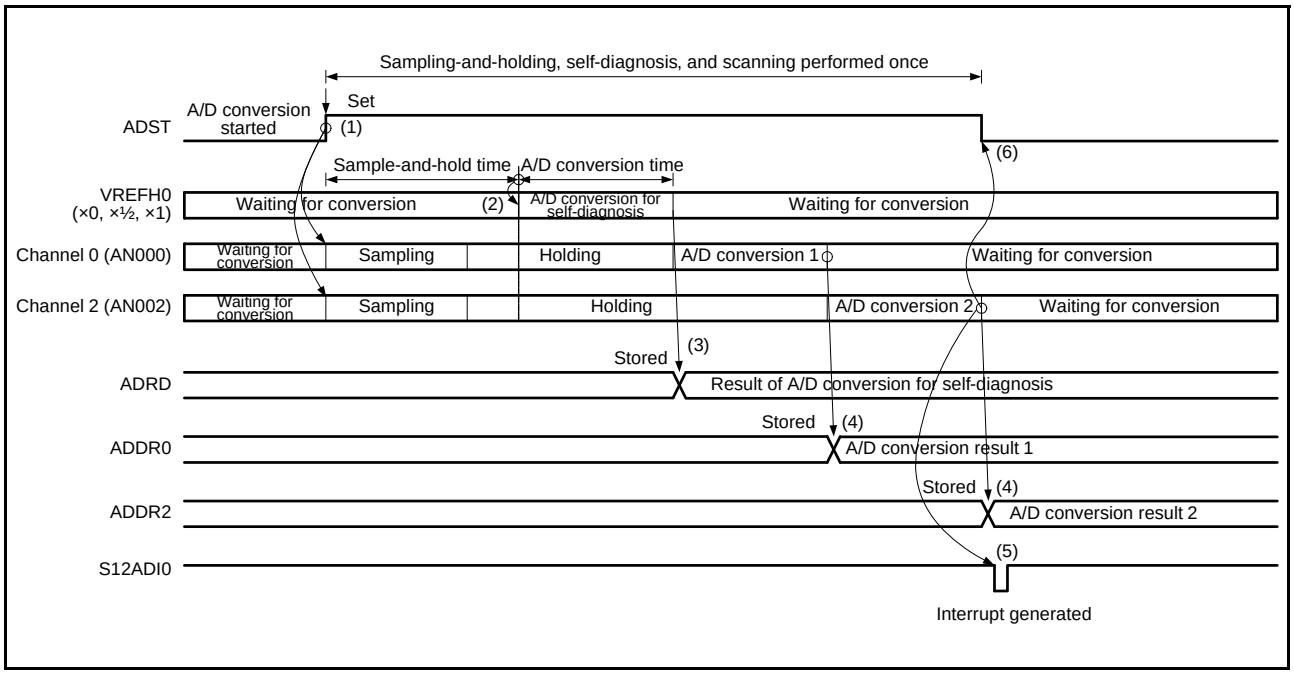


Figure 36.6 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used + Self-Diagnosis)

36.3.2.5 A/D Conversion when Internal Reference Voltage is Selected

A/D conversion of the internal reference voltage should be performed in single scan mode. The operation is as follows. All the channels should be deselected (set the ANSA[15:0] bits in ADANSA to 0000h and DBLE bit in ADCSR to 0) and self-diagnosis should be deselected.

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, the synchronous trigger (MTU, TPU, or ELC), or the asynchronous trigger input, A/D conversion is started for the internal reference voltage.

(2) When A/D conversion is completed, the A/D conversion result is stored into the A/D internal reference voltage data register (ADOCDR). If the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.

(3) The ADST bit remains 1 (A/D conversion start) 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.

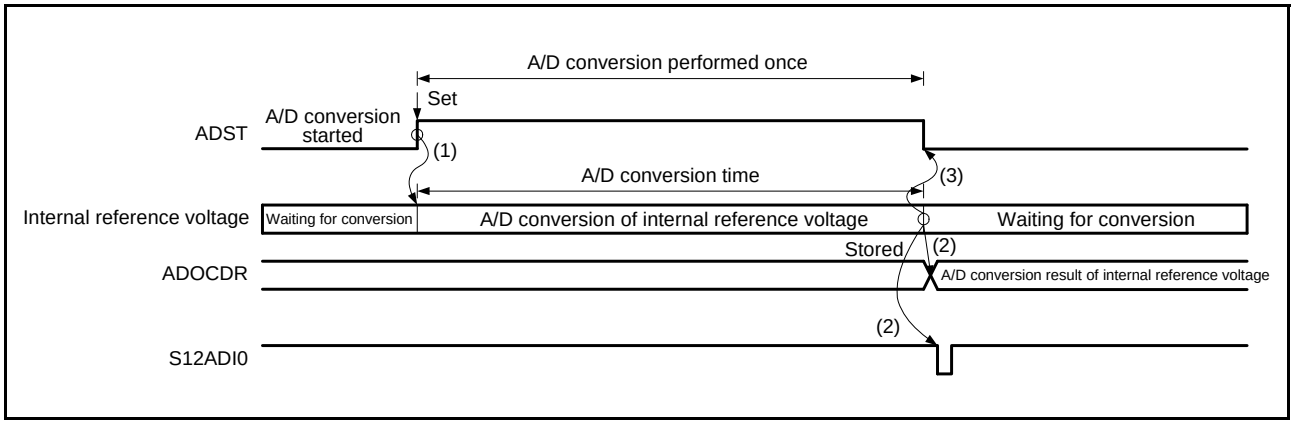


Figure 36.7 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)

36.3.2.6 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by the MTU, TPU, or ELC trigger is performed twice as below.

Self-diagnosis should be deselected, and the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA register is invalid. In double trigger mode, MTU, TPU, or ELC triggers should be selected using the TRSA[3:0] bits in ADSTRGR; the EXTRG bit and TRGE bit in ADCSR should be set to 0 and 1, respectively. Software trigger should not be used.

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by the MTU, TPU, or ELC trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.

(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI0 interrupt request is not generated irrespective of the ADIE (S12ADI0 interrupt upon scanning completion enabled) bit setting in ADCSR.

(4) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.

(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.

(6) If the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.

(7) The ADST bit remains 1 (A/D conversion start) 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.

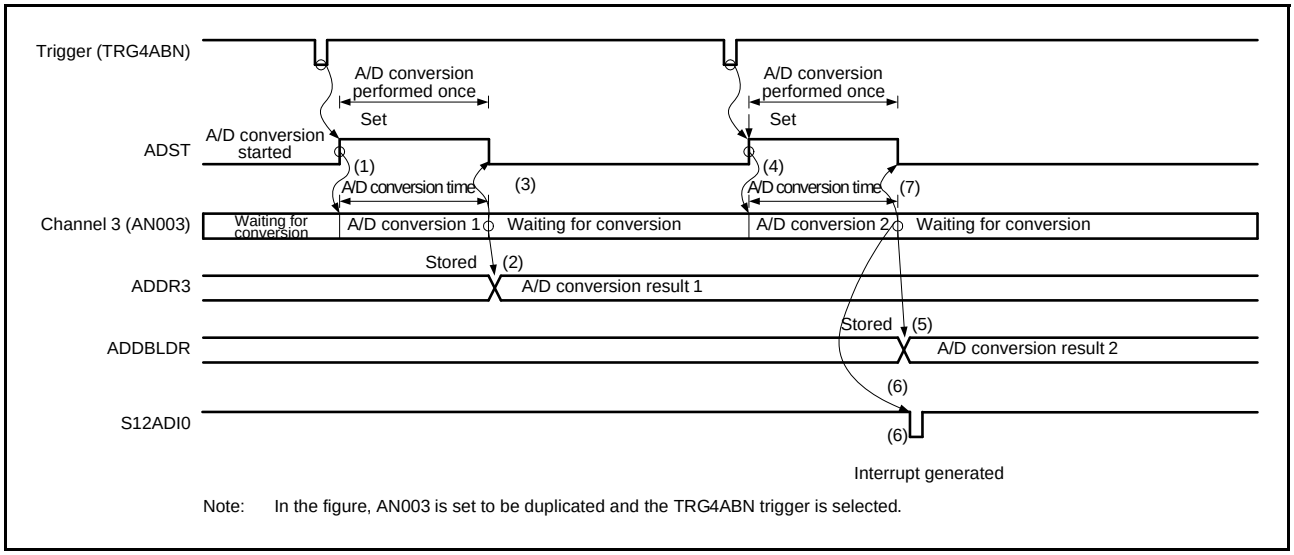


Figure 36.8 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

36.3.3 Continuous Scan Mode

36.3.3.1 Basic Operation (Channel-Dedicated Sample-and-Hold Circuits not Used)

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, the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(4) The ADST bit in ADCSR is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADST bit in ADCSR is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

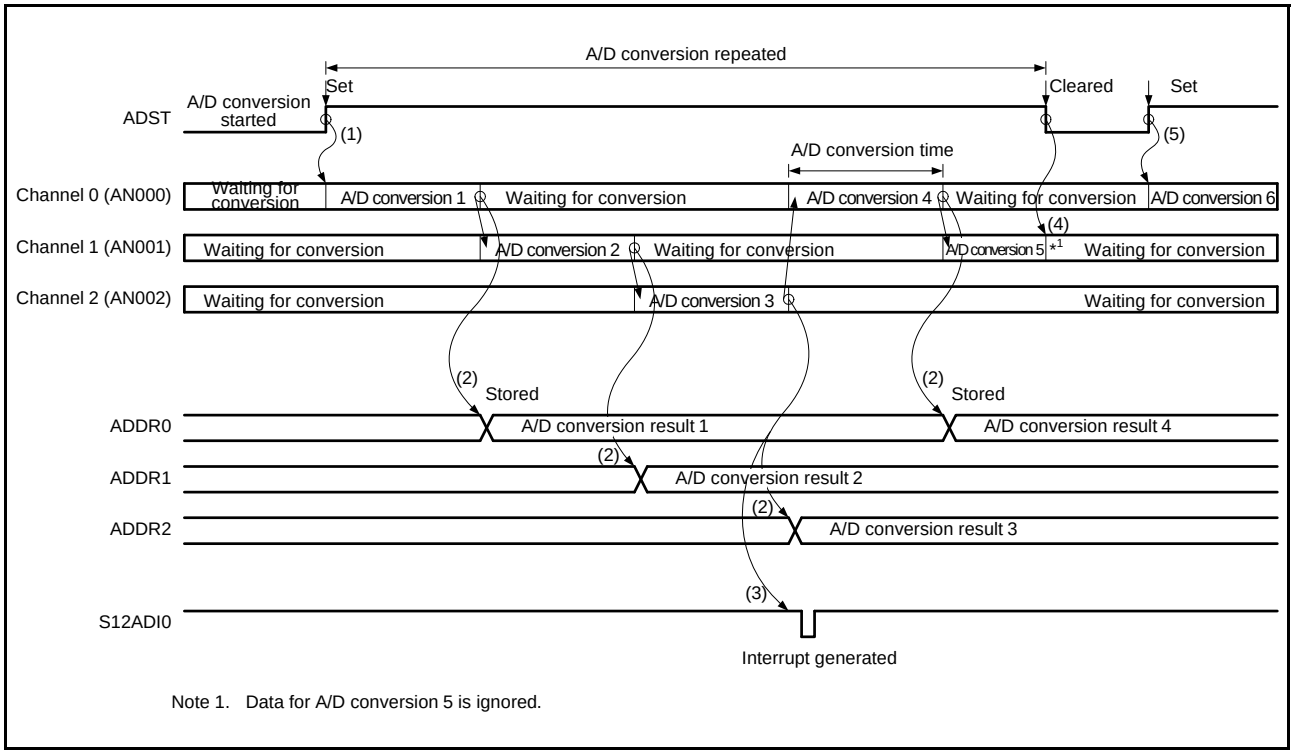


Figure 36.9 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

36.3.3.2 Basic Operation (Channel-Dedicated Sample-and-Hold Circuits Used)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is repeated on the analog input of all the selected channels as below. The channels whose channel-dedicated sample-and-hold circuit is to be used can be selected by the SHANS[2:0] bits in ADSHCR.
In continuous scanning mode, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both be set to 0 (non-selection).

(1) Analog input sampling of all the channels whose channel-dedicated sample-and-hold circuit is to be used is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software or synchronous trigger (MTU, TPU, or ELC) input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.
(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled). At the same time, analog input sampling is started for all the channels whose channel-dedicated sample-and-hold circuit is to be used.
(5) The ADST bit is not automatically cleared to 0 and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADST bit is later set to 1 (A/D conversion start), analog input sampling is started again for all the channels whose channel-dedicated sample-and-hold circuit is to be used.

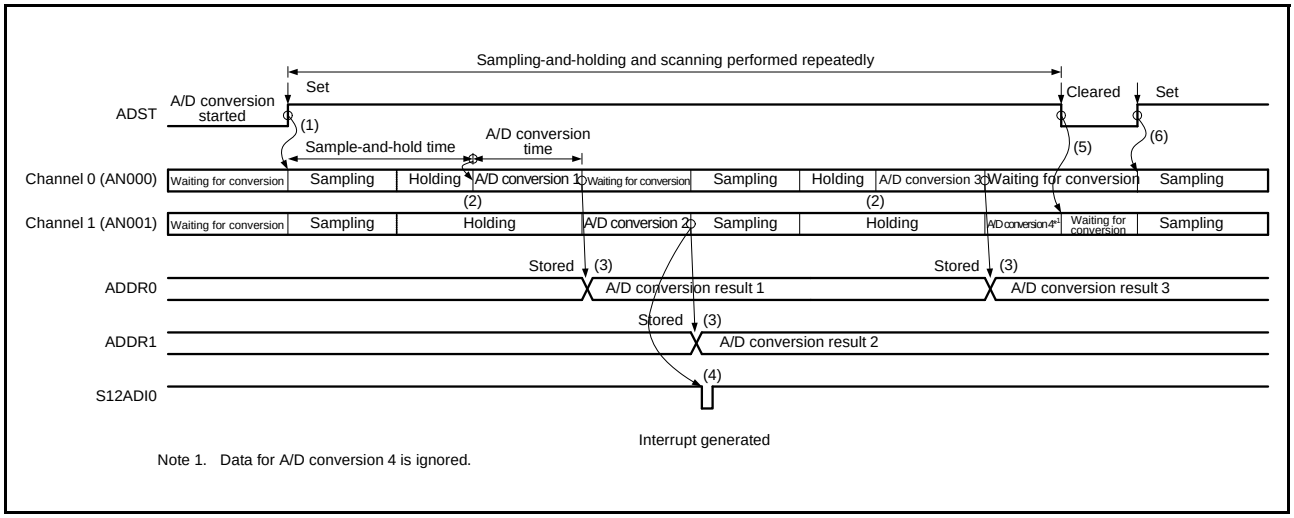


Figure 36.10 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)

36.3.3.3 Channel Selection and Self-Diagnosis (Channel-Dedicated Sample-and-Hold Circuits not Used)

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (×0, ×1/2, or ×1) 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.

In continuous scan mode, the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) A/D conversion for self-diagnosis is first started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input.

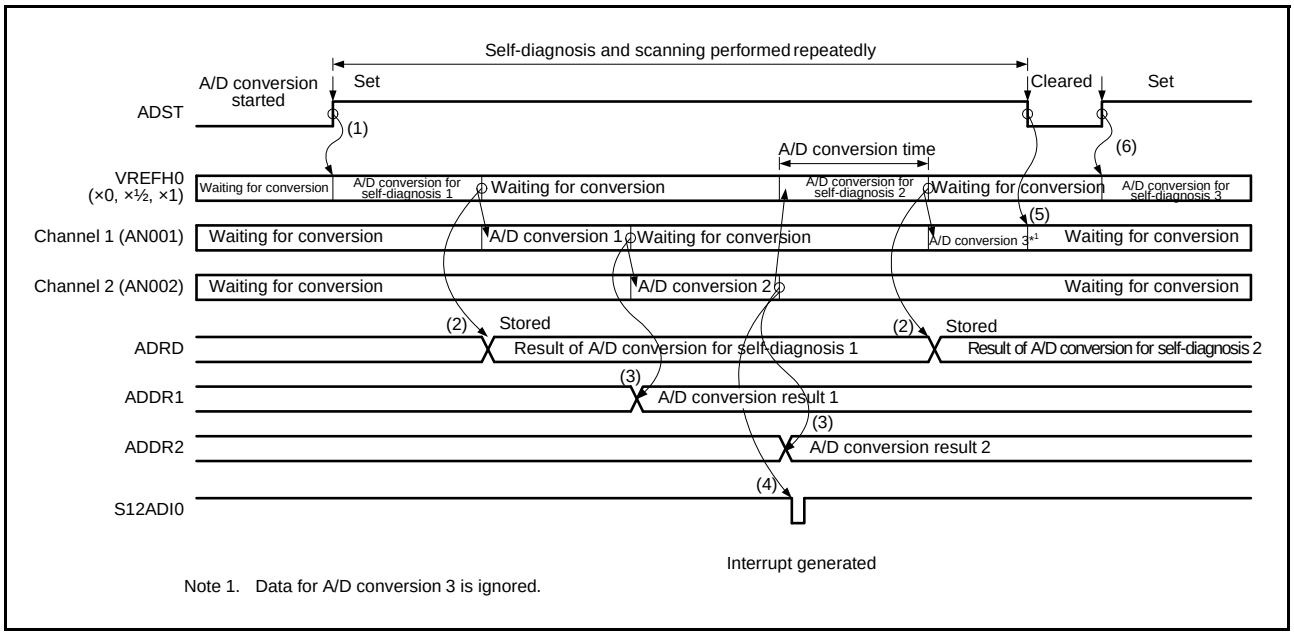
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(5) The ADST bit is not automatically cleared to 0 and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(6) When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note 1. Data for A/D conversion 3 is ignored.

Figure 36.11 Example of Operation in Continuous Scan Mode (Basic Operation + Self-Diagnosis)

36.3.3.4 Channel Selection and Self-Diagnosis (Channel-Dedicated Sample-and-Hold Circuits Used)

When the channel-dedicated sample-and-hold circuit is used and channels and self-diagnosis are selected, sample-and-hold operation is first performed, and then A/D conversion is performed for the reference voltage VREFH0 (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter, and A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

In continuous scan mode, internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

(1) Analog input sampling of all the channels whose channel-dedicated sample-and-hold circuit is to be used is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or ELC), or asynchronous trigger input.

(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.

(3) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled). At the same time, analog input sampling is started for all the channels whose channel-dedicated sample-and-hold circuit is to be used.

(6) The ADST bit is not automatically cleared to 0 and steps 2 to 5 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(7) When the ADST bit is later set to 1 (A/D conversion start), analog input sampling is started again for all the channels whose channel-dedicated sample-and-hold circuit is to be used.

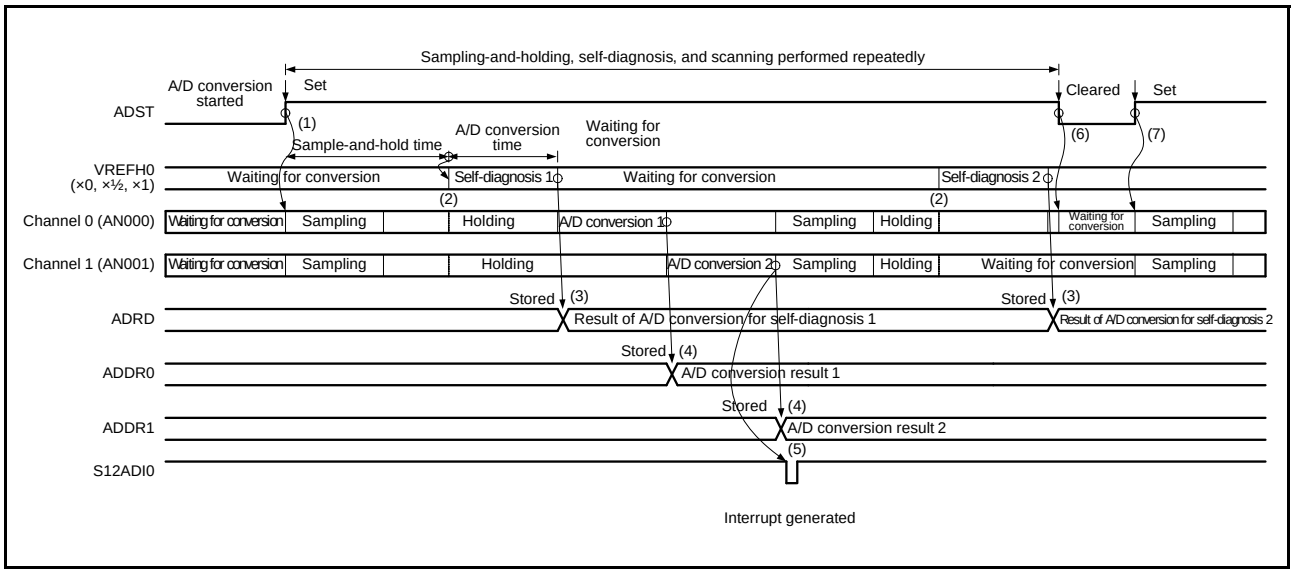


Figure 36.12 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used + Self-Diagnosis)

36.3.4 Group Scan Mode

36.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by the MTU, TPU, or ELC trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.
The group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.
The group A and group B channels to be A/D-converted are selected using the ADANSA register and ADANSB register, respectively. Group A and group B cannot use the same channels.
In group scan mode, the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.
The following describes operation in group scan mode using a trigger from the MTU. Specifically, the TRG4AN and TRG4BN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, an S12ADI0 interrupt is output if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).

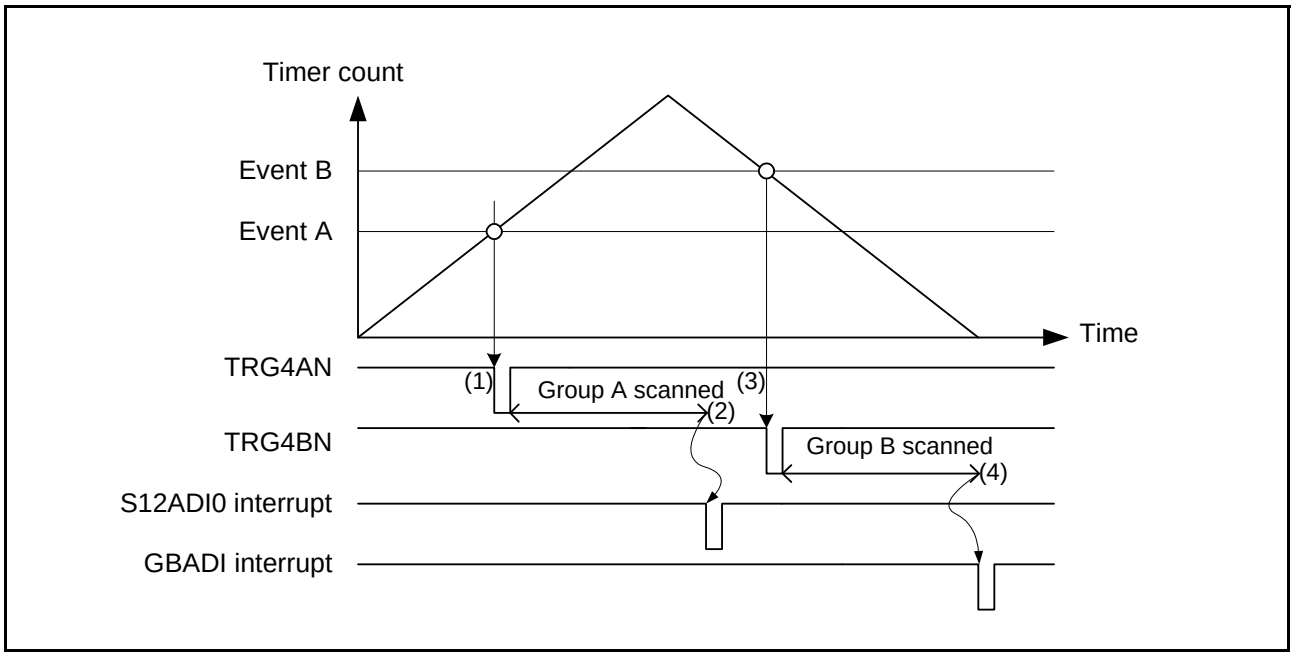


Figure 36.13 Example of Operation in Group Scan Mode (Basic Operation: MTU Triggers Used)

36.3.4.2 A/D Conversion in Double Trigger Mode

In group scan mode with double trigger mode, single scan operation started by the MTU, TPU, or ELC trigger is performed twice for group A. For group B, single scan operation started by the MTU, TPU, or ELC trigger is performed once.

In group scan mode, the group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger or asynchronous trigger (ADTRG0#) should not be used.

The group A and group B channels to be A/D-converted are selected using the DBLANS[4:0] bits in ADCSR register and ADANSB register, respectively. The same channels cannot be selected for both groups.

In group scan mode, the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should be set to 0 (non-selection).

In group scan mode with double trigger mode, self-diagnosis cannot be selected.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1.

The following describes operation in group scan mode with double trigger mode using a trigger from the MTU. Specifically, the TRG4ABN and TRG0AN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group B is started by the TRG0AN trigger from the MTU.
(2) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).
(3) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(4) When the first scanning of group A is completed, the conversion result is stored into ADDRy; an S12ADI0 interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(5) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(6) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An S12ADI0 interrupt is output if the ADIE bit is 1 (S12ADI0 interrupt enabled).

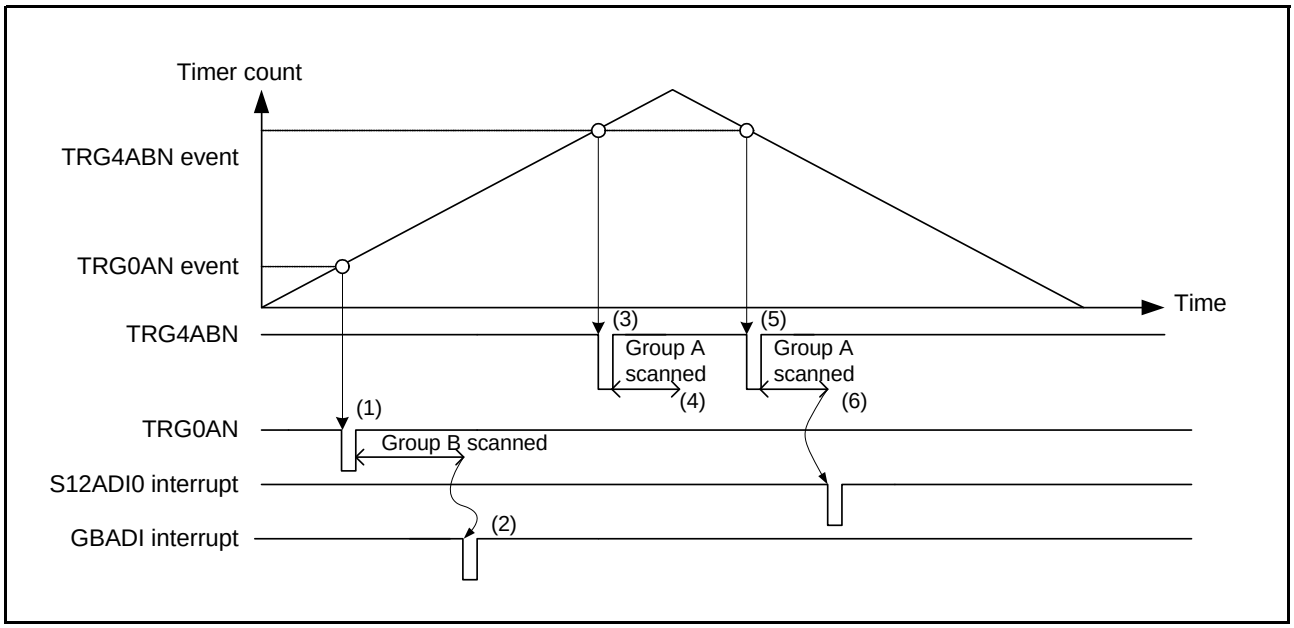


Figure 36.14 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: MTU Triggers Used)

36.3.5 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by software trigger; the triggers from the MTU, TPU, or ELC; or ADTRG0# (external trigger). After start-of-scanning-delay time (t_D) has passed, the A/D converter samples the channel-dedicated sample-and-hold circuits, executes the disconnection detection assist process, the conversion process for self-diagnosis, and then starts the A/D conversion process.

Figure 36.15 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software trigger or triggers from the MTU, TPU, or ELC. Figure 36.16 shows the scan conversion timing in single scan mode, in which scan conversion is activated by ADTRG0# (external trigger). The scan conversion time (t_{SCAN}) includes start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuits sampling time (t_{SH})*1, disconnection detection assist processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit sampling-and-holding end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}). Table 36.8 shows the specific scanning 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:

$$t_{SCAN} = t_D + t_{SH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{ED}$$

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 $t_{SH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{SHED}$.

The disconnection detection assist processing time (t_{DIS}) is the value set in the ADNDIS[3:0] bits.

The channel-dedicated sample-and-hold circuits sampling time (t_{SH}) is 10 states (fixed) + the value set in the ADSHCR.SSTSH[7:0] bits.

The self-diagnosis A/D conversion processing time (t_{DIAG}) is 30 states (fixed) + the value set in the ADSSTR0.SST[7:0] bits.

The A/D conversion processing time (t_{CONV}) is 30 states (fixed) + the value set in the ADSSTn.SST[7:0] bits*5.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SH} = 0$.

Note 2. When the disconnection detection assist function is not used, $t_{DIS} = 0$.

Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$.

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. Registers in Table 36.7.

Table 36.8 Scan Conversion Time (in Terms of PCLKB and ADCLK Cycles)

Item	Symbol	Conditions	Scan Conversion Time (Cycles)
Start-of-scanning-delay time*1	t_D	MTU, TPU, ELC, or software trigger	2 PCLKB + 4 ADCLK
		External trigger	4 PCLKB + 4 ADCLK
Channel-dedicated sample-and-hold circuits sampling time	t_{SH}	Set by ADSHCR.SSTSH[7:0] bits (initial value 14h)	30 ADCLK
Disconnection detection assist processing time	t_{DIS}	Set by ADNDIS[3:0] bits (initial value 00h)	0 ADCLK
Self-diagnosis A/D conversion processing time*1	t_{DIAG}	Set by ADSSTR0.SST[7:0] bits (initial value 14h)	50 ADCLK
A/D conversion processing time*1	t_{CONV}	Set by ADSSTRn.SST[7:0] bits (initial value 14h)	50 ADCLK
Channel-dedicated sample-and-hold circuit sampling-and-holding end time	t_{SHED}	—	2 ADCLK
End-of-scanning-delay time*1	t_{ED}	—	1 PCLKB + 3 ADCLK
Scan conversion time*2	t_{SCAN}	—	5 PCLKB + (50n + 87) ADCLK

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, refer to Figure 36.15 and Figure 36.16.
Note 2. It is assumed that scan conversion is activated by the external trigger, channel-dedicated sample-and-hold circuits are used, disconnection detection assist function is deselected, self-diagnosis A/D conversion is selected, and single scan mode is selected. n indicates the number of channels.

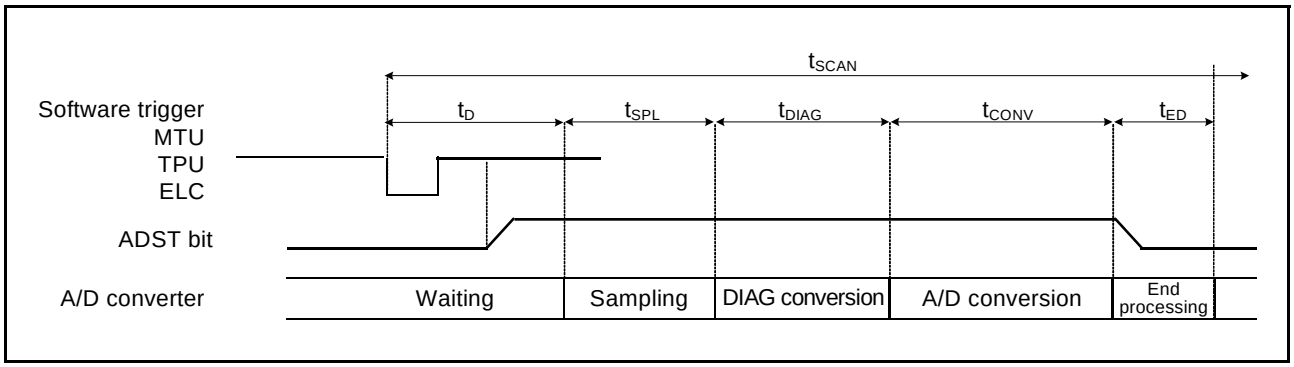


Figure 36.15 Scan Conversion Timing (Activated by Software, or Triggers from the MTU, TPU, or ELC)

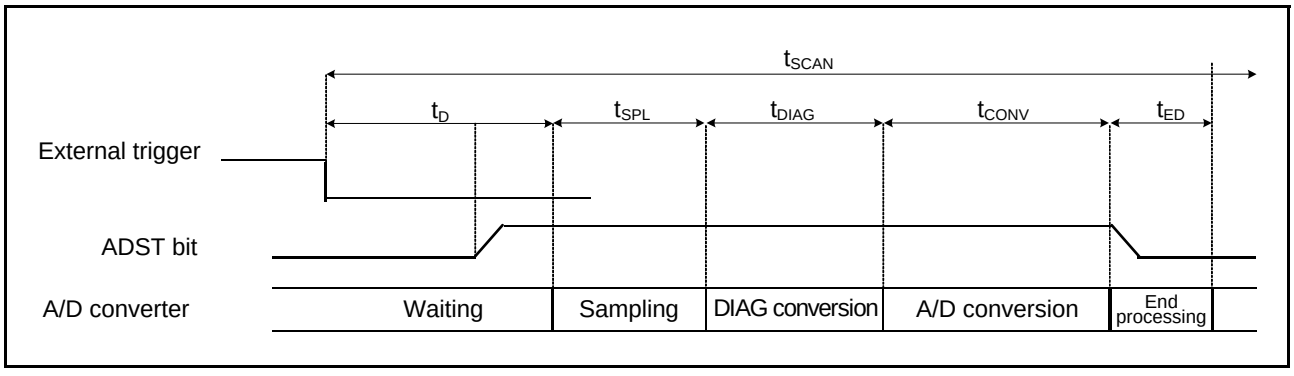


Figure 36.16 Scan Conversion Timing (Activated by ADTRG0#)

36.3.6 Usage Example of Automatic Register Clearing Function

Setting the ACE bit in ADCER to 1 automatically clears the A/D data registers (ADDRy, ADRD, and ADOCDR) to 0000h when the A/D data registers are read by the CPU, DTC, or DMAC.
This function enables detection of update failures of the A/D data registers. The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ACE bit in ADCER is 0 (automatic clearing is 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 value. Furthermore, if this ADDRy value is written to 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 ACE bit in ADCER is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy 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 update failure can be determined by simply checking that the read data value is 0000h.

36.3.7 A/D-Converted Value Addition Function

The same channel is A/D converted two to four consecutive times and the sum 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.
A/D converted value addition function can be used for channel-selected analog input A/D conversion and internal reference voltage A/D conversion.

36.3.8 Disconnection Detection Assist Function

The charge of the sampling capacitors can be fixed to the specified level (VREFH0 or VREFL0) before A/D conversion. This function enables detection of disconnection of the wires connected to the analog inputs.
Figure 36.17 shows the A/D conversion when the disconnection detection assist function is used. Figure 36.18 shows the example of disconnection detection at the VREFH0 (precharge is selected) and Figure 36.19 shows the example of disconnection detection at the VREFL0 side (discharge is selected).

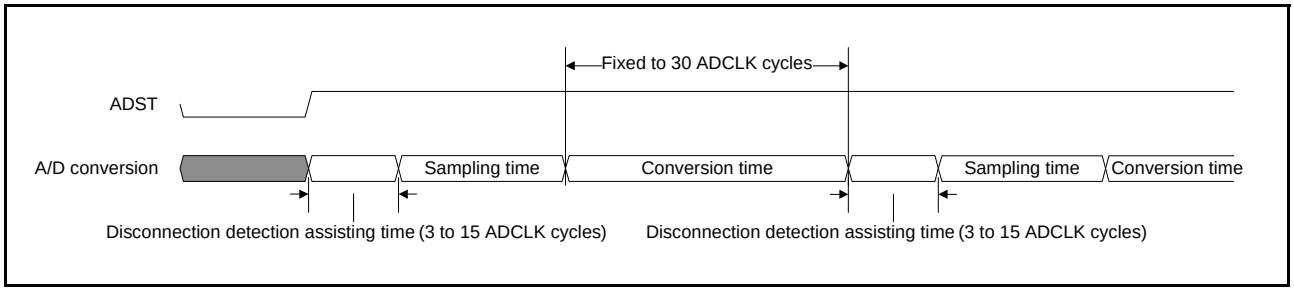
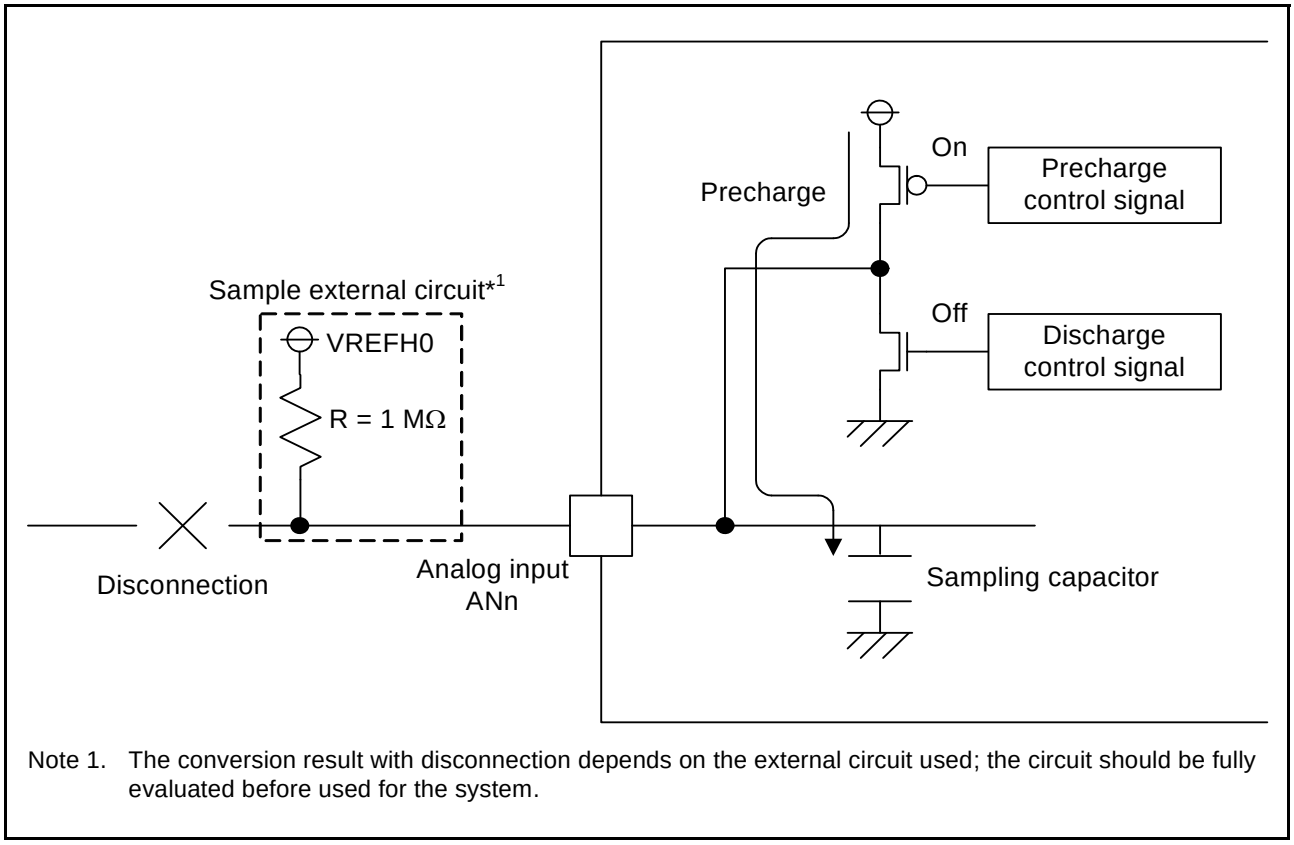
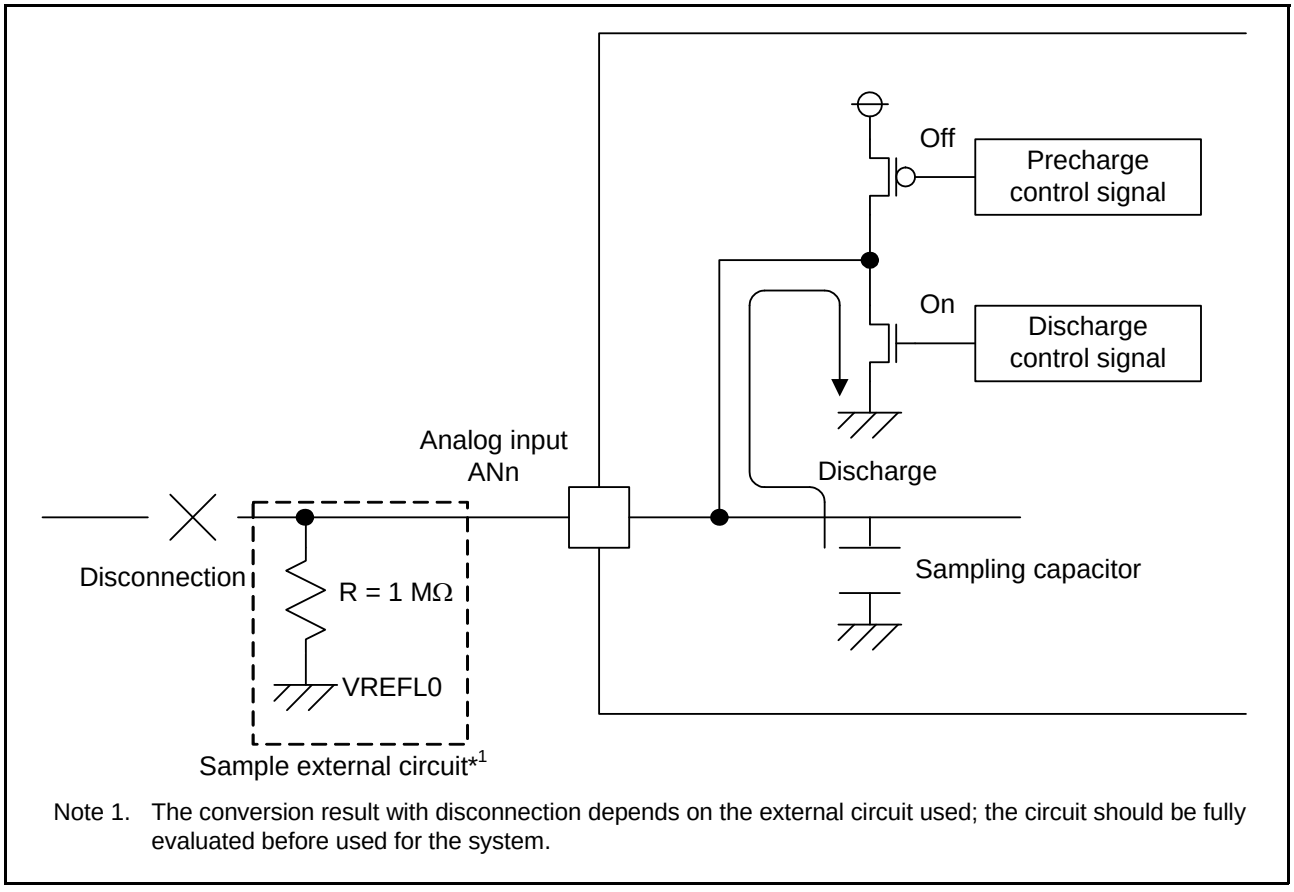


Figure 36.17 A/D Conversion with Disconnection Detection Assist Function Used



Note 1. The conversion result with disconnection depends on the external circuit used; the circuit should be fully evaluated before used for the system.

Figure 36.18 VREFH0 Disconnection Detection Example (Precharge Selected)



Note 1. The conversion result with disconnection depends on the external circuit used; the circuit should be fully evaluated before used for the system.

Figure 36.19 VREFL0 Disconnection Detection Example (Discharge Selected)

36.3.9 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[3:0]) should be set to 0000b and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin), and both the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 36.20 shows a timing of the asynchronous trigger input.
For the time required for the A/D conversion start after the ADCSR.ADST bit is set, refer to section 36.6.3, A/D Conversion Restarting Timing and Termination Timing.

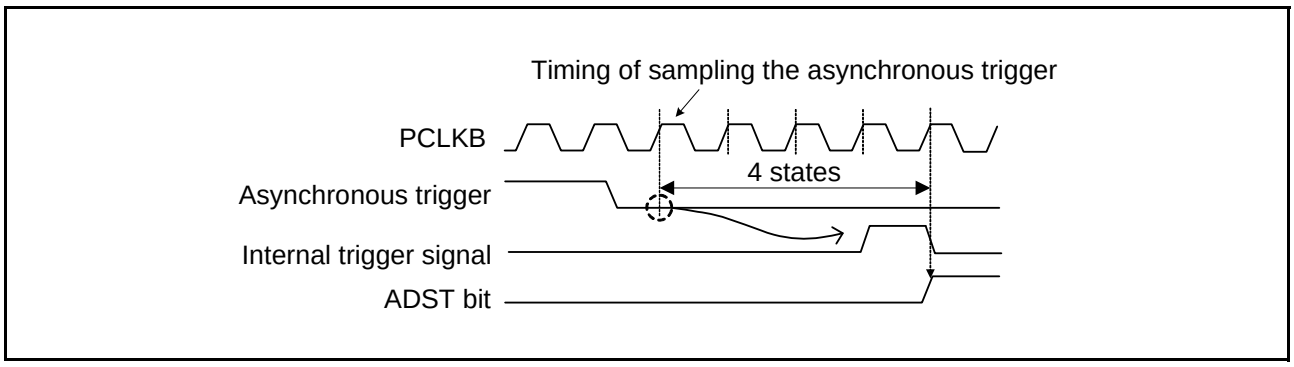


Figure 36.20 Asynchronous Trigger Input Timing

36.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

The A/D conversion can be started by a synchronous trigger of the MTU, TPU, or ELC. 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 source should be selected by ADSTRGR.TRSA[3:0] and TRSB[3:0] bits.

36.4 Interrupt Sources and DMA Transfer Requests

36.4.1 Interrupt Request on Completion of Each Scanning Conversion

The 12-bit A/D converter can send scan end interrupt requests S12ADI0 and GBADI to the CPU.
Setting the ADIE bit in ADCSR to 1 and 0 enables and disables an S12ADI0 interrupt, respectively; similarly, setting the GBADIE bit in ADCSR to 1 and 0 enables and disables a GBADI interrupt, respectively.
In addition, the DTC or DMAC can be started up when an S12ADI0 or a GBADI interrupt is generated. Using an S12ADI0 or a GBADI interrupt to allow the DTC or DMAC to read the converted data enables continuous conversion without burden on software.
For details on DTC settings, see section 18, Data Transfer Controller (DTCa), and for details on DMAC settings, see section 17, DMA Controller (DMACA).

36.5 Event Linkage

36.5.1 Event Output to ELC

The ELC connects the S12ADI0 interrupt request signal to the predetermined module as the event signal (i.e., event linkage). The GBADI interrupt request signal cannot be used as the event signal. The event signal can be output irrespective of the setting of the corresponding interrupt request enable bit. The 12-bit A/D converter outputs the A/D conversion end event.

36.5.2 12-bit A/D converter Operation by Event from ELC

The 2-bit A/D converter can be started by the predetermined event by so setting ELSRn of the ELC.

36.5.3 Notes on Event Reception from ELC during 12-bit A/D Conversion

When an event occurs during A/D conversion, it is invalid.

36.6 Usage Notes

36.6.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication register, A/D internal reference voltage data register, and A/D self-diagnosis data register should be read in word units. If a register is read twice in byte units, that is, the upper byte and lower 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 byte units.

36.6.2 Notes on 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, set the TRGE bit in ADCSR to 0 and select the software trigger as the condition for starting A/D conversion, and then set the ADST bit in ADCSR to 0 (to stop A/D conversion).

36.6.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of four ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADST bit in ADCSR to 1. It takes a maximum of two ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADST bit in ADCSR to 0.

36.6.4 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 out 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.

36.6.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Canceling the module stop state allows registers to be accessed.
For details, see section 11, Low Power Consumption.

36.6.6 Notes on Entering Low Power Consumption States

Before entering all-module clock stop mode or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and allow time for stopping the analog unit of the 12-bit A/D converter. Follow the procedure given below to secure this time.
1. Set the ADCSR.TRGE bit to 0 (software trigger).
2. Clear the ADCSR.ADST bit to 0.
3. After confirming that the A/D converter has been stopped, place the LSI in all-module clock stop mode or software standby mode.

When a transition is made to all-stop module clock stop mode, software standby mode, or deep software standby mode, the 12-bit A/D converter partially enters the wait state for operation. To place the 12-bit A/D converter fully in the standby state, set the MSTPCRA.MSTPA24 bit to 1. In this case, wait for a further 10 ms to start A/D conversion after release from all-module clock stop mode, software standby, or deep software standby.

36.6.7 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.0 µs, the analog input pins of this LSI are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 1 kΩ or less. If an external capacitor of large capacitance is attached in the application in which only a single pin input is converted in single scan mode, the only load on input is virtually 10 kΩ of the internal input resistor; therefore, the impedance of the signal source can be ignored. Being a low-pass filter, however, an analog input circuit may not follow the analog signal with a large differential coefficient (e. g., larger than 5 mV/µs) as shown in Figure 36.21. When high-speed analog signals are to be converted or multiple pins are to be converted in scan mode, a low-impedance buffer should be used.

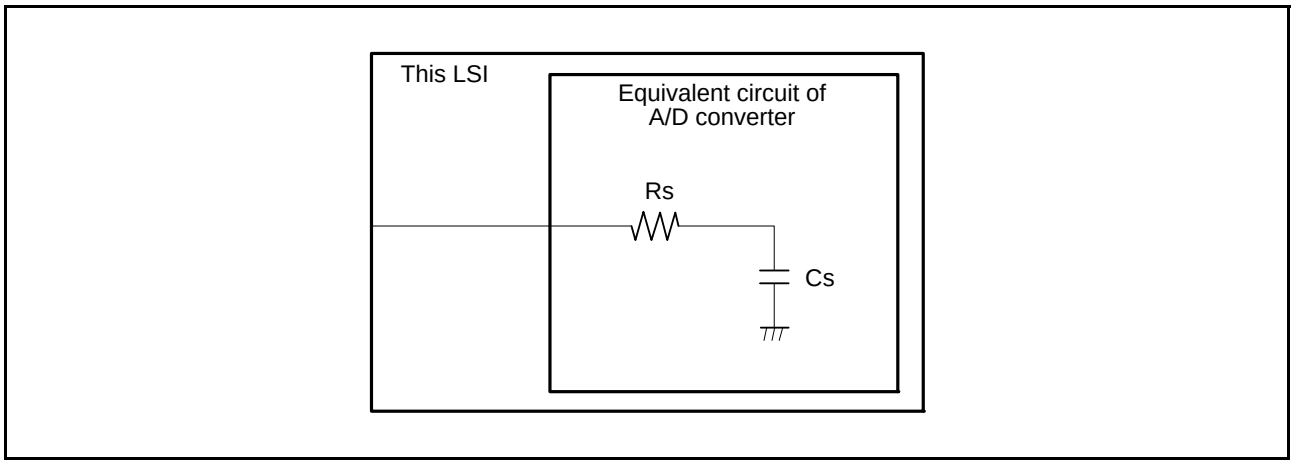


Figure 36.21 Internal Equivalent Circuit of Analog Input Pin

Table 36.9 Specifications of Analog Input Pins

Item		Min.	Max.	Unit
Allowable signal source impedance*1		—	1	kΩ
Internal equivalent circuit of a pin	Rs	—	10	kΩ
	Cs	—	10	pF

Note 1. The value differs depending on the analog power supply voltage and analog input pins to be used. For details, see section 41, Electrical Characteristics.

36.6.8 Influence on Absolute Accuracy

Attaching a capacitor creates coupling with GND and may affect the absolute accuracy when noisy GND is used; therefore, a capacitor should be connected to electrically stable GND such as AVSS0.
The filter circuit should be designed so that it does not interfere digital signals or it does not serve as an antenna on the circuit board.

36.6.9 Voltage Range of Analog Power Supply Pins

If this LSI is used with the voltages outside the following ranges, the reliability of the LSI may be affected.

- Analog input voltage range
  Voltage (VAN) applied to analog input pins ANn: VREFL0 ≤ VAN ≤ VREFH0
- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  AVCC0 = VCC and AVSS0 = VSS
  A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest rout possible as shown in Figure 36.22, and connection should be made so that the following conditions are satisfied at the supply side.
  VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS
  When the A/D converter is not used, the following conditions should be satisfied.
  VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS

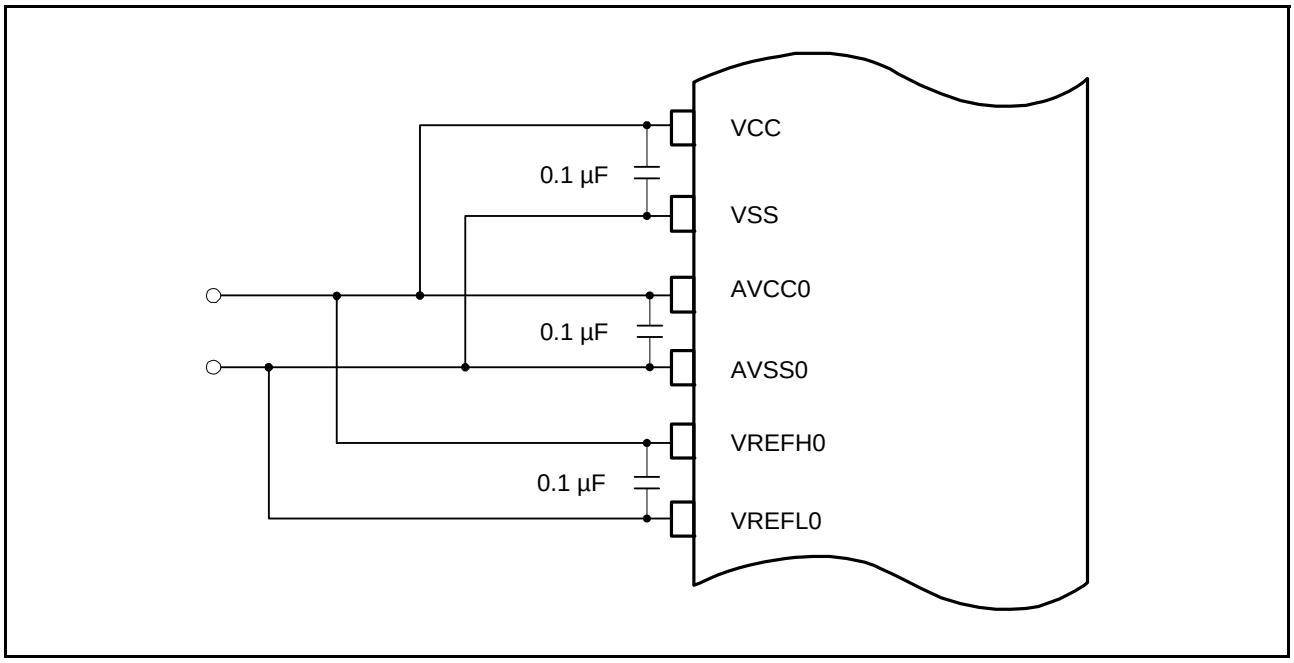


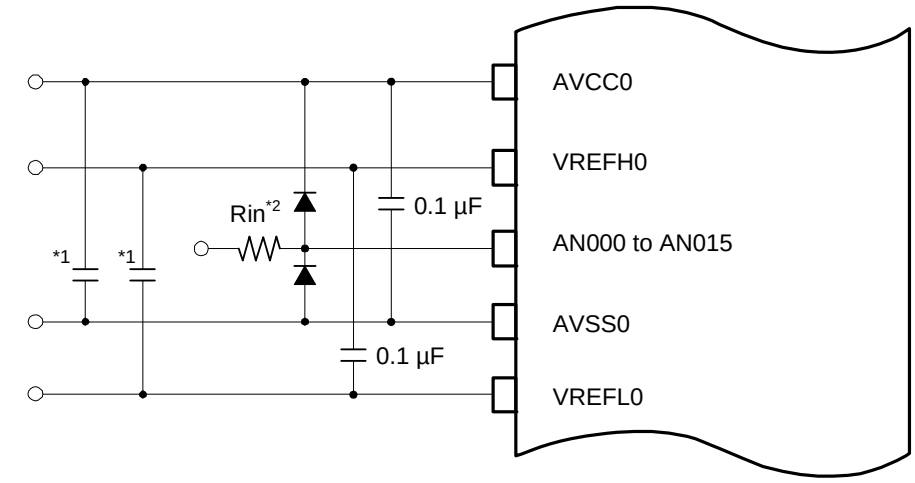
Figure 36.22 Power Supply Pin Connection Example

36.6.10 Notes on Board Design

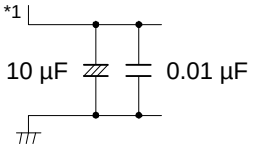
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN015), analog reference voltages (VREFH0 and VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

36.6.11 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN015) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN015) as shown Figure 36.23.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 36.23 Sample Protection Circuit for Analog Inputs

36.6.12 Error in Absolute Accuracy when Disconnection Detection Assist Function is Used

When the disconnection detection assist function is used, the error voltage is input to the analog input pins, thus affecting the absolute accuracy of the A/D converter. The error voltage here is caused by dividing the voltage by the pull-up/pull-down resistor (Rp) and signal source resistor (Rs). The error in the absolute accuracy is represented by the expression below. When using the disconnection detection assist function, fully evaluate the system in terms of the error in absolute accuracy.

Maximum error in the absolute accuracy (LSB) = 4095 × Rs/Rp

36.6.13 Notes on Using External Buses

An A/D conversion during access to an external bus may deteriorate accuracy.
In this case, perform conversion multiple times and use software to average out the A/D converted values excluding the maximum and minimum values.

37. D/A Converter (DA)

In this section, “PCLK” is used to refer to PCLKB.

37.1 Overview

This MCU includes two channels of 10-bit D/A converter.
Table 37.1 lists the specifications of the 10-bit D/A converter and Figure 37.1 shows a block diagram of the 10-bit D/A converter.

Table 37.1 Specifications of 10-Bit D/A Converter

Item	Specifications
Resolution	10 bits
Output channels	Two channels
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.

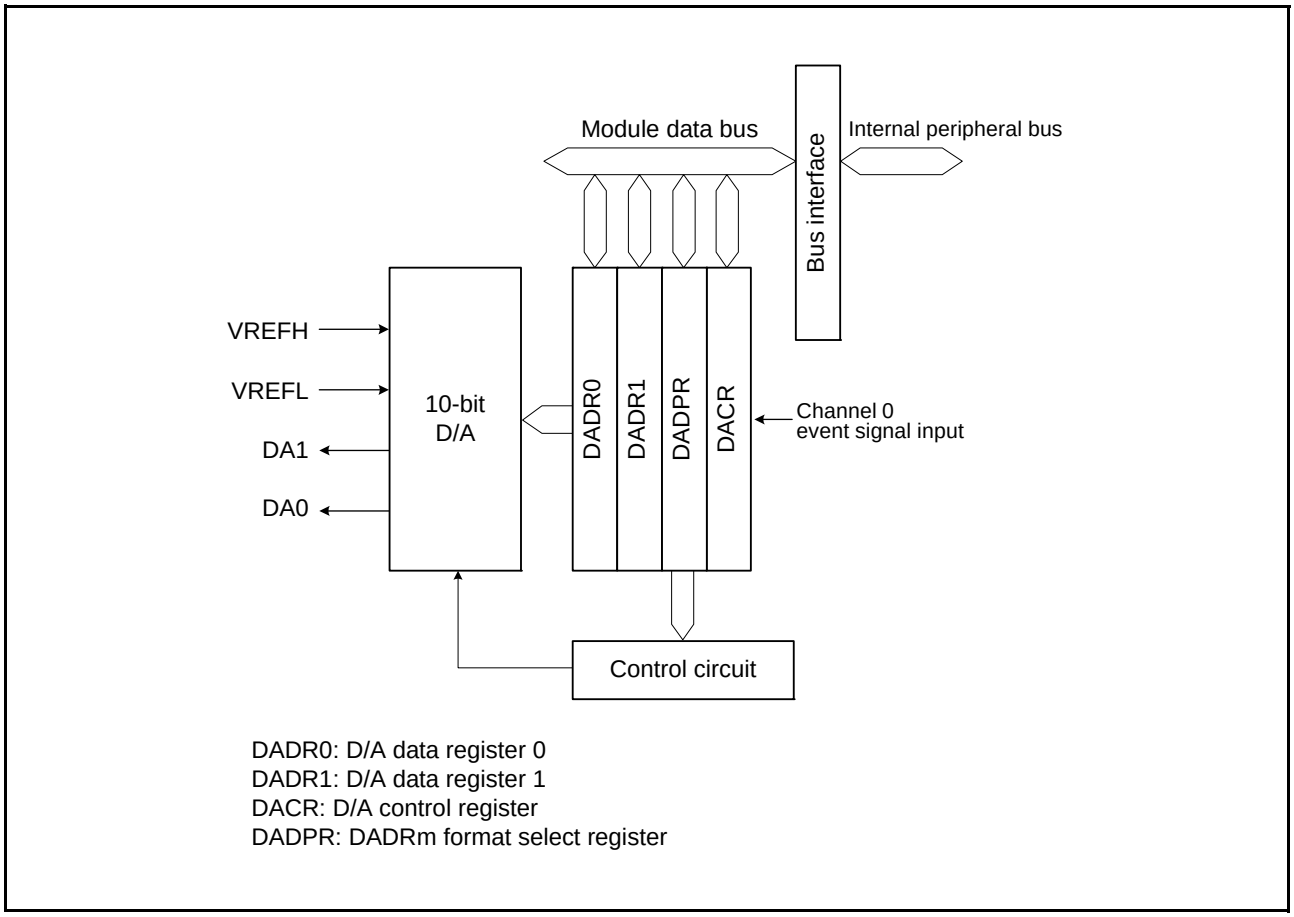


Figure 37.1 Block Diagram of 10-Bit D/A Converter

Table 37.2 lists the pin configuration of the 10-bit D/A converter.

Table 37.2 Pin Configuration of 10-Bit D/A Converter

Pin Name	I/O	Function
VREFH	Input	Reference voltage input pin for the D/A converter. This pin is also used as an analog power supply pin. Connect to VCC when the D/A converter is not used.
VREFL	Input	Reference voltage input pin for the D/A converter. This pin is also used as an analog ground pin. Set VREFL to the same potential as VSS.
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

37.2 Register Descriptions

37.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DADR0 0008 80C0h, DADR1 0008 80C2h

• DADPR.DPSEL bit = 0 (data is flush with the right end of the register)

	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 flush with the left end of the register)

	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

DADRm registers are 16-bit readable/writable registers, which store data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins.
10-bit data can be relocated by setting the DADPR.DPSEL bit.
Bits "—" are read as 0. The write value should be 0.

37.2.2 D/A Control Register (DACR)

Address(es): 0008 80C4h

	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 DAOEi bit (i = 0, 1). The DAOEi bit controls output of the results of conversion. For details, see Table 37.3.

Table 37.3 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	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled collectively.*1
		1	D/A conversion of channel 0 is enabled. 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. 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.

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion in combination with the DAOEi (i = 0, 1) bit.

When the DAE bit is 0, D/A conversion is independently controlled on channels 0 and 1. When the DAE bit is 1, D/A conversion on channels 0 and 1 is controlled as a single whole. The DAOEi bit controls output of the results of conversion.

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion and analog output.

The event link function is capable of setting the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by the setting of the ELSR16 register of the ELC occurs, and output of the results of D/A conversion starts.

DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls the D/A conversion and analog output.

37.2.3 DADRm Format Select Register (DADPR)

Address(es): 0008 80C5h

	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	DADRm Format Select	0: Data is flush with the right end of the D/A data register. 1: Data is flush with the left end of the D/A data register.	R/W

37.3 Operation

The 10-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in DACR 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 37.2 shows the timing of this operation.

1. Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCONV has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting value of DADRm}}{1024} \times \text{VREFH}$$

3. If DADR0 is written to again, the conversion is started. The conversion result is output after the conversion time tDCONV has elapsed.
4. If the DAOE0 bit is set to 0, analog output is disabled.

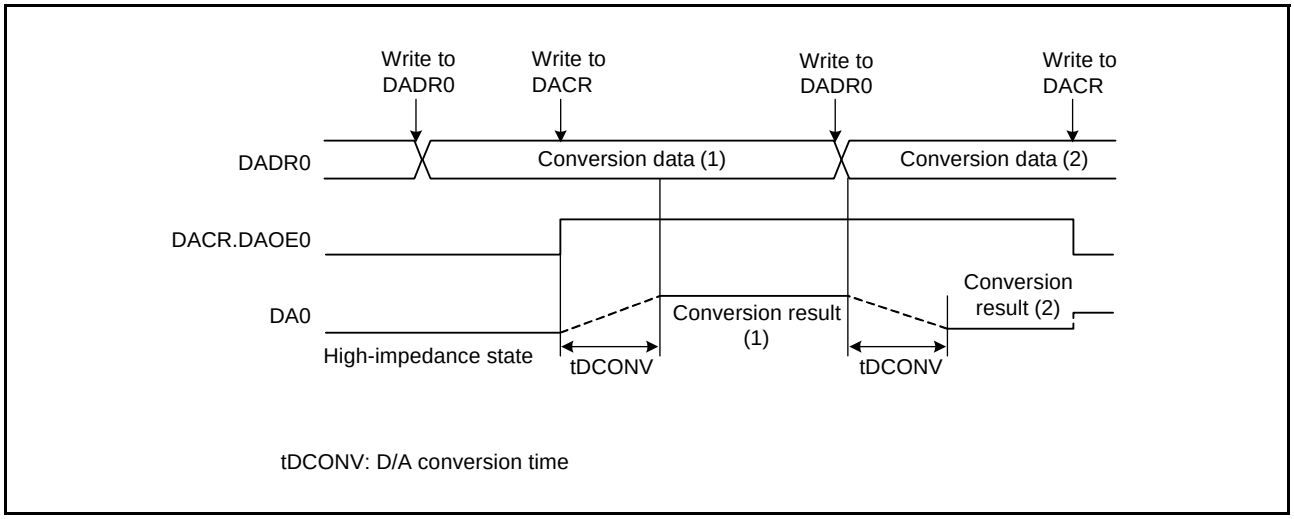


Figure 37.2 Example of 10-Bit D/A Converter Operation

37.4 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. Set the bit value of the ELS16 setting event signal to link the ELSR16 register of the ELC.
3. Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
4. 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.
5. Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 10-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

37.5 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.

37.6 Usage Notes

37.6.1 Module Stop Function Setting

Operation of the 10-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 10-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.

37.6.2 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 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.

37.6.3 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 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.

37.6.4 Note on Entering Deep Software Standby Mode

When the MCU enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

38. Data Operation Circuit (DOC)

38.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.
Table 38.1 lists the data operation circuit specifications and Figure 38.1 shows a block diagram of the data operation circuit.
16-bit data is compared and an interrupt can be generated when a selected condition applies.

Table 38.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module stop state can be set.
Interrupts	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

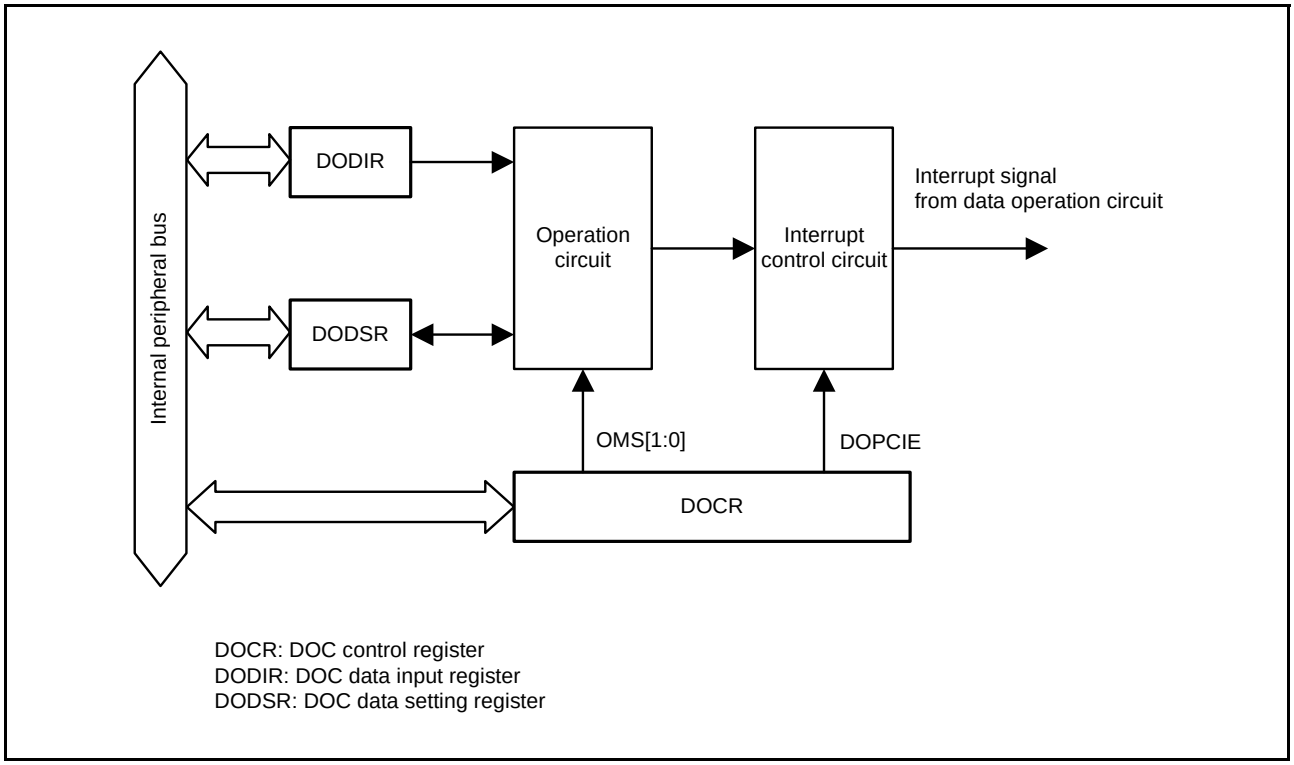


Figure 38.1 DOC Block Diagram

38.2 Register Descriptions

38.2.1 DOC Control Register (DOCR)

Address(es): 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

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	DCSEL*1	Detection Condition Select	Result of data comparison 0: Data mismatch is detected. 1: Data match is detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation.	R
b6	DOPCFCL	DOPCF Clear	0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DCSEL Bit (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 data operation circuit.

DOPCF Flag (Data Operation Circuit Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL Bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

This bit is read as 0.

38.2.2 DOC Data Input Register (DODIR)

Address(es): 0008 B082h

	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

DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

38.2.3 DOC Data Setting Register (DODSR)

Address(es): 0008 B084h

	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

DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

38.3 Operation

38.3.1 Data Comparison Mode

Figure 38.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

(1) Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
(2) The 16-bit reference data is set in DODSR.
(3) 16-bit data for comparison is written to DODIR.
(4) Writing of 16-bit data continues until all data for comparison have been written to DODIR.
(5) If a value written to DODIR does not match that in DODSR*1, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

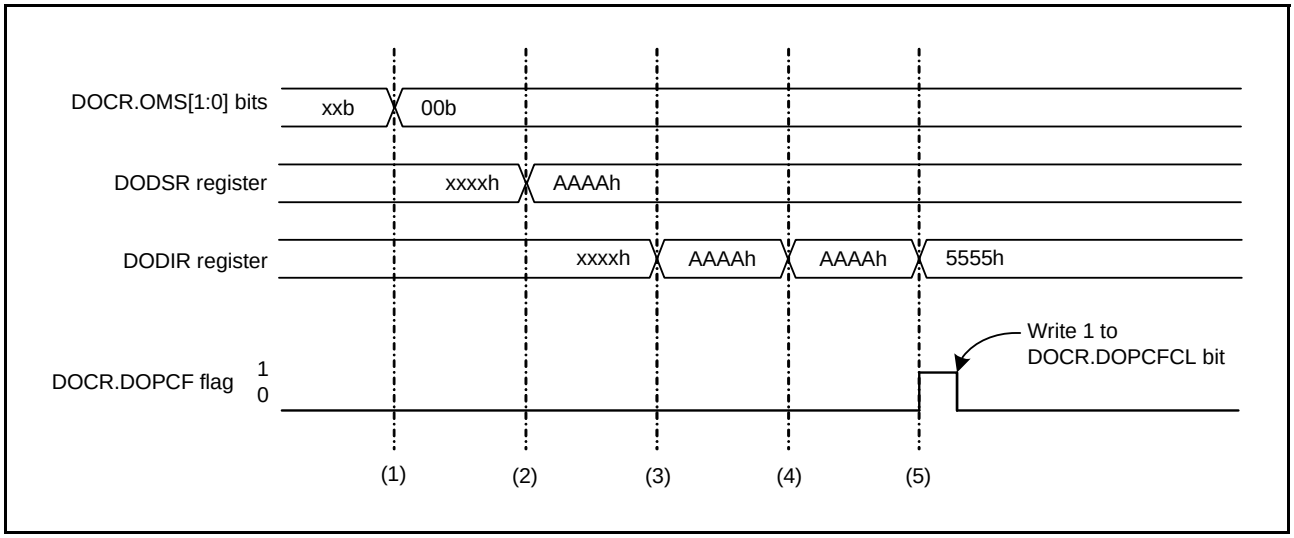


Figure 38.2 Example of Operation in Data Comparison Mode

38.3.2 Data Addition Mode

Figure 38.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for addition have been written to DODIR.
(5) If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

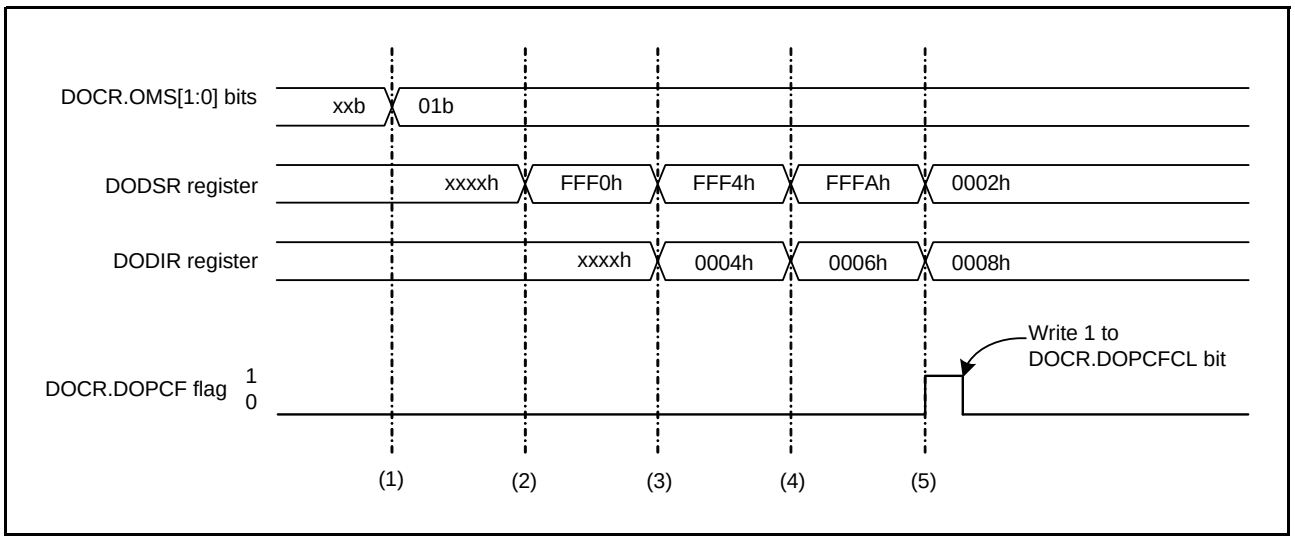


Figure 38.3 Example of Operation in Data Addition Mode

38.3.3 Data Subtraction Mode

Figure 38.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
(5) If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

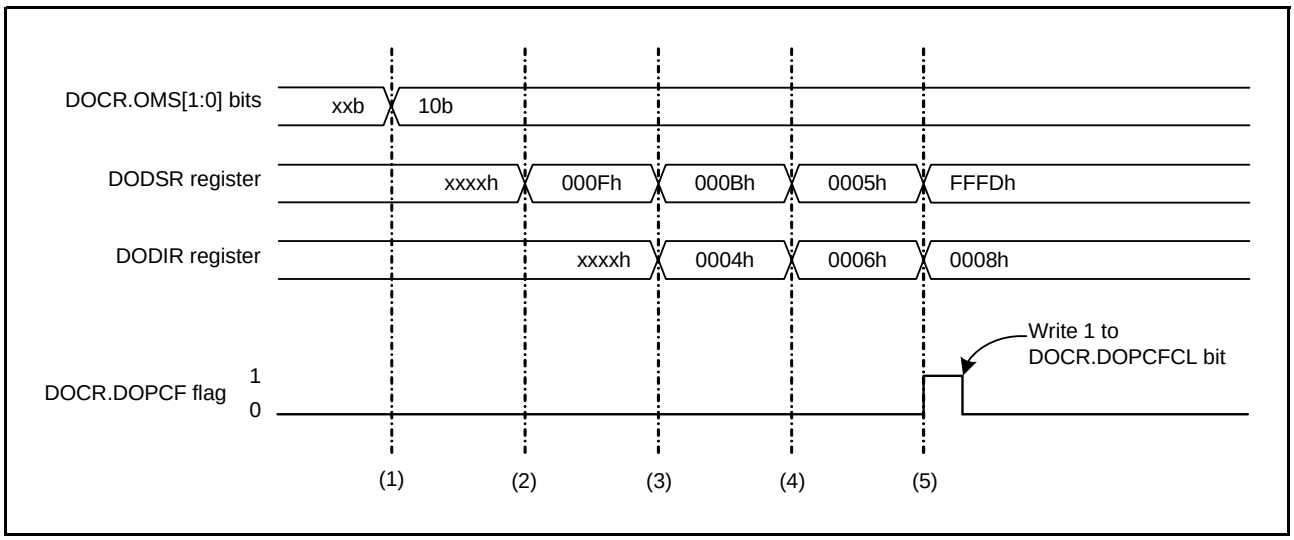


Figure 38.4 Example of Operation in Data Subtraction Mode

38.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1. Table 38.2 describes the interrupt request.

Table 38.2 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Generation Timing
Data operation circuit interrupt	DOPCF	• The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

38.5 Usage Note

38.5.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

39. RAM

This MCU has an on-chip high-speed static RAM.

39.1 Overview

Table 39.1 lists the specifications of the RAM.

Table 39.1 Specifications of RAM

Item	Description
RAM capacity	128 Kbytes (RAM0: 64 Kbytes, RAM1: 64 Kbytes)
RAM address	RAM0: 0000 0000h to 0000 FFFFh (64 Kbytes) RAM1: 0001 0000h to 0001 FFFFh (64 Kbytes)
Access	• Single-cycle access is possible for both reading and writing. • RAM can be enabled or disabled.*1
Data retention function	Data in RAM0 can be retained in deep software standby mode.
Low-power consumption function	The module-stop state is independently selectable for RAM0 and RAM1.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

39.2 Operation

39.2.1 Data Retention

The address space for RAM is divided into the RAM0 and RAM1 areas. The difference between the two is whether internal power can be supplied in deep software standby mode.
Whether or not the supply of internal power to RAM0 continues in deep software standby mode is selectable by the DPSBYCR.DEEPCUT[1:0] bits.
If continuation of the supply of internal power is selected, data in RAM0 are retained in deep software standby mode. The supply of internal power supply to RAM1 is stopped at this time, so data are not retained in RAM1.
See section 11, Low Power Consumption, for details on the DPSBYCR.DEEPCUT[1:0] bits.

39.2.2 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 RAM.
If the MSTPC0 bit in MSTPCRC is set to 1, supply of the clock signal to RAM0 is stopped. If the MSTPC1 bit in MSTPCRC is set to 1, supply of the clock signal to RAM1 is stopped.
The respective modules (RAM0 and RAM1) are thus placed in the module-stop state by stopping supply of the clock signals. The RAM operates after a reset.
RAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to RAM is in progress.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

40. Flash Memory

This MCU has a maximum 2-Mbyte flash memory for storing code (ROM) and a 32-Kbyte flash memory for storing data (E2 DataFlash). P/E indicates programming/erasure in this section.

40.1 Overview

Table 40.1 lists the specifications of the ROM and E2 DataFlash memory, and Figure 40.1 is a block diagram of the ROM, E2 DataFlash memory, and related modules.
Regarding the configuration of the ROM memory area, refer to section 40.1.1, Configuration of the ROM Area, and for the configuration of the memory area for the E2 DataFlash, refer to section 40.1.3, Configuration of the E2 DataFlash Area.

Table 40.1 Specifications of ROM/E2 DataFlash

Item		ROM	E2 DataFlash
Memory space		User area: maximum 2-Mbyte User boot area: 16 Kbytes	Data area: 32 Kbytes
Read cycle		A read operation takes one cycle of ICLK	A read operation takes six cycles of FCLK in words or bytes
Programming/erasing method		• The chip incorporates a dedicated sequencer (FCU) for programming of the ROM/E2 DataFlash. • Programming and erasing the ROM/E2 DataFlash are handled by issuing commands to the FCU.	
Value after erasure		FFh	Undefined
BGO (background operation)		• The CPU is able to execute program code from the ROM while the E2 DataFlash memory is being programmed or erased.	
Suspension and resumption		• The CPU is able to execute program code from the ROM during suspension of programming or erasure. • The CPU is able to execute program code from the E2 DataFlash during suspension of programming or erasure. • Programming and erasure of the ROM/E2 DataFlash can be restarted (resumed) after suspension.	
Units of programming and erasure		• Units of programming for the user area or user boot area: 128 bytes • Units of erasure for the user area: In block units • Units of erasure for the user boot area: 16 Kbytes	• Unit of programming for the data area: 2 bytes • Unit of erasure for the data area: 32 bytes (1024 or 256 blocks)
On-board programming (three types)		Programming in boot mode • The asynchronous serial interface (SCI1) is used. • The transfer rate is adjusted automatically. • The user boot area can also be programmed. Programming in the user boot mode • Able to create original boot programs of the user's making. Programming by a routine for ROM/E2 DataFlash programming within the user program • This allows ROM/E2 DataFlash programming without resetting the system.	
Off-board programming		A Flash programmer can be used to program the user area and user boot area.	A Flash programmer cannot be used to program the data area.
Protection	Software-controlled protection	The registers and lock bits can be set to prevent unintentional programming.	• The registers can be used to prevent unintentional programming or reading. • Protection with the registers is performed on a 2-Kbyte basis.
	FCU command-lock	When abnormal operations are detected during programming/erasure, this function disables any further programming/erasure.	

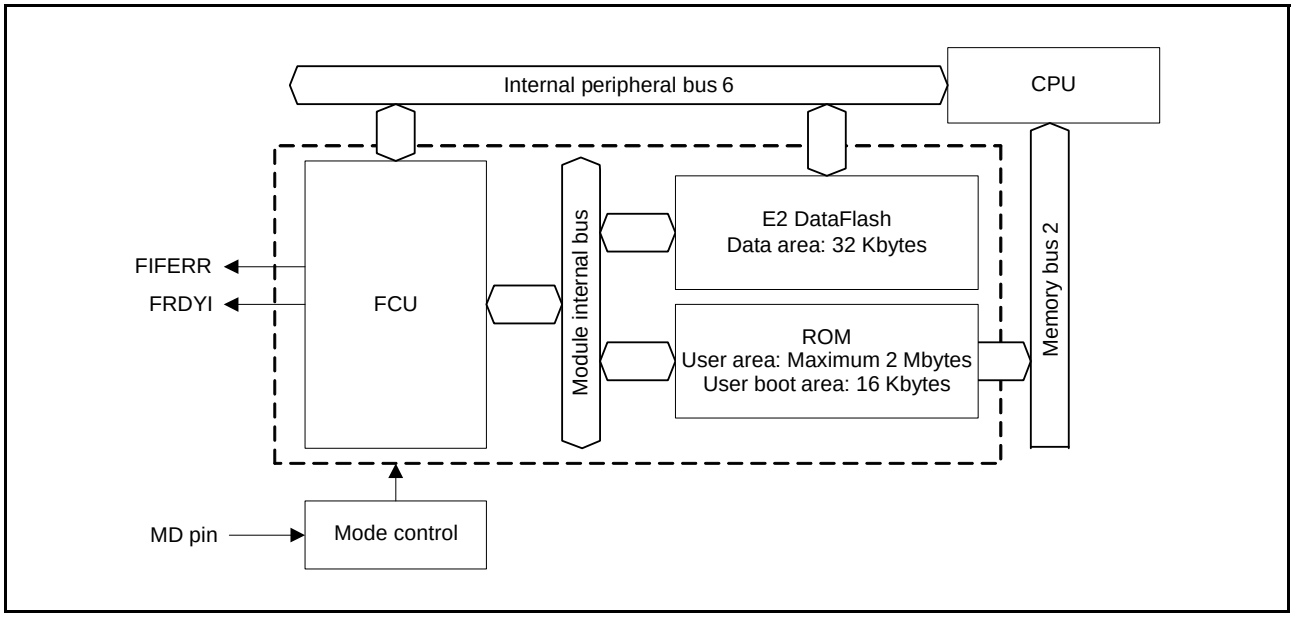


Figure 40.1 Block Diagram of ROM/E2 DataFlash

40.1.1 Configuration of the ROM Area

The ROM of products in this MCU is configured of a maximum 2-Mbyte user area and a 16-Kbyte user boot area. Figure 40.2 shows the address ranges of these areas.
Note that for the user area, the address range for reading differs from the address range for programming and erasure.

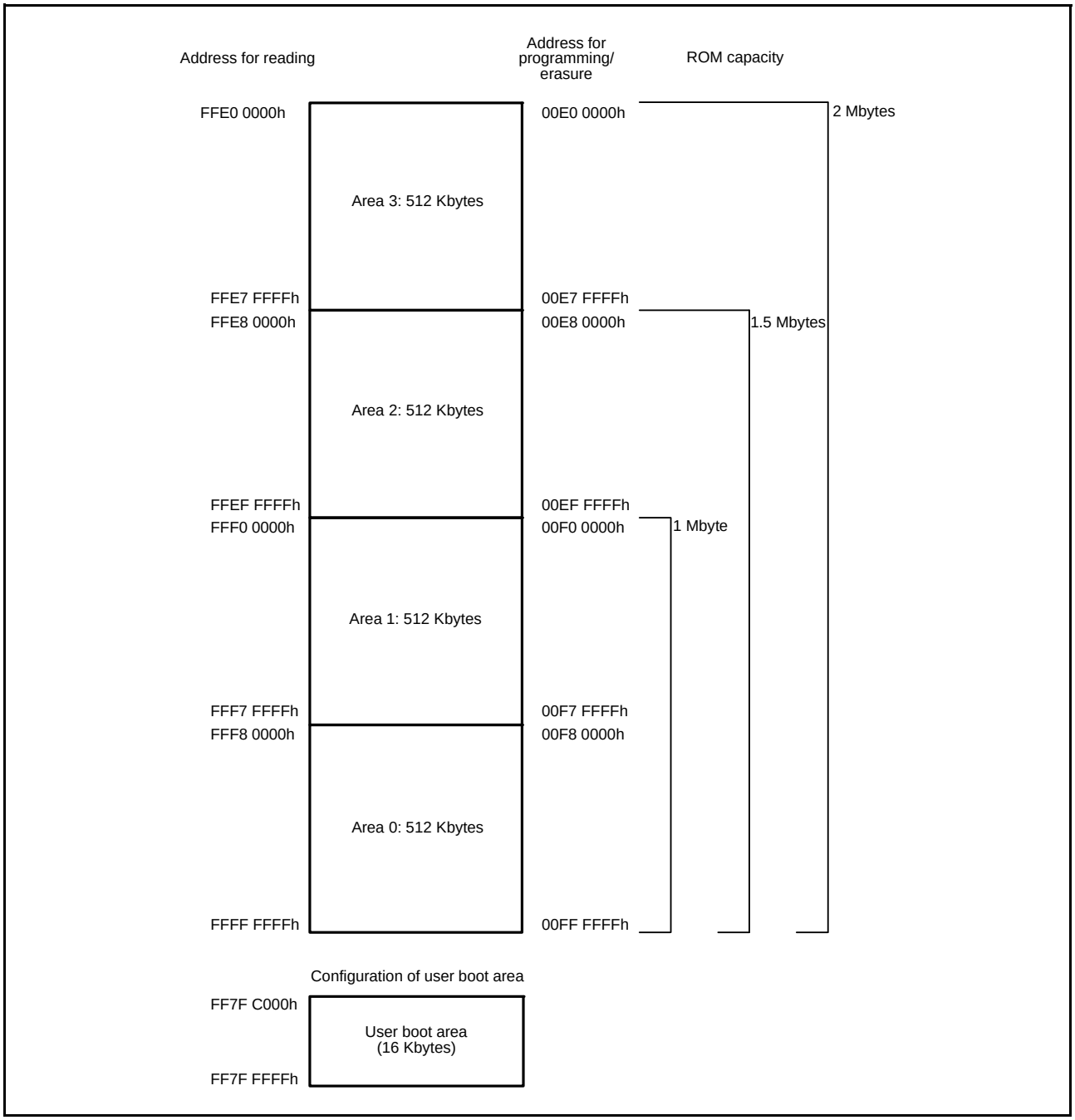


Figure 40.2 Memory Area Configuration of ROM

40.1.2 Block Configuration of the ROM

The user area (area 0 to area 3) is divided into blocks with different sizes, and erasure proceeds in block units. The configuration of the blocks of the user area is shown in Figure 40.3, and the relations between blocks and addresses of the user area are listed in Table 40.2.

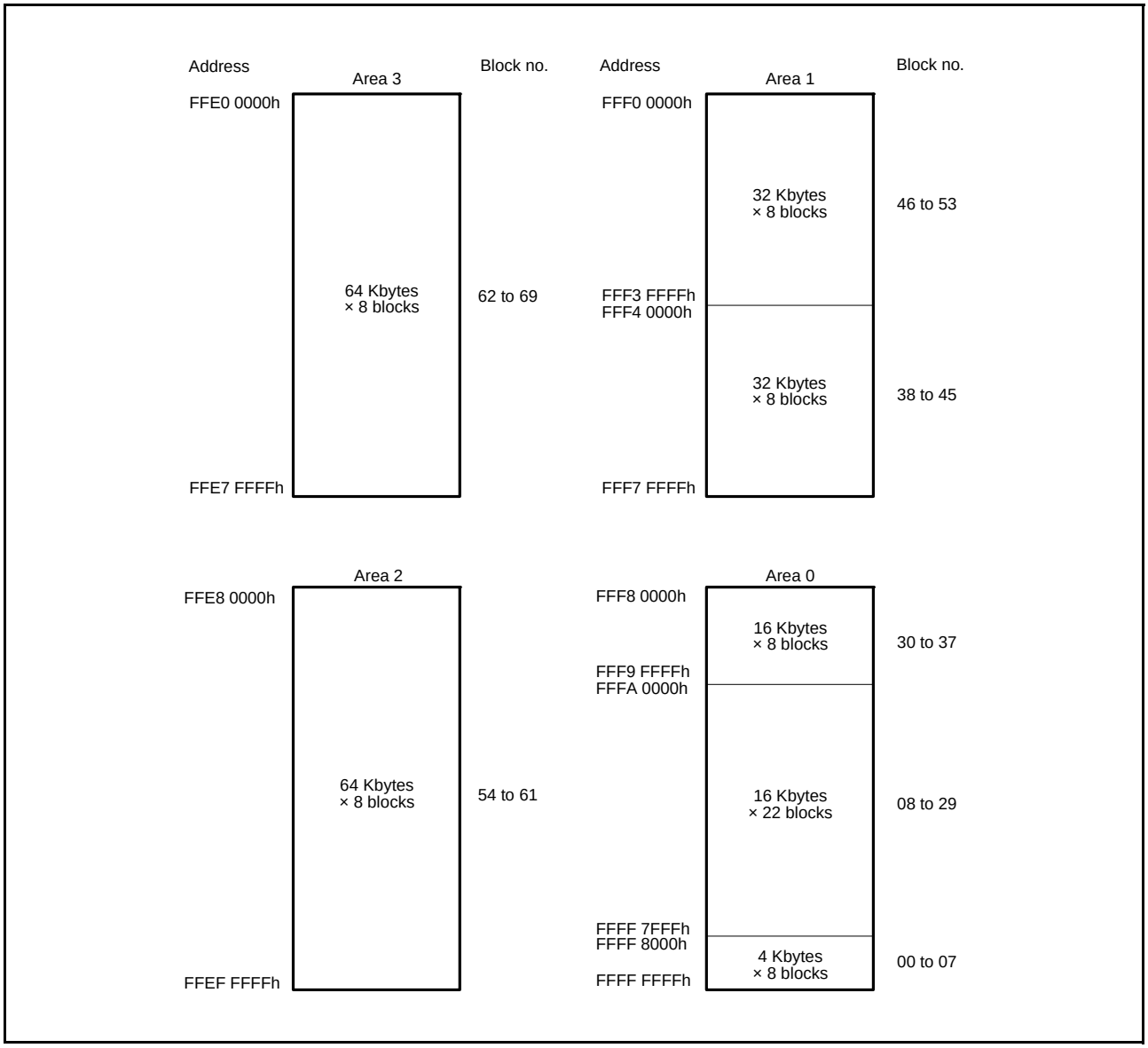


Figure 40.3 Block Configuration of the User Area

Table 40.2 Correspondence between Blocks and Addresses of the User Area

Block No.	Start Address	Block Configuration	Area
69	FFE0 0000h	64 Kbyte × 8 blocks	Area 3
68	FFE1 0000h		
67	FFE2 0000h		
66	FFE3 0000h		
65	FFE4 0000h		
64	FFE5 0000h		
63	FFE6 0000h		
62	FFE7 0000h		
61	FFE8 0000h	64 Kbyte × 8 blocks	Area 2
60	FFE9 0000h		
59	FFEA 0000h		
58	FFEB 0000h		
57	FFEC 0000h		
56	FFED 0000h		
55	FFEE 0000h		
54	FFEF 0000h		
53	FFF0 0000h	32 Kbyte × 8 blocks	Area 1
52	FFF0 8000h		
51	FFF1 0000h		
50	FFF1 8000h		
49	FFF2 0000h		
48	FFF2 8000h		
47	FFF3 0000h		
46	FFF3 8000h		
45	FFF4 0000h	32 Kbyte × 8 blocks	
44	FFF4 8000h		
43	FFF5 0000h		
42	FFF5 8000h		
41	FFF6 0000h		
40	FFF6 8000h		
39	FFF7 0000h		
38	FFF7 8000h		

Block No.	Start Address	Block Configuration	Area
37	FFF8 0000h	16 Kbyte × 8 blocks	Area 0
36	FFF8 4000h		
35	FFF8 8000h		
34	FFF8 C000h		
33	FFF9 0000h		
32	FFF9 4000h		
31	FFF9 8000h		
30	FFF9 C000h		
29	FFFA 0000h	16 Kbyte × 22 blocks	
28	FFFA 4000h		
27	FFFA 8000h		
26	FFFA C000h		
25	FFFB 0000h		
24	FFFB 4000h		
23	FFFB 8000h		
22	FFFB C000h		
21	FFFC 0000h		
20	FFFC 4000h		
19	FFFC 8000h		
18	FFFC C000h		
17	FFFD 0000h		
16	FFFD 4000h		
15	FFFD 8000h		
14	FFFD C000h		
13	FFFE 0000h		
12	FFFE 4000h		
11	FFFE 8000h		
10	FFFE C000h		
9	FFFF 0000h		
8	FFFF 4000h		
7	FFFF 8000h	4 Kbyte × 8 blocks	
6	FFFF 9000h		
5	FFFF A000h		
4	FFFF B000h		
3	FFFF C000h		
2	FFFF D000h		
1	FFFF E000h		
0	FFFF F000h		

40.1.3 Configuration of the E2 DataFlash Area

The E2 DataFlash memory of products in this MCU is configured as a 32-Kbyte data area whose addresses range from 0010 0000h to 0010 7FFFh.
Unlike the user area, the address ranges for reading from and programming and erasure of the data area are the same.

40.1.4 Block Configuration of the E2 DataFlash

The data area is divided into 1024 blocks, and erasure is executed in block units. The relations between blocks and addresses of the data area and the corresponding bits to support permission of reading and of programming and erasure are listed in Table 40.3.
The address of block N (address where block N starts) is calculated from the following formula.
Address of block N = (N × 32) + address where the data area starts (0010 0000h).

Table 40.3 Block Configuration of the Data Area

Block No.	Start Address	Reading and Programming/ Erasure Enable Bit
0000 ⋮ 0063	0010 0000h ⋮ 0010 07E0h	DFLRE0.DBRE00 DFLWE0.DBWE00
0064 ⋮ 0127	0010 0800h ⋮ 0010 0FE0h	DFLRE0.DBRE01 DFLWE0.DBWE01
0128 ⋮ 0191	0010 1000h ⋮ 0010 17E0h	DFLRE0.DBRE02 DFLWE0.DBWE02
0192 ⋮ 0255	0010 1800h ⋮ 0010 1FE0h	DFLRE0.DBRE03 DFLWE0.DBWE03
0256 ⋮ 0319	0010 2000h ⋮ 0010 27E0h	DFLRE0.DBRE04 DFLWE0.DBWE04
0320 ⋮ 0383	0010 2800h ⋮ 0010 2FE0h	DFLRE0.DBRE05 DFLWE0.DBWE05
0384 ⋮ 0447	0010 3000h ⋮ 0010 37E0h	DFLRE0.DBRE06 DFLWE0.DBWE06
0448 ⋮ 0511	0010 3800h ⋮ 0010 3FE0h	DFLRE0.DBRE07 DFLWE0.DBWE07
0512 ⋮ 0575	0010 4000h ⋮ 0010 47E0h	DFLRE1.DBRE08 DFLWE1.DBWE08
0576 ⋮ 0639	0010 4800h ⋮ 0010 4FE0h	DFLRE1.DBRE09 DFLWE1.DBWE09
0640 ⋮ 0703	0010 5000h ⋮ 0010 57E0h	DFLRE1.DBRE10 DFLWE1.DBWE10
0704 ⋮ 0767	0010 5800h ⋮ 0010 5FE0h	DFLRE1.DBRE11 DFLWE1.DBWE11
0768 ⋮ 0831	0010 6000h ⋮ 0010 67E0h	DFLRE1.DBRE12 DFLWE1.DBWE12
0832 ⋮ 0895	0010 6800h ⋮ 0010 6FE0h	DFLRE1.DBRE13 DFLWE1.DBWE13
0896 ⋮ 0959	0010 7000h ⋮ 0010 77E0h	DFLRE1.DBRE14 DFLWE1.DBWE14
0960 ⋮ 1023	0010 7800h ⋮ 0010 7FE0h	DFLRE1.DBRE15 DFLWE1.DBWE15

40.2 Register Descriptions

Some registers are common to the ROM and the E2 DataFlash, while others are dedicated to one or the other.

40.2.1 Flash Write Erase Protection Register (FWEPROR)

Address(es): 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 to b0	FLWE[1:0]	Flash Programming/ Erasure	b1 b0 0 0: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 0 1: Enables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 1 0: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 1 1: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Programming and erasure of the ROM or E2 DataFlash memory, programming and erasure of lock bits, reading of lock bits, and blank checking by software are prohibited.

FWEPROR is initialized by a reset due to the signal on the RES# pin, by transitions to software standby and deep software standby, and by the power supply voltage falling below the threshold for detection.

40.2.2 Flash Mode Register (FMODR)

Address(es): 007F C402h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FRDMD	—	—	—	—
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	FRDMD	FCU Read Method Select	This bit selects internal processing by the FCU when a 0x71 command is issued. For details, see “FRDMD Bit (FCU Read Method Select)”.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Regarding the FMODR, when a 0x71 command is issued in relation to the FCU, this bit selects internal processing by the FCU. Internal processing by the FCU differs according to the address where the 0x71 command was issued.
This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

FRDMD Bit (FCU Read Method Select)

Table 40.4 shows the relations between the status of the FRDMD bit and internal processing by the FCU when a 0x71 command is issued.

Table 40.4 Correspondence between the FRDMD Bit and 0x71 Command

Where the 0x71 Command is Issued	FRDMD Bit Status	Function	Internal Processing by the FCU
ROM	0	Lock bits are read using the memory area read method (lock bit read 1)	Transition to ROM lock bit read mode
	1	Lock bits are read using the register read method (lock bit read 2)	By going on to issue a 0xD0 command, the lock bit read 2 command is executed.
E2 DataFlash	0	Transition to E2 DataFlash lock bit read mode	The E2 DataFlash memory does not have lock bits. Accordingly, when the E2 DataFlash memory makes a transition to E2 DataFlash lock bit read mode, the values read from the E2 DataFlash area are undefined.
	1	Blank check command	By going on to issue a 0xD0 command, the blank check command is executed.

40.2.3 Flash Access Status Register (FASTAT)

Address(es): 007F C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	E2 DataFlash Programming/Erasure Protection Violation Flag	0: No programming/erasure protection violation 1: Programming/erasure protection violation	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation Flag	0: No read protection violation 1: Read protection violation	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAE	E2 DataFlash Access Violation Flag	0: No E2 DataFlash access violation 1: E2 DataFlash access violation	R/(W) *1
b4	CMDLK	FCU Command Lock Flag	0: FCU accepts the command 1: FCU does not accept the command	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation Flag	0: No ROM access violation 1: ROM access violation	R/(W) *1

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

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.
When any bit from among the DFLWPE, DFLRPE, DFLAE, and ROMAE bits in FASTAT is set to 1, the FSTAT0.ILGLERR bit is set to 1, which places the FCU in the command-locked state.

DFLWPE Bit (E2 DataFlash Programming/Erasure Protection Violation)

This bit is used to indicate whether or not the programming/erasure protection set by DFLWEy (y = 0, 1) is violated. When the DFLWPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, and the CMDLK bit become 1 (command-locked state).

[Setting condition]

- A programming/erasure command is issued for an E2 DataFlash area for which programming or erasure is disabled by DFLWEy (y = 0, 1).

[Clearing condition]

- When 0 is written after reading 1

DFLRPE Bit (E2 DataFlash Read Protection Violation Flag)

This bit is used to indicate whether or not the reading protection set by DFLREy (y = 0, 1) is violated.
When the DFLRPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting condition]

- An E2 DataFlash area for which reading is disabled by DFLREy (y = 0, 1) is read out.

[Clearing condition]

- When 0 is written after reading 1

DFLAE Bit (E2 DataFlash Access Violation Flag)

This bit indicates whether an E2 DataFlash access violation occurred.
When the DFLAE bit is set to 1, the ILGLERR bit in FSTATR0 is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An E2 DataFlash area is read in E2 DataFlash P/E normal mode and when the FENTRYD bit in FENTRYR is set to 1.
- An FCU command is issued for an E2 DataFlash area when the FENTRYD bit is set to 0.
- An FCU command is issued for an E2 DataFlash area or the area isread out when the FENTRYR.FENTRYn bit (n = 0 to 3) is set to 1.

[Clearing condition]

- When 0 is written after reading 1

CMDLK Bit (FCU Command Lock Flag)

This bit is used to indicate whether the FCU can receive commands.
When any bit of the FASTAT register is set to 1, the CMDLK bit is set to 1, and the FCU receives no commands (see section 40.6.2, Command-Locked State). To enable the FCU to receive commands, a status register clear command must be issued to the FCU after setting FASTAT to 10h.

[Setting condition]

- The FCU detects an error, and the CMDLK bit is set to 1 (command-locked state).

[Clearing condition]

- A status register clear command is issued while the FASTAT register is 10h.

ROMAE Bit (ROM Access Violation Flag)

This bit indicates whether a ROM access violation occurred.
When the ROMAE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- A ROM programming/erasure address is read out when the FCU is in ROM P/E normal mode*1

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 1	The FENTRY1 bit is 1	The FENTRY2 bit is 1	The FENTRY3 bit is 1
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

- An FCU command is issued for a ROM programming/erasure address or the address is read out*1

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 0	The FENTRY1 bit is 0	The FENTRY2 bit is 0	The FENTRY3 bit is 0
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

Note 1. The FENTRY2 bit is available when the capacity of the user area exceeds 1 Mbyte. The FENTRY3 bit is available when the capacity of the user area exceeds 1.5 Mbytes.

- A ROM-reading address is read out while FENTRYR has placed the ROM in ROM P/E mode

ROM Capacity	ROM-reading address ranges
1 Mbyte	FFF0 0000h to FFFF FFFFh
1.5 Mbytes	FFE8 0000h to FFFF FFFFh
2 Mbytes	FFE0 0000h to FFFF FFFFh

[Clearing condition]

- When 0 is written after reading 1

40.2.4 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
Value after reset:	1	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	E2 DataFlash Programming/Erasure Protection Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of protection against programming and erasure. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of protection against reading. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAEIE	E2 DataFlash Access Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of access to the E2 DataFlash. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt when the FASTAT.CMDLK bit is set to 1 (command-locked state). 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of access to the ROM. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

DFLWPEIE Bit (E2 DataFlash Programming/Erasure Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash programming/erasure protection violation occurs and the DFLWPE bit in FASTAT is set to 1.

DFLRPEIE Bit (E2 DataFlash Read Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash read protection violation occurs and the DFLRPE bit in FASTAT is set to 1.

DFLAEIE Bit (E2 DataFlash Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash access violation occurs and the DFLAE bit in FASTAT is set to 1.

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when the FASTAT.CMDLK bit is set to 1 (command-locked state).

ROMAEIE Bit (ROM Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a ROM access violation occurs and the FASTAT.ROMAE bit is set to 1.

40.2.5 Flash Ready Interrupt Enable Register (FRDYIE)

Address(es): 007F C412h

	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: FRDYI interrupt requests disabled 1: FRDYI interrupt requests enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is to enable or disable a flash ready interrupt request when programming/erasure is completed.
If the FRDYIE bit is set to 1, a flash ready interrupt request (FRDYI) is generated when execution of the FCU command has completed (FSTATR0.FRDY bit changes from 0 to 1).

40.2.6 E2 DataFlash Read Enable Register 0 (DFLRE0)

Address(es): 007F C440h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
Value after 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	DBRE00	0000-0063 Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE01	0064-0127 Block Read Enable		R/W
b2	DBRE02	0128-0191 Block Read Enable		R/W
b3	DBRE03	0192-0255 Block Read Enable		R/W
b4	DBRE04	0256-0319 Block Read Enable		R/W
b5	DBRE05	0320-0383 Block Read Enable		R/W
b6	DBRE06	0384-0447 Block Read Enable		R/W
b7	DBRE07	0448-0511 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLRE0 register. To modify the DFLRE0 register, write 2Dh 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. Write data is not retained.

DFLRE0 is a register to enable or disable the 0000 to 0511 blocks of the data area (see Table 40.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 blocks).
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

DBREj Bit (DBj Block Read Enable) (j = 00 to 07)

This bit is used to enable or disable the DB0000 to DB0511 blocks of the data area to be read.

40.2.7 E2 DataFlash Read Enable Register 1 (DFLRE1)

Address(es): 007F C442h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
Value after 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	DBRE08	0512-0575 Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE09	0576-0639 Block Read Enable		R/W
b2	DBRE10	0640-0703 Block Read Enable		R/W
b3	DBRE11	0704-0767 Block Read Enable		R/W
b4	DBRE12	0768-0831 Block Read Enable		R/W
b5	DBRE13	0832-0895 Block Read Enable		R/W
b6	DBRE14	0896-0959 Block Read Enable		R/W
b7	DBRE15	0960-1023 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLRE1 register. To modify the DFLRE1 register, write D2h 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. Write data is not retained.

DFLRE1 is a register to enable or disable the 0512 to 1023 blocks of the data area (see Table 40.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 blocks).
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

DBREj Bit (Block Read Enable) (j = 08 to 15)

This bit is used to enable or disable the 0512 to 1023 blocks of the data area to be read.

40.2.8 E2 DataFlash P/E Enable Register 0 (DFLWE0)

Address(es): 007F C450h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
Value after 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	DBWE00	0000-0063 Block Programming/ Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE01	0064-0127 Block Programming/ Erasure Enable		R/W
b2	DBWE02	0128-0191 Block Programming/ Erasure Enable		R/W
b3	DBWE03	0192-0255 Block Programming/ Erasure Enable		R/W
b4	DBWE04	0256-0319 Block Programming/ Erasure Enable		R/W
b5	DBWE05	0320-0383 Block Programming/ Erasure Enable		R/W
b6	DBWE06	0384-0447 Block Programming/ Erasure Enable		R/W
b7	DBWE07	0448-0511 Block Programming/ Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLWE0 register. To modify the DFLWE0 register, write 1Eh 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. Write data is not retained.

DFLWE0 is a register to enable or disable the 0000 to 0511 blocks of the data area (see Table 40.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 blocks).
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

DBWEj Bit (Block Programming/Erasure Enable) (j = 00 to 07)

This bit is used to enable or disable the 0000 to 0511 blocks of the data area to be programmed or erased.

40.2.9 E2 DataFlash P/E Enable Register 1 (DFLWE1)

Address(es): 007F C452h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
Value after 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	DBWE08	0512-0575 Block Programming/ Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE09	0576-0639 Block Programming/ Erasure Enable		R/W
b2	DBWE10	0640-0703 Block Programming/ Erasure Enable		R/W
b3	DBWE11	0704-0767 Block Programming/ Erasure Enable		R/W
b4	DBWE12	0768-0831 Block Programming/ Erasure Enable		R/W
b5	DBWE13	0832-0895 Block Programming/ Erasure Enable		R/W
b6	DBWE14	0896-0959 Block Programming/ Erasure Enable		R/W
b7	DBWE15	0960-1023 Block Programming/ Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLWE1 register. To modify the DFLWE1 register, write E1h 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. Write data is not retained.

DFLWE1 is a register to enable or disable the 0512 to 1023 blocks of the data area (see Table 40.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 blocks).
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

DBWEj Bit (Block Programming/Erasure Enable) (j = 08 to 15)

This bit is used to enable or disable the 0512 to 1023 blocks of the data area to be programmed or erased.

40.2.10 Flash Status Register 0 (FSTATR0)

Address(es): 007F FFB0h

	b7	b6	b5	b4	b3	b2	b1	b0
	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRGSPD	Programming Suspend Status Flag	0: Other than the status described below 1: During programming suspend processing or programming suspended	R
b1	ERSSPD	Erasure Suspend Status Flag	0: Other than the status described below 1: When erasure suspend processing or erasure suspended	R
b2	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b3	SUSRDY	Suspend Ready Flag	0: P/E suspend commands cannot be received 1: P/E suspend commands can be received	R
b4	PRGERR	Programming Error Flag	0: Programming terminates normally 1: An error occurs during programming	R
b5	ERSERR	Erasure Error	0: Erasure terminates normally 1: An error occurs during erasure	R
b6	ILGLERR	Illegal Command Error Flag	0: FCU detects no illegal command or illegal ROM/E2 DataFlash access 1: FCU detects an illegal command or illegal ROM/E2 DataFlash access	R
b7	FRDY	Flash Ready Flag	0: During programming/erasure, During suspending programming/erasure, During the lock bit read 2 command processing, During the peripheral clock notification command processing, During the blank check processing of E2 DataFlash. 1: Processing described above is not performed	R

FSTATR0 is also reset by setting the FRESETR.FRESET bit to 1.
When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

PRGSPD Bit (Programming Suspend Status Flag)

This bit is used to indicate that the FCU enters the programming suspend processing state or programming suspended state. For details, see section 40.5, Suspending Operation.

[Setting condition]

- The FCU has initiated a write suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

ERSSPD Bit (Erasure Suspend Status Flag)

This bit is used to indicate that the FCU enters the erasure suspend processing state or erasure suspended state. For details, see section 40.5, Suspending Operation.

[Setting condition]

- The FCU has initiated an erasure suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

SUSRDY Bit (Suspend Ready Flag)

This bit is used to indicate whether the FCU can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure process, the FCU enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- The FCU has accepted a P/E suspend command.
- During programming/erasure process, the FASTAT.CMDLK bit becomes 1 (command-locked state).

PRGERR Bit (Programming Error Flag)

This bit is used to indicate the result of the ROM/E2 DataFlash programming process by the FCU.
When the PRGERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An error occurs during programming.
- A programming command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

ERSERR Bit (Erasure Error Flag)

This bit is used to indicate the result of the ROM/E2 DataFlash erasure process by the FCU.
When the ERSERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An error occurs during erasure.
- A block erase command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

ILGLERR Bit (Illegal Command Error Flag)

This bit is used to indicate that the FCU detects any illegal command or ROM/E2 DataFlash access.
When the ILGLERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- The FCU detects an illegal command.
- The FCU detects an illegal ROM/E2 DataFlash access
  (one of the ROMAE, DFLAE, DFLRPE, and DFLWPE bits in FASTAT is 1).
- The setting of FENTRYR is invalid.

[Clearing condition]

- After a status register clear command has been issued under conditions where FASTAT is set to 10h

40.2.11 Flash Status Register 1 (FSTATR1)

Address(es): 007F FFB1h

	b7	b6	b5	b4	b3	b2	b1	b0
	FCUERR	—	—	FLOCKST	—	—	—	—
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	FLOCKST	Lock Bit Status	0: Protected 1: Not protected	R
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	FCUERR	FCU Error Flag	0: No error occurs in the FCU processing 1: An error occurs in the FCU processing	R

FSTATR1 is also reset by setting the FRESETR.FRESET bit to 1. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

FLOCKST Bit (Lock Bit Status)

This bit is to reflect the read data of a lock bit when using the lock bit read 2 command.
When the FSTATR0.FRDY bit is set to 1 after a lock bit read 2 command is issued, the value of the lock bit status is stored in the FLOCKST bit. The value of the FLOCKST bit is retained until the completion of the next lock bit read 2 command.

FCUERR Bit (FCU Error Flag)

This bit is used to indicate that an error occurs in the FCU internal processing.
When the FCUERR bit is set to 1, set the FRESETR.FRESET bit to 1 to initialize the FCU.

40.2.12 Flash P/E Mode Entry Register (FENTRYR)

Address(es): 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	FENTRY3	FENTRY2	FENTRY1	FENTRY0
Value after 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	FENTRY0	ROM P/E Mode Entry 0	0: Area 0 is in ROM read mode. 1: Area 0 is in ROM P/E mode.	R/W
b1	FENTRY1	ROM P/E Mode Entry 1*1	0: Area 1 is in ROM read mode. 1: Area 1 is in ROM P/E mode.	R/W
b2	FENTRY2	ROM P/E Mode Entry 2*1	0: Area 2 is in ROM read mode. 1: Area 2 is in ROM P/E mode.	R/W
b3	FENTRY3	ROM P/E Mode Entry 3*1	0: Area 3 is in ROM read mode. 1: Area 3 is in ROM P/E mode.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode. 1: E2 DataFlash is in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FENTRYR register. To modify the FENTRYR register, write AAh 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. Write data is not retained.

To place the ROM/E2 DataFlash in ROM P/E mode so that the FCU can accept commands, the FENTRYD and FENTRYn (n = 0 to 3) bits must be set to 1. Note that if a value is set other than AA01h, AA02h, AA04h, AA08h, and AA80h in FENTRYR, the FSTATR0.ILGLERR bit is set to 1 and the FSATAT.CMDLK bit is set to 1 (command-locked state).

The ROM exists from area 0 up to area 3 at the maximum, and the FENTRY0 to FENTRY3 bits correspond to the respective areas. The FENTRYn (n = 0 to 3) bits for areas that are not present cannot be set to 1. See Figure 40.2, Memory Area Configuration of ROM, for the relation between the ROM capacity and areas.

FENTRYR is also reset when the FRESETR.FRESET bit is set to 1. When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

FENTRYn Bit (ROM P/E Mode Entry FENTRYn (n = 0 to 3))

This bit is used to place area n (n = 0 to 3) in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYn (n = 0 to 3) bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYn (n = 0 to 3) bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

FENTRYD Bit (E2 DataFlash P/E Mode Entry)

The FENTRYD bit is used to place the E2 DataFlash in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FRDY bit in FSTATR0 is set to 1
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- When the writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYD bit

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYD bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

40.2.13 Flash Protection Register (FPROTR)

Address(es): 007F FFB4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FPKEY[7:0]								—	—	—	—	—	—	—	FPROTCN
Value after 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	FPROTCN	Lock Bit Protection Cancel	0: Protection with a lock bit enabled 1: Protection with a lock bit disabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FPKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FPROTR register. To modify the FPROTR register, write 55h 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. Write data is not retained.

FPROTR is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from FPROTR is 0000h and writing is disabled. This register is dedicated to the ROM.

FPROTCN Bit (Lock Bit Protection Cancel)

This bit is used to enable/disable the programming/erasure protection with a lock bit.

[Setting condition]

- 55h is written to the FPKEY[7:0] bits and 1 is written to the FPROTCN bit in word access when the value of FENTRYR is other than 0000h.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FPKEY[7:0] bits are other than 55h.
- 55h is written to the FPKEY[7:0] bits and 0 is written to the FPROTCN bit in word access.
- The value of FENTRYR is 0000h.

40.2.14 Flash Reset Register (FRESETR)

Address(es): 007F FFB6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRKEY[7:0]								—	—	—	—	—	—	—	FRESET
Value after 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	FRESET	Flash Reset	0: FCU is not reset 1: FCU is reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FRKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FRESETR register. To modify the FRESETR register, write CCh 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. Write data is not retained.

When on-chip ROM is disabled, the data read from FRESETR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

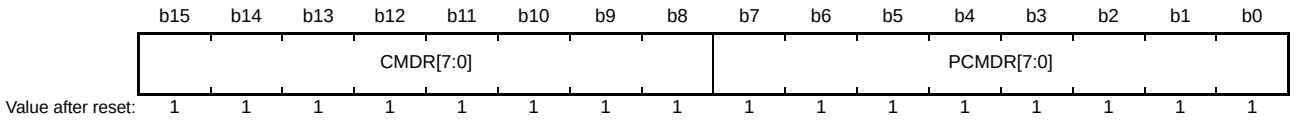
FRESET Bit (Flash Reset)

When the FRESET bit is set to 1, programming/erasure operations for the ROM/E2 DataFlash are forcibly terminated, and the FCU is initialized.

High voltage is applied to the memory of the ROM/E2 DataFlash during programming/erasure. To ensure the time required for dropping the voltage applied to the memory, keep the FRESET bit set to 1 for tFCUR (FCU reset time, see section 41, Electrical Characteristics) when initializing the FCU. While the FRESET bit is kept as 1, prohibit the ROM/E2 DataFlash from being read. Additionally, when the FRESET bit is set to 1, the FCU commands cannot be used because FENTRYR is initialized.

40.2.15 FCU Command Register (FCMDR)

Address(es): 007F FFBAh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand	Store the command immediately before the last command received by the FCU.	R
b15 to b8	CMDR[7:0]	Command	Store the last command received by the FCU.	R

FCMDR is also initialized when the FRESETR.FRESET bit is set to 1.

Table 40.5 lists the states of FCMDR after receiving each command.

When on-chip ROM is disabled, data read from FCMDR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

Table 40.5 States of FCMDR after Receiving Each Command

Command	CMDR[7:0]	PCMDR[7:0]
Normal mode transition	FFh	Previous command
Status read mode transition	70h	Previous command
Lock bit read mode transition (lock bit read 1)	71h	Previous command
Peripheral clock notification	E9h	Previous command
Programming	E8h	Previous command
Block erase	D0h	20h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status register clear	50h	Previous command
Lock bit read 2/blank check	D0h	71h
Lock bit programming	D0h	77h

40.2.16 FCU Processing Switching Register (FCPSR)

Address(es): 007F FFC8h

	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	Erasure Suspend Mode	0: Suspension priority mode 1: Erasure priority mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FCPSR is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from FCPSR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

ESUSPMD Bit (Erasure Suspend Mode)

This bit is to select the erasure suspend mode for when a P/E suspend command is issued while the FCU executes the erasure processing for the ROM/E2 DataFlash. For details, see section 40.5, Suspending Operation.

40.2.17 E2 DataFlash Blank Check Control Register (DFLBCCNT)

Address(es): 007F FFCAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BCSIZE	—	—	—	—	BCADR[10:0]										
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
b10 to b0	BCADR[10:0]	Blank Check Address Setting	Set the address of the area to be checked	R/W
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BCSIZE	Blank Check Size Setting	0: The size of the area to be blank checked is 2 bytes. 1: The size of the area to be blank checked is 2 Kbytes.	R/W

DFLBCCNT is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from DFLBCCNT is 0000h and writing is disabled. This register is dedicated to the E2 DataFlash.

BCADR[10:0] Bits (Blank Check Address Setting)

These bits are used to set the address of the area to be checked when the size of the area to be checked by a blank check command is 2 bytes (the BCSIZE bit is 0). Set the BCADR[0] bit to 0.
When the BCSIZE bit is 0, the start address of the area to be checked is obtained by adding the DFLBCCNT setting value to the block start address (in 2-Kbyte units) specified at issuance of a blank check command.
When the BCSIZE bit is 1, the setting of the BCADR[10:0] bits will be ignored.

40.2.18 Flash P/E Status Register (FPESTAT)

Address(es): 007F FFCCh

	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	00h: No error 01h: Programming error against areas protected by a lock bit 02h: Programming error due to sources other than the lock bit protection 11h: Erasure error against areas protected by a lock bit 12h: Erasure error due to sources other than the lock bit protection (Values other than above are reserved)	R
b15 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

FPESTAT is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from FPESTAT is 0000h and writing is disabled. This register is dedicated to the ROM.

PEERRST[7:0] Bits (P/E Error Status)

These bits are used to indicate the reason of an error that occurs during the programming/erasure processing for the ROM. The value of the PEERRST[7:0] bits is valid only when the FSTATR0.FRDY bit is set to 1 while the FSTATR0.ERSERR bit or FSTATR0.PRGERR bit is 1. The value of the reason of the past error is retained in the PEERRST[7:0] bits when the ERSERR bit and PRGERR bit is 0.

40.2.19 E2 DataFlash Blank Check Status Register (DFLBCSTAT)

Address(es): 007F FFCEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCST
Value after 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	BCST	Blank Check Status	0: The area to be blank-checked is erased (blank) 1: 0 or 1 is written in the area to be blank-checked	R
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DFLBCSTAT is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from DFLBCSTAT is 0000h and writing is disabled. This register is dedicated to the E2 DataFlash.

40.2.20 Peripheral Clock Notification Register (PCKAR)

Address(es): 007F FFE8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PCKA[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	PCKA[7:0]	Peripheral Clock Notification	These bits are used to set the FlashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This setting is used to control the programming/erasure time.
PCKAR is also reset when the FRESETR.FRESET bit is set to 1.
When on-chip ROM is disabled, the data read from PCKAR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

PCKA[7:0] Bits (Peripheral Clock Notification)

These bits are used to set the FlashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.
Set the FCLK frequency in the PCKA[7:0] bits and issue a peripheral clock notification command before programming/erasure. Do not change the frequency during programming/erasure for the ROM/E2 DataFlash.
Write a setting to the PCKA[7:0] bits as a binary value that selects the operating frequency in MHz units.
For example, when the operating frequency of the FlashIF clock is 31.9 MHz, the setting value is calculated as follows:

- Round 31.9 off to a whole number.
- Convert 32 to binary and set the upper bits and lower bits of the PCKA[7:0] bits to 20h (0010 0000b).

Note 1. When the PCKA[7:0] bits are set to values outside the range from 4 to 32 MHz, do not issue a programming command to the ROM/E2 DataFlash.
Note 2. When the PCKA[7:0] bits are set to a frequency that is different from the FCLK, the data of the ROM/E2 DataFlash may be damaged.
Note 3. Note that the programming time depends on the frequency to some extent even if the PCKA[7:0] bits are used.

40.3 Operating Modes Associated with Flash Memory

For information on the relationship between the setting of the level on the MD pin and the operating mode for this MCU, refer to section 3, Operating Modes.
The ROM and E2 DataFlash can be read, programmed, and erased on board in boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), or on-chip ROM enabled extended mode.
The area where programming and erasure are permitted, the area from which booting up proceeds, and areas erased at the time of booting up differ with the mode. The differences between modes are indicated in Table 40.6.

Table 40.6 Differences between Modes

Item	Boot Mode	User Boot Mode	Single-Chip Mode (with On-Chip ROM Enabled) or On-chip ROM Enabled Extended Mode
Environment for programming and erasure	On-board programming		
Programmable and erasable area	User area/user boot area/data area	User area/data area	User area/data area
Division into erasure blocks	Possible*1	Possible	Possible
Boot program at a reset	Boot program	User boot program	User program

Note 1. The entire ROM may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 40.8.2, State Transitions in Boot Mode and section 40.8.4, ID Code Protection (Boot Mode).

- Programming and erasure of the user boot area are only possible in boot mode.
- In boot mode, a host is able to program, erase, or read the user area, user boot area, or data area via an SCI.
- In boot mode, on-chip RAM is employed for the boot program. Therefore, the data on the on-chip RAM is not retained.
- Booting-up in user boot mode is from the user boot area. Rewriting of the user boot area in boot mode can enable reading from or writing to the user area and data area via any desired interface.

40.3.1 Erasure of Areas that are Subject to ID Code Protection

The areas to be erased depend on the operating mode at the time of booting up and the result of reference to the ID code. For a description of the ID code protection, refer to section 40.8.4, ID Code Protection (Boot Mode).

Table 40.7 Connection between areas for erasure, the operating mode, and ID code protection

Operating Mode at the Time of Booting Up	Control Code of ID Code Protection	Matching or non-matching ID codes	User Area	User Boot Area	Data Area
Boot mode	45h	Matching	—	—	—
		Non-matching three times consecutively	Erase	Erase	Erase
	52h	Matching or non-matching	—	—	—
	Other than 45h and 52h (ID code protection is disabled)	—	Erase	Erase	Erase
User boot mode	ID code protection depends on the user boot mode specification.				
Single-chip mode	Erasure does not proceed when booting up is in single-chip mode.				

40.4 FCU

The ROM and E2 DataFlash operations are performed by issuing commands to a dedicated sequencer (FCU).
The mode transitions of the FCU and the system of commands are described below. The descriptions apply in common to boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), and on-chip ROM enabled extended mode.

40.4.1 FCU Modes

The FCU has five modes. Transitions between modes are caused by modifying FENTRYR or issuing FCU commands. Since the E2 DataFlash P/E mode is included in ROM read mode, high-speed reading from the ROM is possible in E2 DataFlash P/E mode. Figure 40.4 is a diagram of the FCU mode transitions.

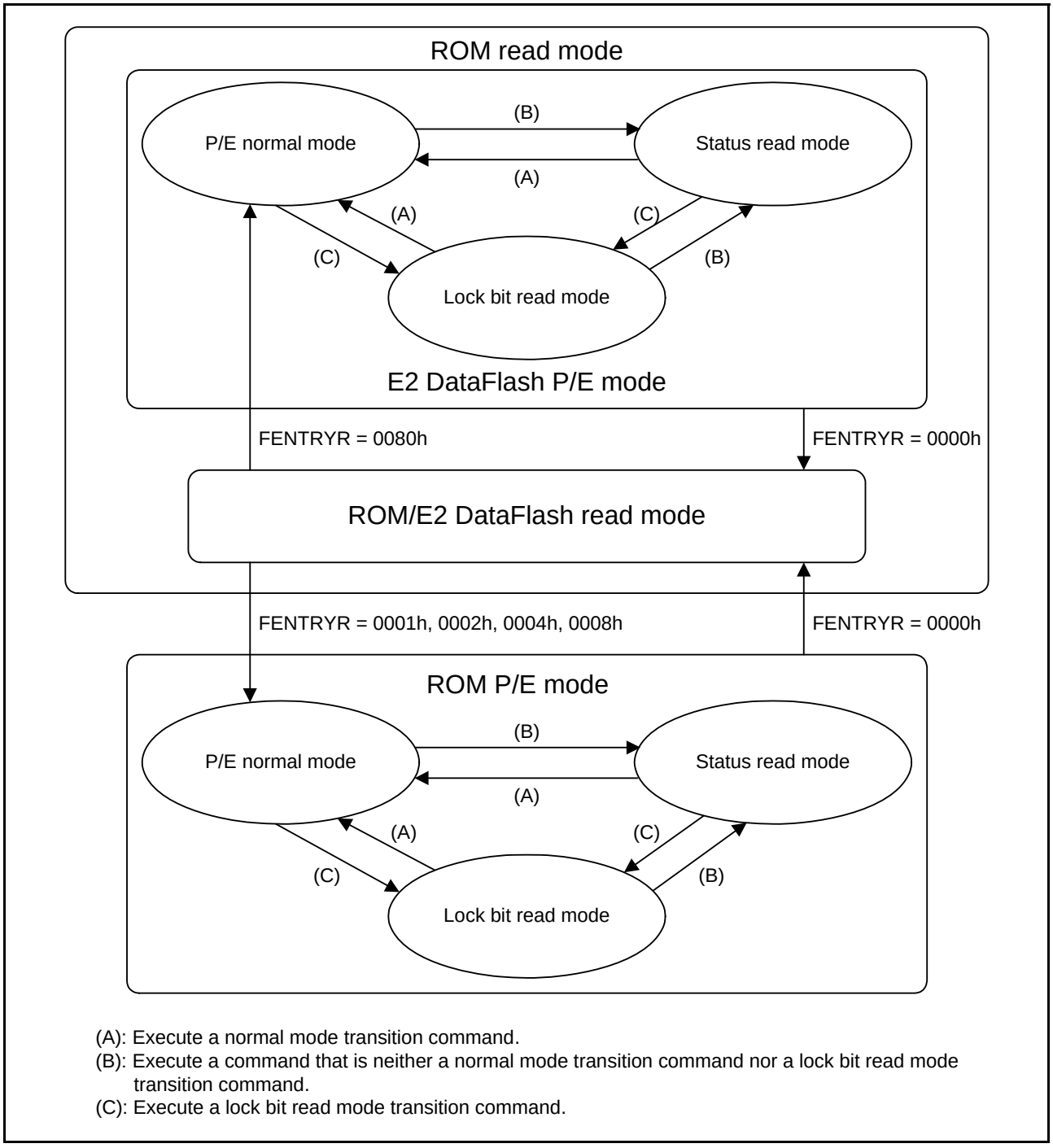


Figure 40.4 Mode Transitions of the FCU

40.4.1.1 ROM Read Modes

The ROM read modes are for high-speed reading of the ROM. Reading out from an address for reading can be accomplished in one cycle of ICLK.
ROM/E2 DataFlash read mode and E2 DataFlash P/E mode are the two kinds of ROM read modes.

40.4.1.2 ROM/E2 DataFlash Read Mode

This mode is for reading the ROM and E2 DataFlash memory. The FCU does not accept FCU commands. The FCU enters this mode when the FENTRYR.FENTRYn (n = 0 to 3) bits are set to 0 with the FENTRYR.FENTRYD bit set to 0.

40.4.1.3 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM. High-speed reading of the ROM is not possible in these modes. Reading out from an address within the range for reading causes a ROM-access violation, and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).
There are three ROM P/E modes.

(1) ROM P/E Normal Mode

The transition to ROM P/E normal mode is the first transition in the process of programming or erasing the ROM. The FCU enters this mode when the FENTRYR.FENTRYD bit is set to 0, with any of the FENTRYR.FENTRYn (n = 0 to 3) bits read mode, or when the normal mode transition command is received in ROM P/E modes. Table 40.11 lists the acceptable commands in this mode.
Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRYn (n = 0 to 3) bits are set to 1 causes a ROM-access violation, and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

(2) ROM Status Read Mode

In the ROM status read mode, the state of the ROM can be read. The FCU enters this mode when a status read mode transition command is received, or when a command other than the normal mode transition or lock bit read mode transition command is received in ROM P/E modes.
ROM status read mode encompasses the states where the FSTATR0.FRDY bit is 0 and the FASTAT.CMDLK bit is set to 1 (command-locked state) after an error has occurred. Table 40.11 lists the acceptable commands in this mode.
Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRYn (n = 0 to 3) bits are 1 allows the value of FSTATR0 to be read.

(3) ROM Lock-Bit Read Mode

In the ROM lock-bit read mode, reading the ROM allows the lock bits to be read. The FCU enters this mode when a lock-bit read mode transition command is received in ROM P/E modes. Table 40.11 lists the acceptable commands in this mode.
Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRYn (n = 0 to 3) bits are 1 allows the value of the lock bit of the block including the accessed address to be read from all the read bits.

40.4.1.4 E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. Although high-speed reading from the ROM is possible, reading from the E2 DataFlash is not executable. Although FCU commands for the E2 flash memory are accepted in this mode, FCU commands for the ROM are not. The FCU enters this mode when the FENTRYR.FENTRYn (n = 0 to 3) bits are set to 0 and the FENTRYR.FENTRYD bit is set to 1.
There are three E2 DataFlash P/E modes.

(1) E2 DataFlash P/E Normal Mode

The transition to E2 DataFlash P/E normal mode is the first transition in the process of programming or erasing the E2 DataFlash.
The FCU enters this mode when the FENTRYR.FENTRYD bit is set to 1 and the FENTRYR.FENTRYn (n = 0 to 3) bits are set to 0 in ROM/E2 DataFlash read mode, or when the normal mode transition command is received in E2 DataFlash P/E modes. Table 40.11 lists the acceptable commands in this mode.

(2) E2 DataFlash Status Read Mode

The E2 DataFlash status read mode is for reading information on the state of the E2 DataFlash.
The FCU enters this mode when a status read mode transition command is received, or when a command other than the normal mode transition and lock bit read mode transition command is received in E2 DataFlash P/E modes. E2 DataFlash status read mode encompasses the states where the FRDY bit in FSTATR0 is 0 and the FASTAT.CMDLK bit is set to 1 (command-locked state) after an error has occurred.Table 40.11 lists the acceptable commands in this mode.
Reading out from an address within the E2 DataFlash area will actually read the value of the FSTATR0 register. High-speed reading of the ROM is possible.

(3) E2 DataFlash Lock-Bit Read Mode

Since the E2 DataFlash memory does not have lock bits, the lock bits are not read even if a transition to this mode is made. If the E2 DataFlash memory area is read after a transition to this mode, an E2 DataFlash access violation is not generated, but the values read are undefined. High-speed reading of the ROM is possible.
The FCU enters this mode when a lock-bit read mode transition command is received in E2 DataFlash P/E modes. Table 40.11 lists the acceptable commands in this mode.

40.4.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and commands for programming and erasure. Table 40.8 lists the FCU commands for use with the ROM and E2 DataFlash.

Table 40.8 FCU Commands

Command	ROM	E2 DataFlash
P/E normal mode transition	Shifts to normal mode (see section 40.4.3, Connections between FCU Modes and Commands)	
Status read mode transition	Shifts to status read mode (see section 40.4.3, Connections between FCU Modes and Commands)	
Lock bit read mode transition (lock bit read 1)	Shifts to lock bit read mode (see section 40.4.3, Connections between FCU Modes and Commands)	
Peripheral clock notification	Sets the FlashIF clock (FCLK)	
Programming	ROM programming (in 128-byte units)	E2 DataFlash programming (in 2-byte units)
Block erase	ROM erasure (in block units, with the lock bit being erased simultaneously)	E2 DataFlash erasure (in block units)
P/E suspend	Suspends programming/erasure	
P/E resume	Resumes programming/erasure	
Status register clear	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and the FASTAT.CMDLK bit (FCU command lock bit)	
Lock bit read 2	Reads the lock bit of a specified block (the value of the lock bit is reflected in the FSTATR1.FLOCKST bit)	—
Lock bit programming	Programs the lock bit of a specified block	—
Blank checking	—	Checks whether the E2 DataFlash memory is blank

The lock bit read 2 command is for the ROM also used as the blank check command for the E2 DataFlash memory. That is, when a lock bit read 2 command is issued for the E2 DataFlash, blank checking is executed for the E2 DataFlash memory.

Commands for the FCU are issued by writing an FCU command to addresses within the range for ROM programming and erasure or an address in the E2 DataFlash. Table 40.9 lists the formats of the FCU commands. Writing data to addresses listed in Table 40.9 in accordance with certain conditions causes the FCU to execute processing for the corresponding command. An explanation in summary of the FCU command format is given in Table 40.10. Furthermore, although almost all of the FCU commands are in byte units, some of the commands have to be issued in word units.

For details on the conditions for the acceptance of the individual FCU commands, see section 40.4.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 40.4.4, FCU Command Usage.

Table 40.9 FCU Command Formats

Command	Number of Bus Cycles	Address / Data	1st Cycle	2nd Cycle	3rd Cycle	4th Cycle	5th Cycle	6th Cycle	7th to 66th Cycles	67st Cycle
P/E normal mode transition	1	Address	RA	—	—	—	—	—	—	—
		Data	FFh	—	—	—	—	—	—	—
Status read mode transition	1	Address	RA	—	—	—	—	—	—	—
		Data	70h	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	Address	RA	—	—	—	—	—	—	—
		Data	71h	—	—	—	—	—	—	—
Peripheral clock notification	6	Address	RA	RA	RA	RA	RA	RA	—	—
		Data	E9h	03h	0F0Fh *1	0F0Fh *1	0F0Fh *1	D0h	—	—
Programming (ROM)	67	Address	RA	RA	WA	RA	RA	RA	RA	RA
		Data	E8h	40h	WDn *1	WDn *1	WDn *1	WDn *1	WDn *1	D0h
Programming (E2 DataFlash)	4	Address	RA	RA	WA	RA	—	—	—	—
		Data	E8h	01h	WDn *1	D0h	—	—	—	—
Block erase	2	Address	RA	BA	—	—	—	—	—	—
		Data	20h	D0h	—	—	—	—	—	—
P/E suspend	1	Address	RA	—	—	—	—	—	—	—
		Data	B0h	—	—	—	—	—	—	—
P/E resume	1	Address	RA	—	—	—	—	—	—	—
		Data	D0h	—	—	—	—	—	—	—
Status register clear	1	Address	RA	—	—	—	—	—	—	—
		Data	50h	—	—	—	—	—	—	—
Lock bit read 2 (ROM)	2	Address	RA	BA	—	—	—	—	—	—
		Data	71h	D0h	—	—	—	—	—	—
Blank checking (E2 DataFlash)	2	Address	RA	BA	—	—	—	—	—	—
		Data	71h	D0h	—	—	—	—	—	—
Lock bit programming (ROM)	2	Address	RA	BA	—	—	—	—	—	—
		Data	77h	D0h	—	—	—	—	—	—

Note 1. Write data in word units.

Table 40.10 FCU Commands

Item		ROM	E2 DataFlash
Address	RA	Any address for programming or erasure within the target ROM area*1	Any address within the E2 DataFlash
	WA	Programming-destination address in the range for programming or erasure (128-byte alignment)	Programming-destination address (2-byte alignment)
	BA	Any address for programming or erasure within the target erasure block	Any address within the target erasure block
Data	WDn	nth word of data for programming (n = 1 to 64)	nth word of data for programming (n = 1)
	Others	Command issued to the target address	

Note 1. Addresses that can be used for programming or erasure differ according to the ROM capacity and the settings of the FENTRYR register. For the ROM capacity, see section 40.1.1, Configuration of the ROM Area, and for FENTRYR, see section 40.2.12, Flash P/E Mode Entry Register (FENTRYR).

40.4.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode also depends on the state of the FCU.

Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.

Commands that are acceptable in the various FCU modes and states are listed in Table 40.11. Issuing a command that is not currently acceptable leads to the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

Issuing of an FCU command must follow checking of the values of the FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FSTATR1.FCUERR bit after transitions of the FCU mode. Furthermore, the FASTAT.CMDLK bit can be checked to see if an error has occurred. The value of the FASTAT.CMDLK bit is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit.

Table 40.11 Acceptable Commands and the State and Mode (ROM P/E Mode and E2 DataFlash P/E Mode) of the FCU

	P/E Normal Mode			Status Read Mode										Lock-Bit Read Mode		
	Programming Suspended	Erasure Suspended	Other State	Programming or Erasure	Programming while erasure is suspended	Processing to Suspend Programming or Erasure	Lock Bit Read 2 Processing (ROM)	Blank checking (E2 DataFlash Memory)	Programming Suspended	Erasure Suspended	Command-Locked State (FRDY = 0)	Command-Locked State (FRDY = 1)	Other State	Programming Suspended	Erasure Suspended	Other State
FSTATR0.FRDY bit	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**1**
FSTATR0.SUSRDY bit	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
FSTATR0.ERSSPD bit	**0**	**1**	**0**	**0**	**1**	**0/1**	**0/1**	**0/1**	**0**	**1**	**0/1**	**0/1**	**0**	**0**	**1**	**0**
FSTATR0.PRGSPD bit	**1**	**0**	**0**	**0**	**0**	**0/1**	**0/1**	**0/1**	**1**	**0**	**0/1**	**0/1**	**0**	**1**	**0**	**0**
FASTAT.CMDLK bit	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
Normal mode transition	A	A	A	×	×	×	×	×	A	A	×	×	A	A	A	A
Status read transition	A	A	A	×	×	×	×	×	A	A	×	×	A	A	A	A
Lock-bit read transition (lock bit read 1)	A	A	A	×	×	×	×	×	A	A	×	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	×	×	×	A	×	×	A
Programming	×	*	A	×	×	×	×	×	×	*	×	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clear	A	A	A	×	×	×	×	×	A	A	A	A	A	A	A	A
Lock bit read 2 (ROM)	A	A	A	×	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit programming (ROM)	×	*	A	×	×	×	×	×	×	*	×	×	A	×	*	A
Blank checking (E2 DataFlash)	A	A	A	×	×	×	×	×	A	A	×	×	A	A	A	A

A: Acceptable

*: Only programming is acceptable for blocks other than the block where erasure was suspended

×: Not acceptable

40.4.4 FCU Command Usage

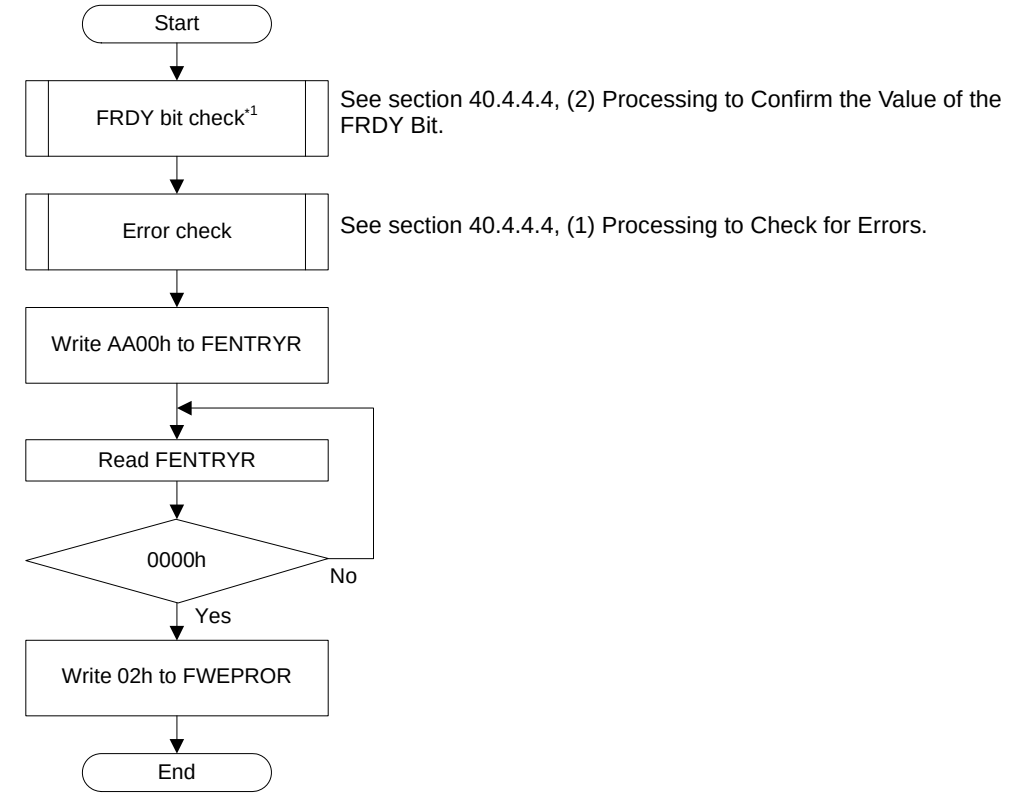
The set of FCU commands consists of commands for FCU mode transitions, actually programming or erasing the ROM, error processing, and suspension and resumption. The following passages describe the various commands. For a description of the modes and states where the respective commands are acceptable, see section 40.4.3, Connections between FCU Modes and Commands.

40.4.4.1 Mode Transitions

This subsection covers the commands for mode transitions. For an illustration of the various transitions between modes, see Figure 40.4.

(1) Switching to ROM Read Mode or ROM/E2 DataFlash Read Mode

High-speed reading of the ROM requires clearing of the FENTRYR.FENTRYn (n = 0 to 3) bits to 0, which places the FCU in ROM read mode. Writing of 02h as a byte to FWEPROR is also required to disable programming and erasure (see section 40.2.1, Flash Write Erase Protection Register (FWEPROR)).
Before switching the FCU from ROM P/E mode to read mode, ensure that all processing of FCU commands has been completed and that the FCU has not detected an error.
For a transition to ROM/E2 DataFlash read mode, the FENTRYR.FENTRYn (n = 0 to 3) bits and FENTRYD bits must be set to 0.



Note 1. tE16K: Erasure time for a 16-Kbyte block (Refer to the Electrical Characteristics chapter.)

Figure 40.5 Procedure for Transition to ROM Read Mode or ROM/E2 DataFlash Read Mode

(2) Switching to P/E Mode

A transition to ROM P/E mode is required before executing an FCU command for programming or erasure of the ROM. Setting any of the FENTRYR.FENTRYn (n = 0 to 3) bits to 1 causes a transition to ROM P/E mode for programming and erasure of the corresponding address range.

A transition to E2 DataFlash P/E mode is required before executing an FCU command for programming or erasure of the E2 DataFlash. For a transition to E2 DataFlash P/E mode, set the FENTRYR.FENTRYD bit to 1.

Before actually proceeding to program or erase the ROM, enable programming and erasure by writing 01h as a byte to FWEPROR (see section 40.2.1, Flash Write Erase Protection Register (FWEPROR)).

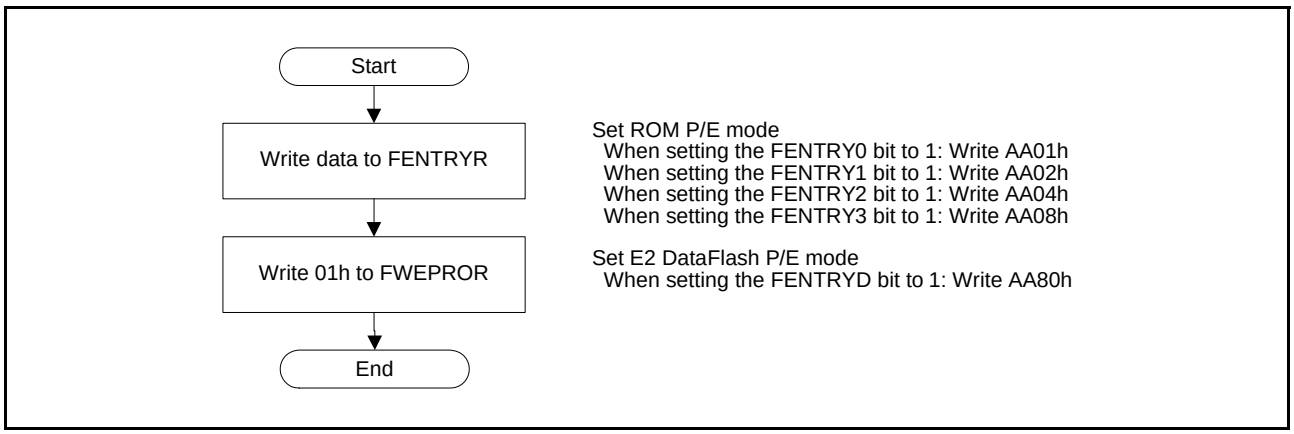


Figure 40.6 Procedure for Transition to ROM P/E Mode or E2 DataFlash P/E Mode

(3) Switching to P/E Normal Mode

Two methods are available for the transition to P/E normal mode: setting FENTRYR while the FCU is in ROM/E2 DataFlash read mode (see section 40.4.1, FCU Modes), or issuing the normal mode transition command while the FCU is in P/E mode (see Figure 40.7). The normal mode transition command is issued by writing FFh to a ROM programming/erasure address or to an E2 DataFlash address.

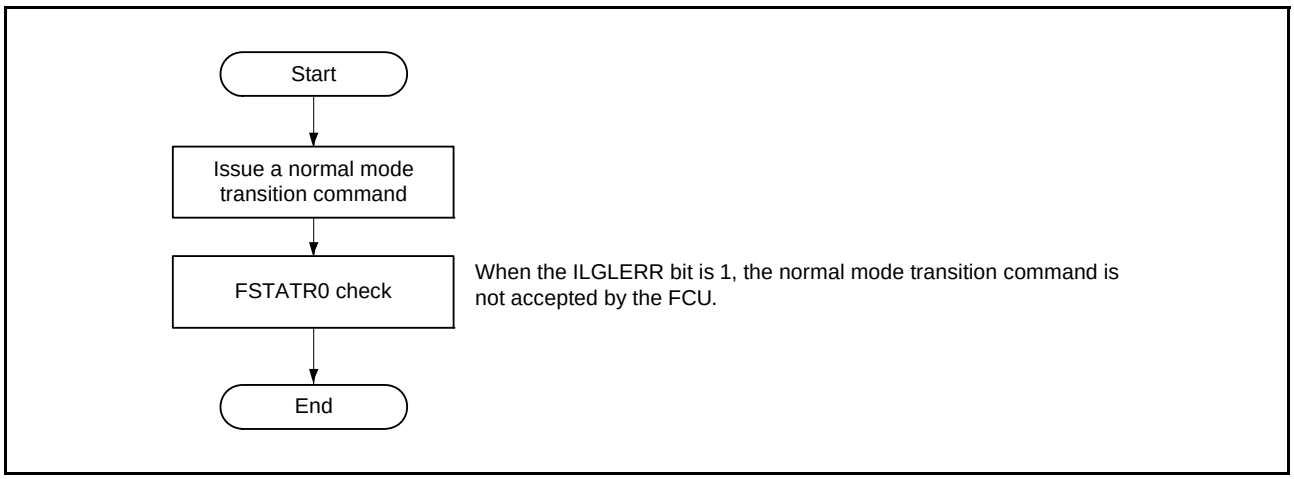


Figure 40.7 Procedure for Transition to ROM P/E Normal Mode

(4) Switching to Status Read Mode

Issuing a status read mode transition command or an FCU command other than a normal mode transition or lock bit read mode transition command places the FCU in status read mode. Figure 40.8 shows the procedure for checking FSTATR0 as an example. In the example, the status read mode transition command is issued to place the FCU in ROM status read mode, and the value of FSTATR0 is obtained by reading out from a ROM programming/erasure address or an E2 DataFlash address and then checked.

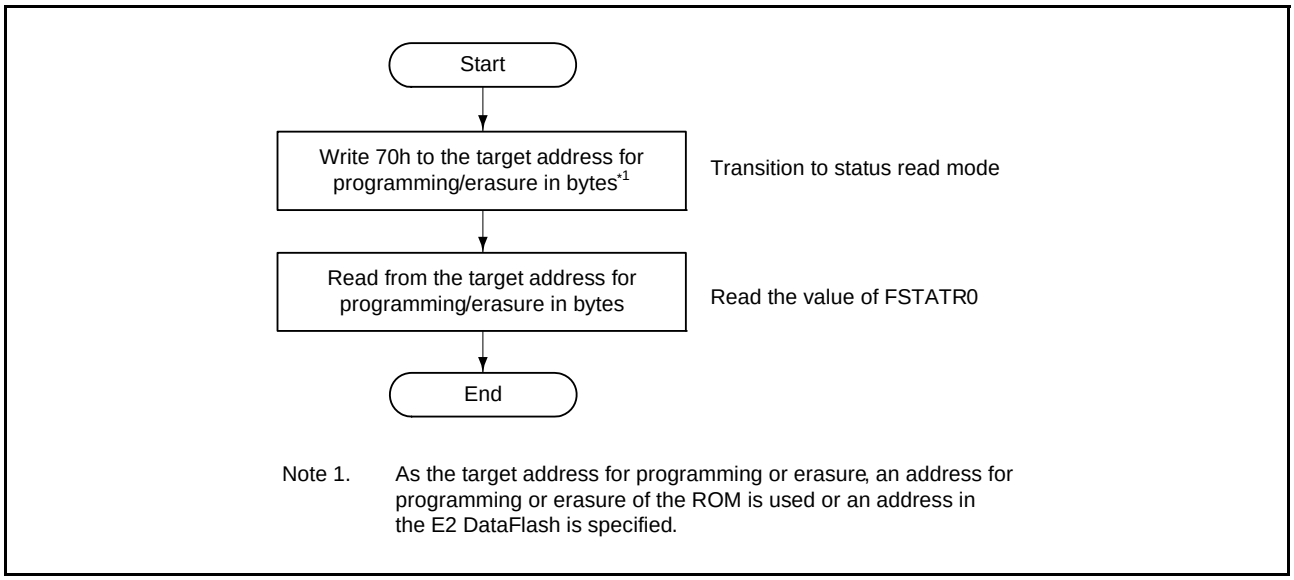


Figure 40.8 Procedure for Transition to ROM Status Read Mode and the Status Checking

(5) Switching to ROM Lock-Bit Read Mode

Clearing the FMODR.FRDMD bit (memory area reading method) issues a lock bit read mode transition (lock bit read 1) command. After the transition to lock bit read mode, the lock bit value is obtained by reading out from a ROM programming/erasure address. All bits of a value thus read have the value of the lock bit of the block that contains the accessed address (Figure 40.9).

Since there are no lock bits for the E2 DataFlash, undefined data are read from the E2 DataFlash area after a transition of the lock bit read mode is made.

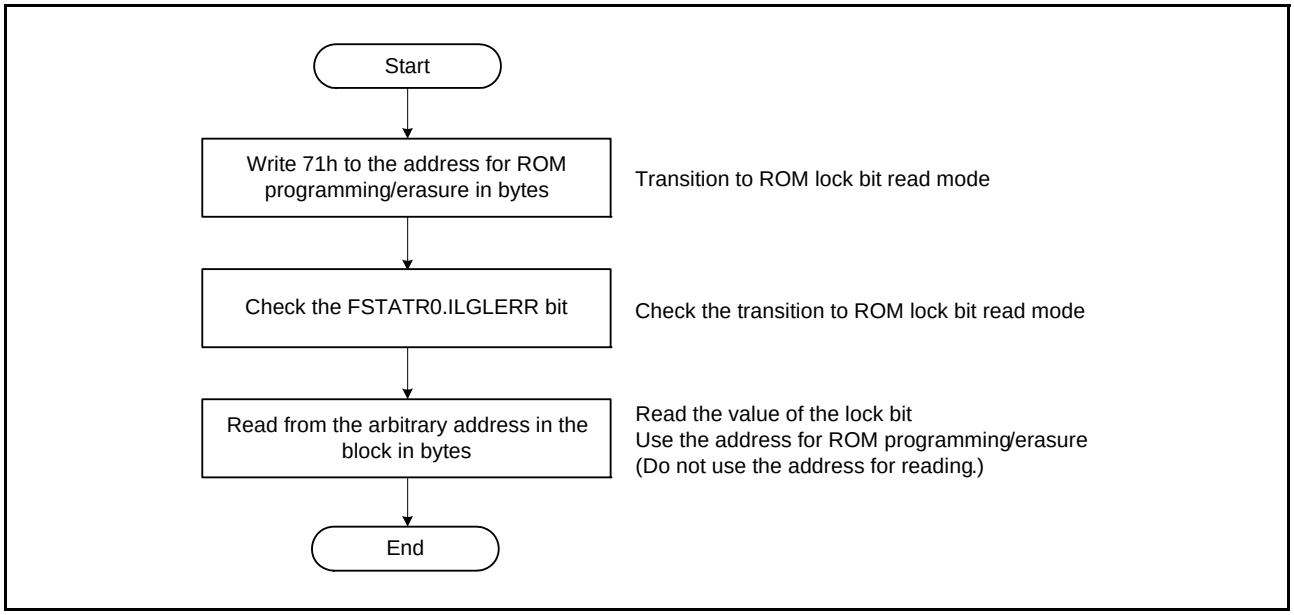
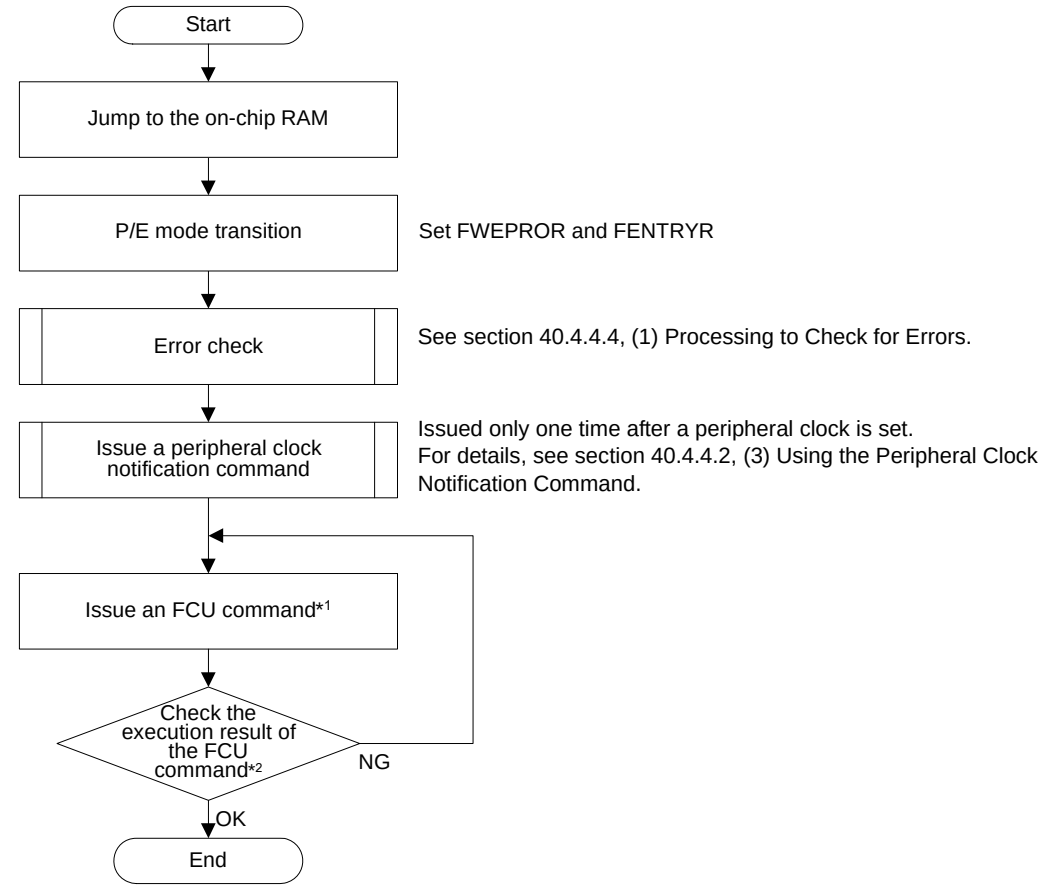


Figure 40.9 Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading

40.4.4.2 Programming and Erasure Procedures

The following passages describe the flow of procedures for programming or erasing the ROM or E2 DataFlash. For details on the acceptance of commands by the FCU, see section 40.4.3, Connections between FCU Modes and Commands.
Figure 40.10 is a simple flowchart of the procedure for executing FCU commands.



Note 1. Selected among the programming, block erase, lock bit programming, and lock bit read 2 commands.
Note 2. To check the result of programming/erasure processing, enter ROM read mode or ROM/E2 DataFlash read mode and read data. (see section 40.4.4.1, (1) Switching to ROM Read Mode or ROM/E2 DataFlash Read Mode).

Figure 40.10 Simple Flowchart of the Procedure for Programming and Erasure

(1) Jumping to Locations in On-chip RAM

Since fetching instructions from the ROM is not possible while the ROM is being programmed or erased, instructions have to be fetched from an area other than the ROM. Copy the required program code to on-chip RAM and then make a jump to the address where the code starts in on-chip RAM.

(2) Transition to P/E Mode

The FCU is placed in P/E mode by setting the FENTRYR.FENTRYn (n = 0 to 3) and FENTRYD bits and the FWEPROR register. For details, see section 40.4.4.1, (2) Switching to P/E Mode.

(3) Using the Peripheral Clock Notification Command

The FlashIF clock (FCLK) is used in programming and erasing the ROM or E2 DataFlash, so the frequency of this clock has to be set in PCKAR. Frequencies in the range from 4 to 32 MHz are selectable. If a frequency within this range has not been set, the FCU will detect the error leading the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

Note that if the PCKA[7:0] bits in the PCKAR register are set to values outside the range from 4 MHz to 32 MHz, do not issue a programming command to the ROM/E2 DataFlash.

The peripheral clock notification command is used after the PCKAR setting has been made. In the first and second cycles for the peripheral clock notification command, respectively, the values E9h and 03h are written to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash. Write 0F0Fh to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash three times in the third to fifth cycles of the command. After 0F0Fh has been written three times (as a word) to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash, the process of the FCU setting the frequency of the peripheral clock starts once the value D0h has been written in the sixth cycle. The FSTATR0.FRDY bit can be used to check whether or not the settings have been completed.

In the case of the ROM, addresses that can be used in the first to sixth cycles differ according to the setting of the FENTRYR.FENTRYn (n = 0 to 3) bits. Ensure that the addresses suit the setting of the FENTRYR.FENTRYn bits. If issuing of the command is attempted for an address in the area for which P/E mode is disabled by the FENTRYR register, the FCU will detect the error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

Furthermore, if the setting for the peripheral clock in use will not be changed from this setting after release from the reset state, the setting by the peripheral clock notification command is also valid for the next FCU command.

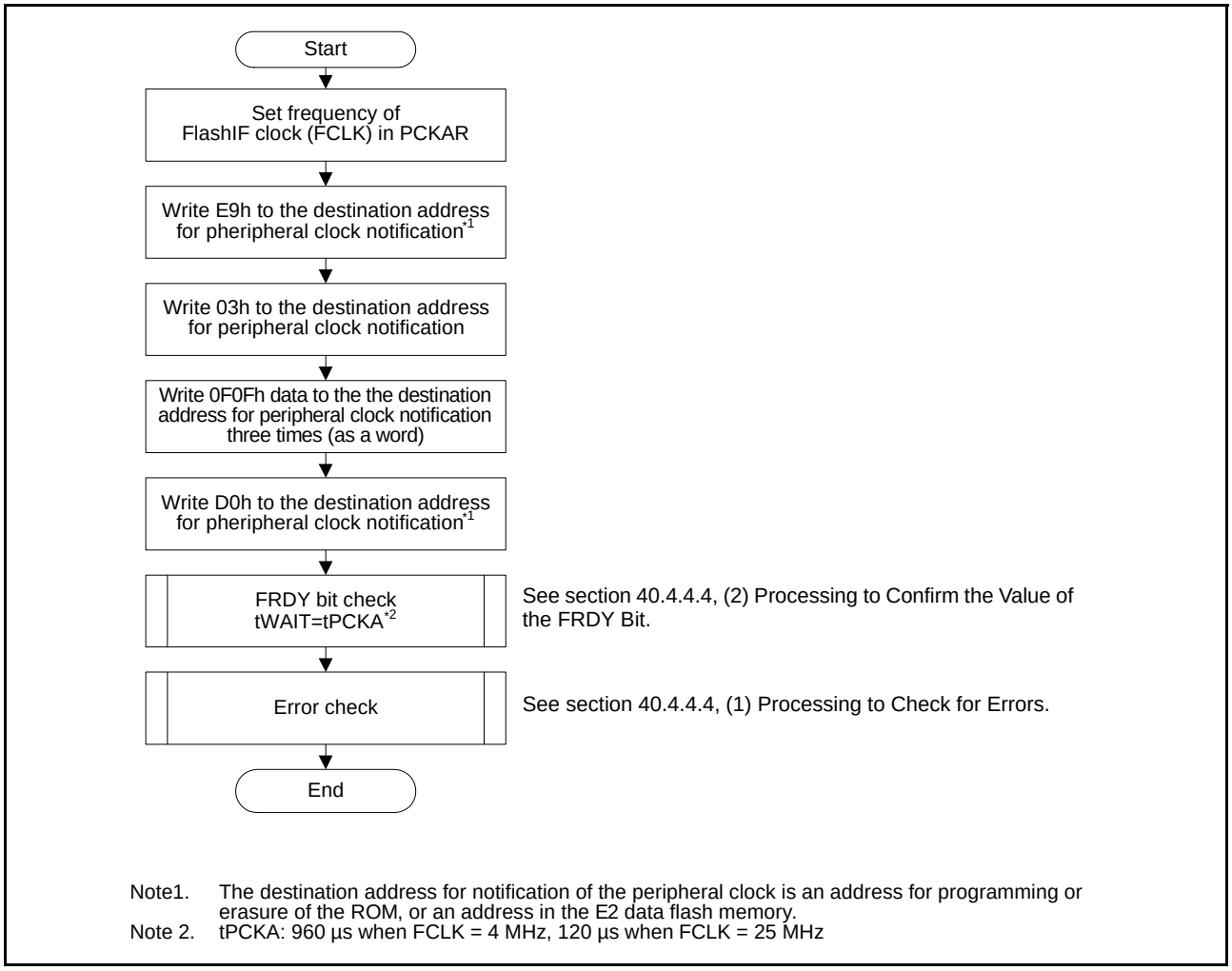


Figure 40.11 Using the Peripheral Clock Notification Command

(4) Programming

The programming command is used to write data to the ROM or the E2 DataFlash.

- ROM Programming

In the first and second cycles for the programming command, respectively, the values E8h and 40h are written to the address range for programming and erasure of the ROM. In the third cycle, write the actual data to be programmed, as a word unit, to the start address of the target area for programming. For this start address, use an address that is aligned on a 128-byte boundary. In the fourth to the 66th cycles, write the data for programming in 63 word-unit rounds to the address range for programming and erasure of the ROM. Once the value D0h has been written to the address range for programming and erasure of the ROM in the 67th cycle, the FCU begins the actual process of programming the ROM. The FSTATR0.FRDY bit can be used to check whether or not the programming has been completed.

Addresses that can be used in the first to 67th cycles differ according to the setting of the FENTRYR.FENTRYn (n = 0 to 3) bits. Ensure that the addresses suit the setting of the FENTRYR.FENTRYn bits. If issuing of the command is attempted for an address in the area for which P/E mode is disabled by the FENTRYR register, the FCU will detect the error leading the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

In cases where the target range in the third to 66th cycles includes addresses that do not require programming, use FFFFh as the data for programming to those addresses. Furthermore, when a lock is programmed so that protection by the lock bit becomes effective, the FPROTR.FPROTCN bit must be set to 1.

- E2 DataFlash Programming

Write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and 01h in the second cycle. In the third cycle, write the first word of data for programming to the address where the target area for programming starts. This address must be on a 2-byte boundary.
After writing words to addresses in the E2 DataFlash area one time, write byte D0h to an address within the E2 DataFlash area in the fourth cycle; the FCU will then start actual programming of the E2 DataFlash. Read the FRDY bit in FSTATR0 to confirm the completion of E2 DataFlash programming.
When programming for locking to prohibit programming and erasure according to the setting of the DFLWEy (y = 0, 1) register proceeds, the relevant bit of the DFLWEy (y = 0, 1) register must be set to 1.

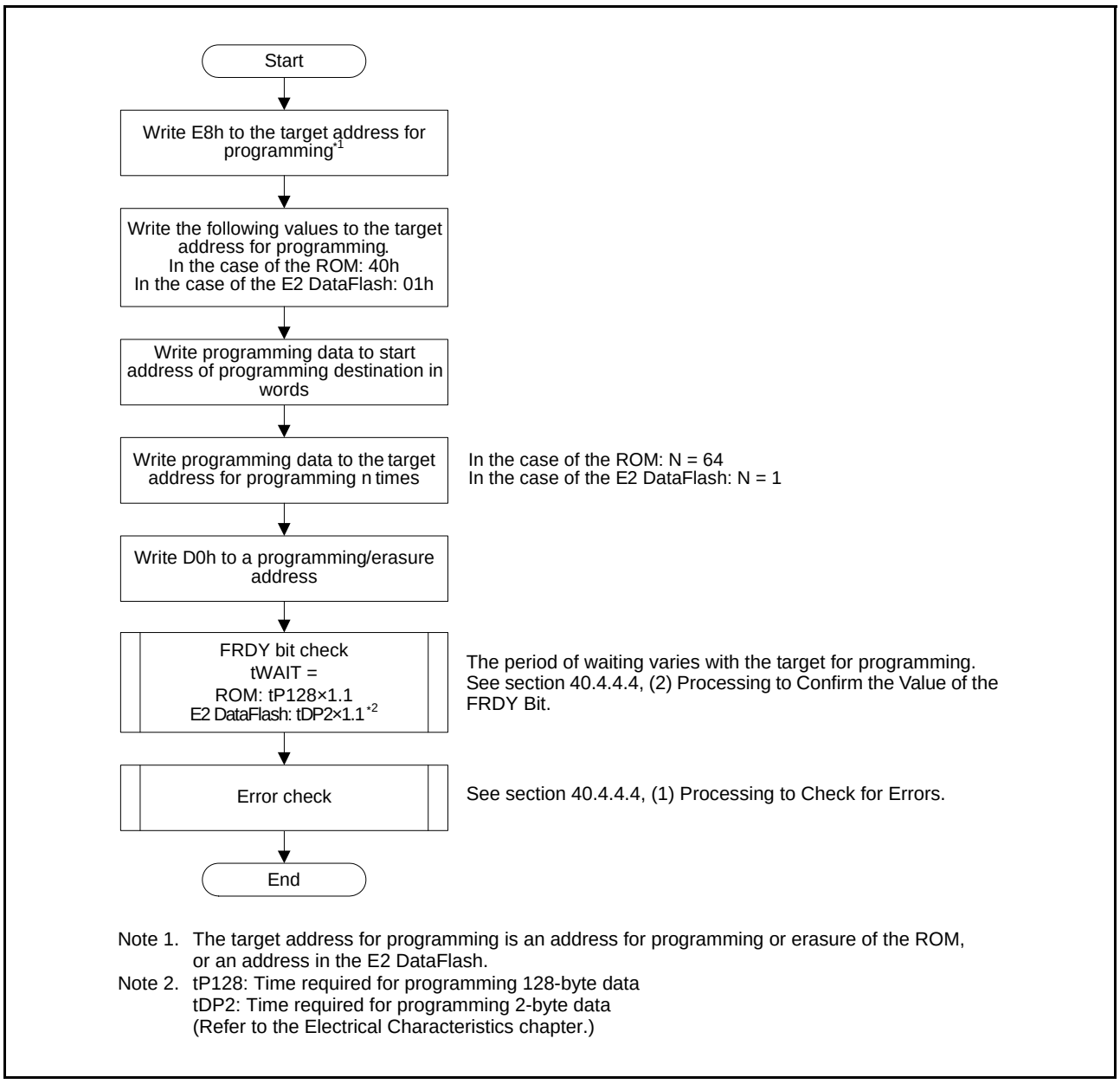


Figure 40.12 Procedure for ROM/E2 DataFlash Programming

(5) Erasure

To erase the ROM/E2 DataFlash, use the block erasure command.
In the first cycle of a block-erasure command, 20h is written to an address for programming or erasure of the ROM or an address in the E2 DataFlash. In the second cycle, when D0h is written to any address within the target block for erasure, the FCU starts processing to erase the ROM or E2 DataFlash. The FSTATR0.FRDY bit can be checked to confirm the completion of erasure. When the CPU reads ROM that has been erased, the value read is FFFF FFFFh. In the case of the E2 DataFlash, values read are undefined. In the case of the ROM, the FPROTR.FPROTCN bit must be set to 1 if protection by the lock bit is in effect for a block to be erased.
Note that the E2 DataFlash has a programming and erasure protection function that is controlled by DFLWEy (y = 0, 1). When erasure for locking to prohibit programming and erasure according to the setting of the DFLWEy (y = 0, 1) register proceeds, the relevant bit of the DFLWEy (y = 0, 1) register must be set to 1.

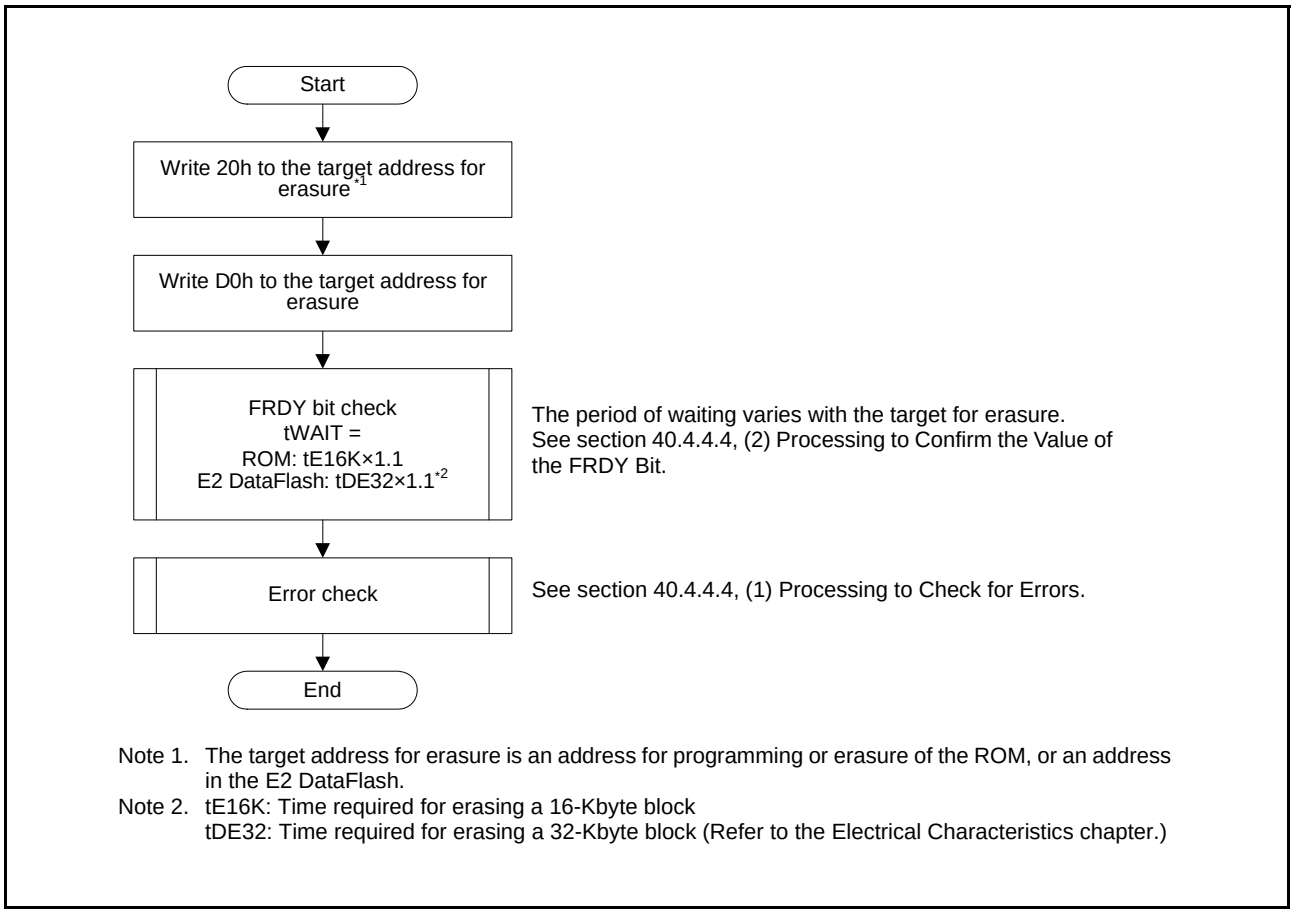
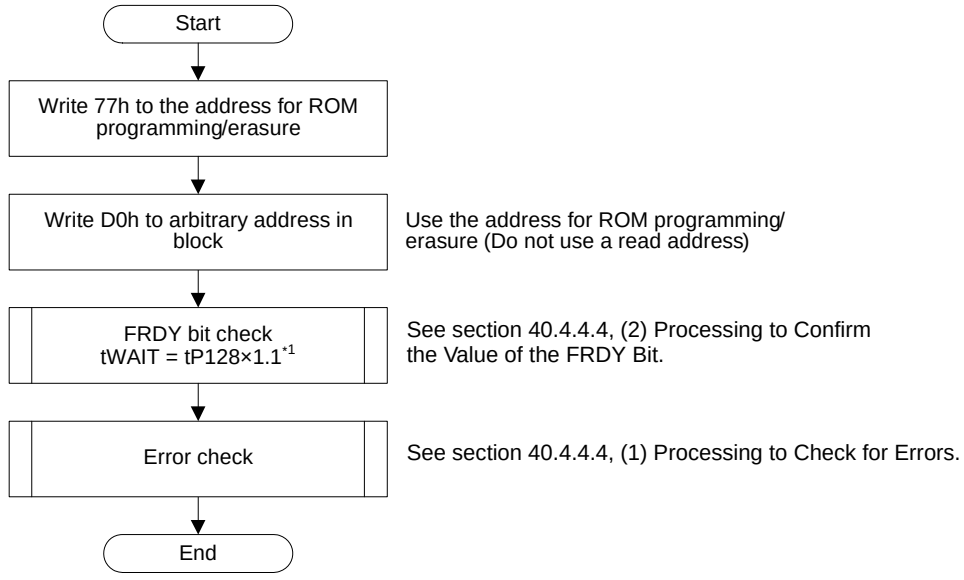


Figure 40.13 Procedure for ROM/E2 DataFlash Erasure

(6) Programming or Erasing Lock Bits

Lock-bit programming can only be executed on the ROM. Each block in the user area includes a lock bit. To write to a lock bit, use the lock bit programming command. In the first cycle of the lock bit programming command, 77h is written to the ROM programming/erasure address. When D0h is written to an arbitrary address in a block whose lock bit is to be written to in the second cycle, the FCU starts the processing to write to the lock bit. Whether writing is completed can be checked with the FSTATR0.FRDY bit.



Note 1. tP128: Time required for programming 128-byte data (Refer to the Electrical Characteristics chapter.)

Figure 40.14 Procedure for Programming the Lock Bit

The block erase command is used to erase lock bits.
When the FPROTR.FPROTCN bit is 0, erasure blocks whose lock bit is set to 0 cannot be erased. When erasing a lock bit, issue a block erase command with the FPROTCN bit set to 1. Using the block erase command erases all data in the block. It is impossible to erase only a lock bit.

(7) Reading Lock Bits

Lock-bit reading can only be executed on the ROM. Lock bits can be read by either reading from a memory area or reading from a register.

The lock bit read 2 command is issued in the case of the register reading method (i.e. when the FMODR.FRDMD bit is set to 1). This command is issued to an address within the block for which the lock bit is to be read; the address range is that for programming and erasing the ROM. In the first and second cycles of the lock bit read 2 command, the values 71h and D0h are written; once these values have been written, the value of the lock bit for the specified block is copied to the FSTATR1.FLOCKST bit.

In the case of the memory area reading method (i.e. when the FMODR.FRDMD bit is 0), the FCU is placed in ROM lock-bit read mode, and the lock bit is obtained by reading from an address within the address range for programming and erasure of the ROM. For details, see Figure 40.9, Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading.

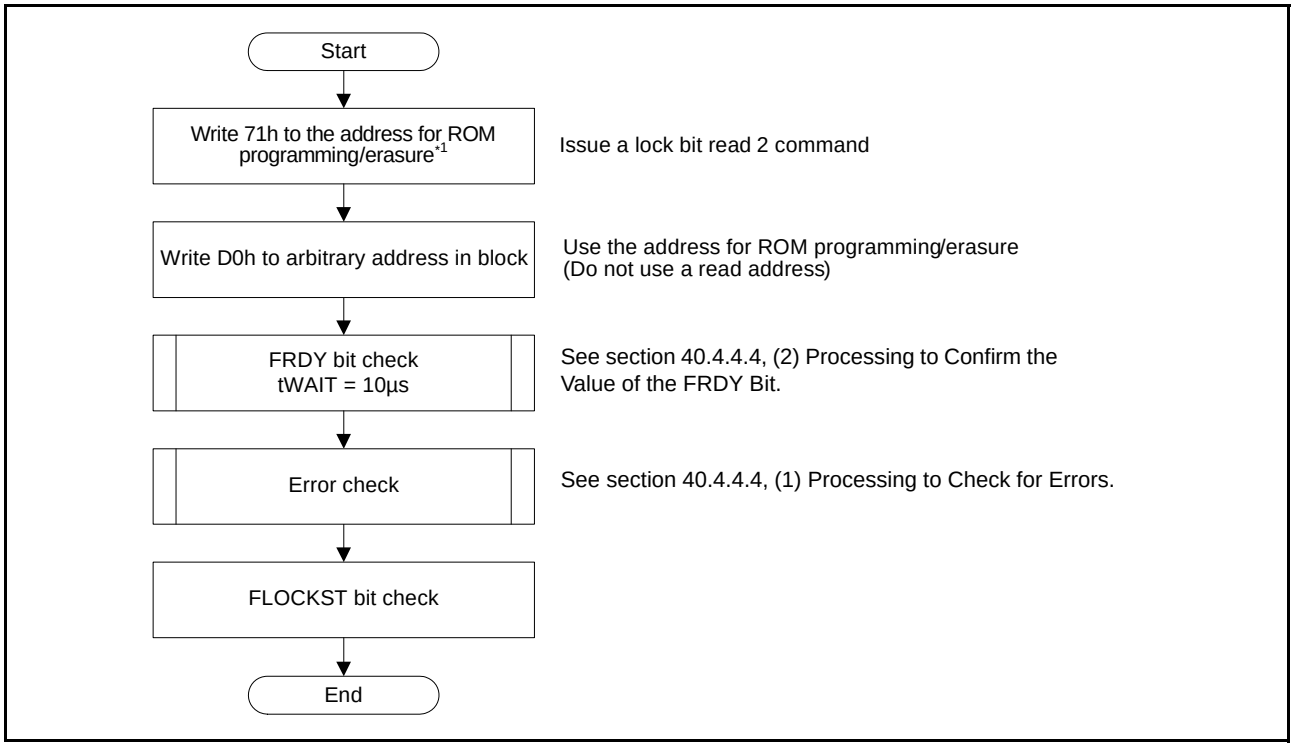
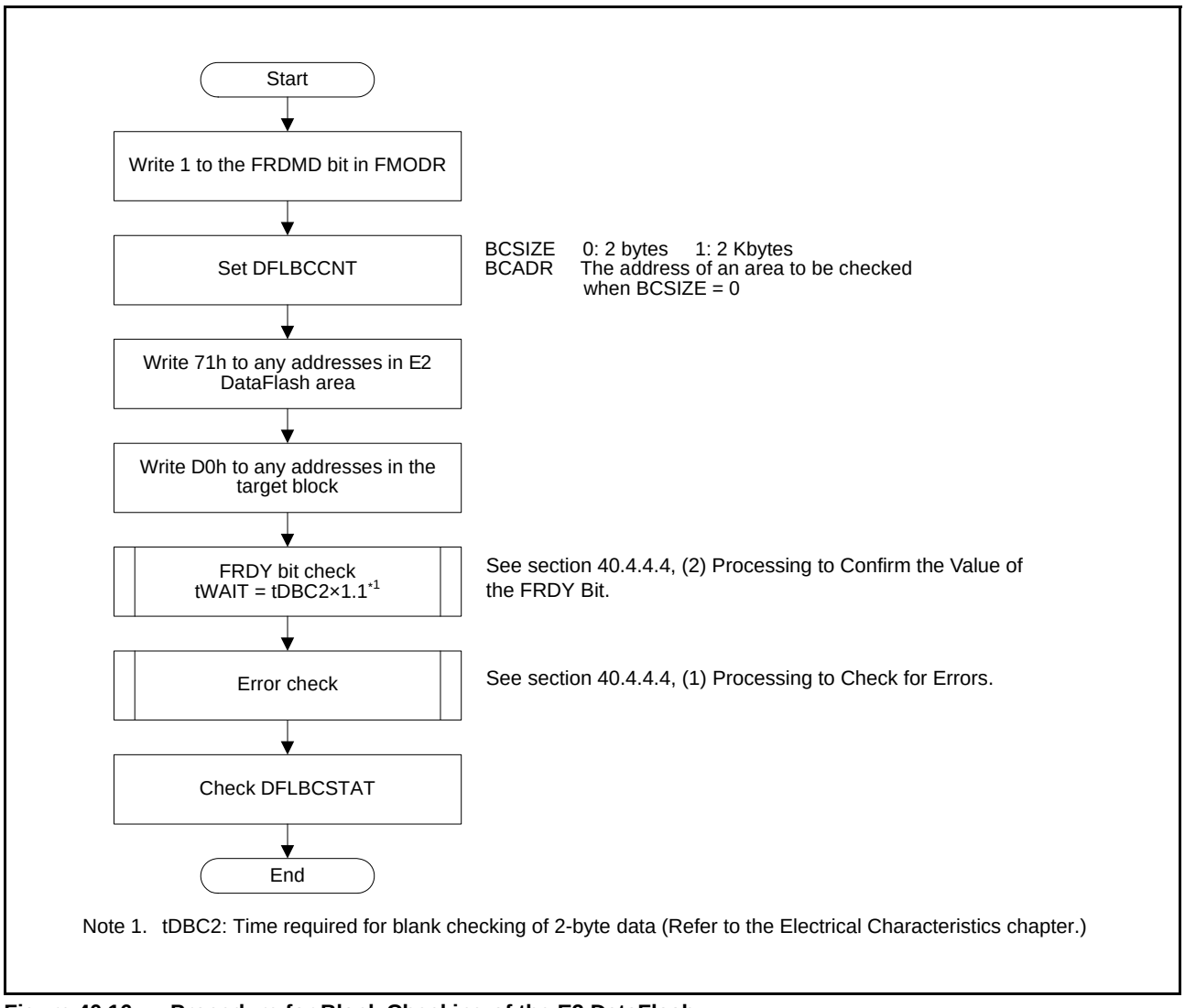


Figure 40.15 Procedure for Reading Lock Bit in Register Read Mode

(8) Blank Checking

Blank checking can only be executed on the E2 DataFlash. Since using the CPU to read erased areas of the E2 DataFlash produces undefined values, the blank checking command should be used to check whether the E2 DataFlash has actually been erased. To make the blank checking command available for use, start by setting the FRDMD bit in FMODR to 1 to enable the command, and then specify the size and start address of the target area in DFLBCCNT. When the DFLBCCNT.BCSIZE bit is 1, checking can be performed on the entire area (2 Kbytes) as specified in the second cycle of the blank check command. When the BCSIZE bit is 0, checking can be performed on the 2-byte range starting from the address obtained by adding the start address of the area as specified in the second cycle of the command and the value held by DFLBCCNT. In the first cycle of the command sequence, the value 71h is written to an address in the E2 DataFlash. In the second cycle, when the value D0h is written to an address in the erasure block within the target area, the FCU starts blank checking of the E2 DataFlash. Test the FRDY bit in the FSTATR0 register to check whether or not the check is complete. On completion of blank checking, check the BCST bit of DFLBCSTAT to see whether the target area has been erased or is filled with 0s and/or 1s.

Figure 40.16 shows the procedure for blank checking of the E2 DataFlash.



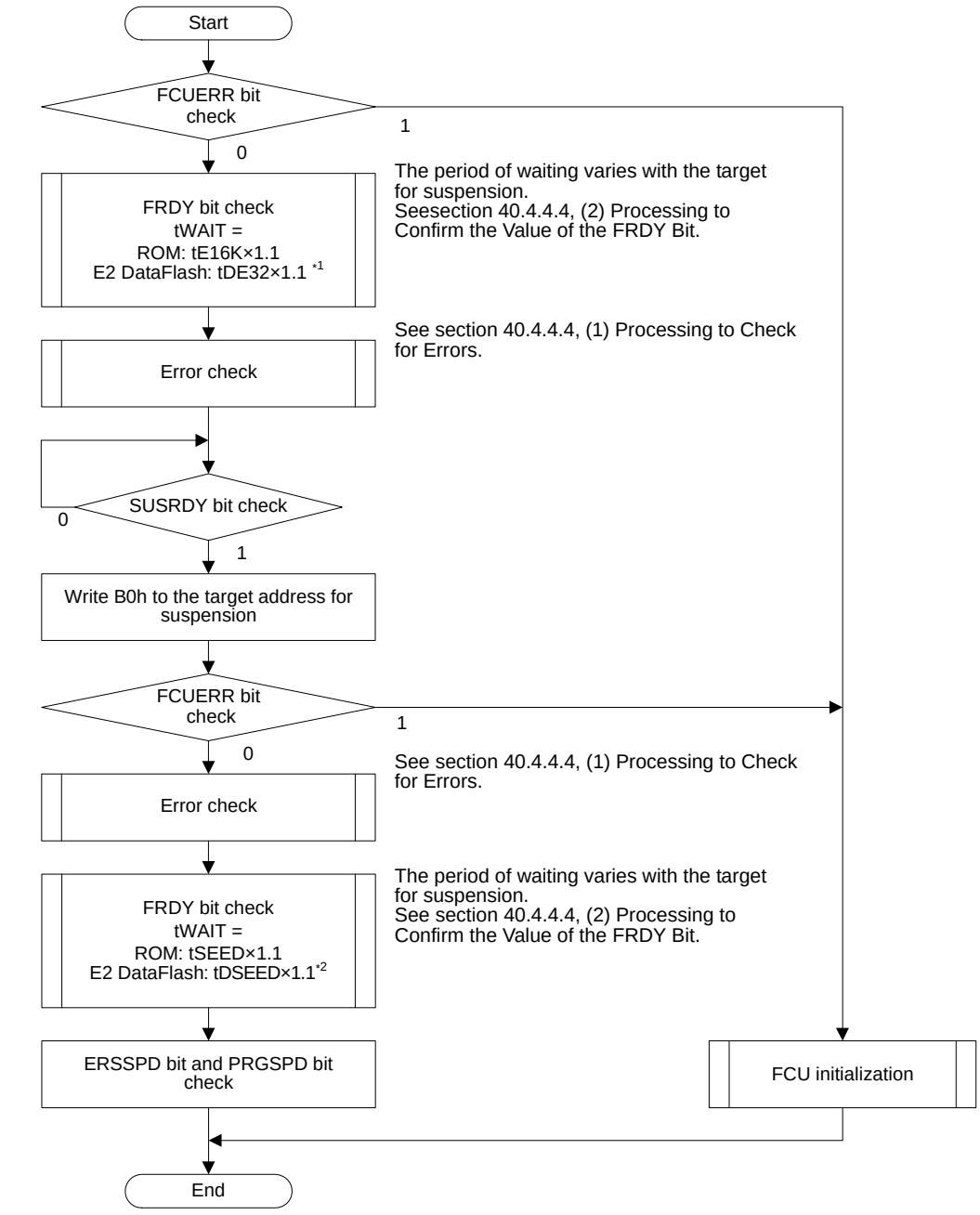
Note 1. tDBC2: Time required for blank checking of 2-byte data (Refer to the Electrical Characteristics chapter.)

Figure 40.16 Procedure for Blank Checking of the E2 DataFlash

40.4.4.3 Suspension and Resumption

(1) Suspending Programming or Erasure

To suspend programming/erasure for the ROM/E2 DataFlash, use the P/E suspend command.
When issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit are 0, and the execution of programming/erasure is normally performed. To confirm that the suspend command can be received, also check that the FSTATR0.SUSRDY bit is 1. After issuing a P/E suspend command, read FSTATR0 and FSTATR1 to confirm that no error occurs.
If an error occurs during programming/erasure, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. When programming/erasure processing has finished during the interval from when the SUSRDY bit is checked as 1 to when a P/E suspend command is received, the ILGLERR bit is set to 1 because the issued P/E suspend command is detected as an illegal command.
When programming/erasure processing has finished simultaneously with the reception of a P/E suspend command, no error occurs and the suspended state is not entered (FSTATR0.FRDY bit is 1 and ERSSPD and PRGSPD bits in FSTATR0 are 0). When a P/E suspend command is received and then the programming/erasure suspend processing finishes normally, the FCU enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is set to 1. After issuing a P/E suspend command, check that the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is 1 and the FCU enters the suspended state, and then decide the subsequent flow. When issuing a P/E resume command in the subsequent flow although the FCU does not enter the suspended state, an illegal command error occurs and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).
If the erasure suspended state is entered, programming to blocks other than an erasure target can be performed. Additionally, the programming and erasure suspended states can shift to ROM read mode by clearing FENTRYR.
For details on FCU operations at the reception of a P/E suspend command, see section 40.5, Suspending Operation.



Note 1. tE16K: Time required for erasing a 16-Kbyte block
tDE32: Time required for erasing a 32-Kbyte block
(Refer to the Electrical Characteristics chapter.)
Note 2. tSEED: Suspend delay time for the ROM
tDSEED: Suspend delay time for the E2 DataFlash
(Refer to the Electrical Characteristics chapter.)

Figure 40.17 Procedure for Programming/Erasure Suspension

(2) Resuming Programming or Erasure

To resume a suspended programming/erasure processing, use the P/E resume command. When the settings of FENTRYR are changed during suspension, reset FENTRYR to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command.

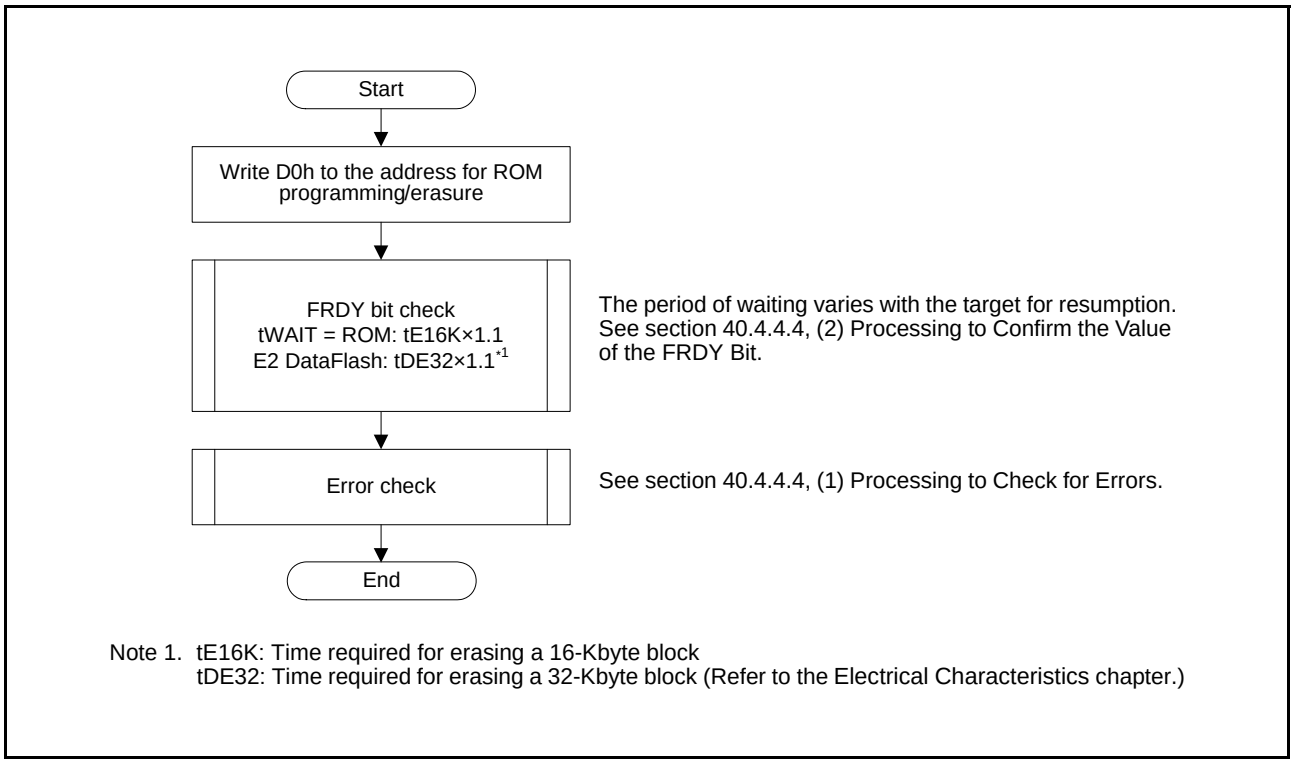


Figure 40.18 Procedure for Resuming Programming or Erasure

40.4.4.4 Processing to Check for Errors and to Confirm the Value of the FRDY Bit

The following passages describe processing to check for errors and processing to confirm the value of the FRDY bit. For details on errors, see section 40.6, Protection.

(1) Processing to Check for Errors

- Checking Flash Status Register 0 (FSTATR0)

To check FSTATR0, read FSTATR0 directly or read the ROM programming/erasure address in ROM status read mode. For the reading in ROM status read mode, see section 40.4.4.1, (4) Switching to Status Read Mode.

- Clearing Flash Status Register 0 (FSTATR0)

To clear the ILGLERR, ERSERR and PRGERR bits in FSTATR0 to 0, use the status register clear command. When one of the ILGLERR, ERSERR and PRGERR bits in FSTATR0 is 1, the FASTAT.CMDLK bit is set to 1 (command-locked state) and the FCU receives no FCU commands other than the status register clear command. If the ILGLERR bit is 1, also check the values of the ROMAE, DFLAE, DFLRPE, and DFLWPE bits in FASTAT. Even if issuing a status register clear command without clearing these bits, the ILGLERR bit is not cleared to 0.
Figure 40.19 shows the flow of processing to check for errors and of the subsequent processing.

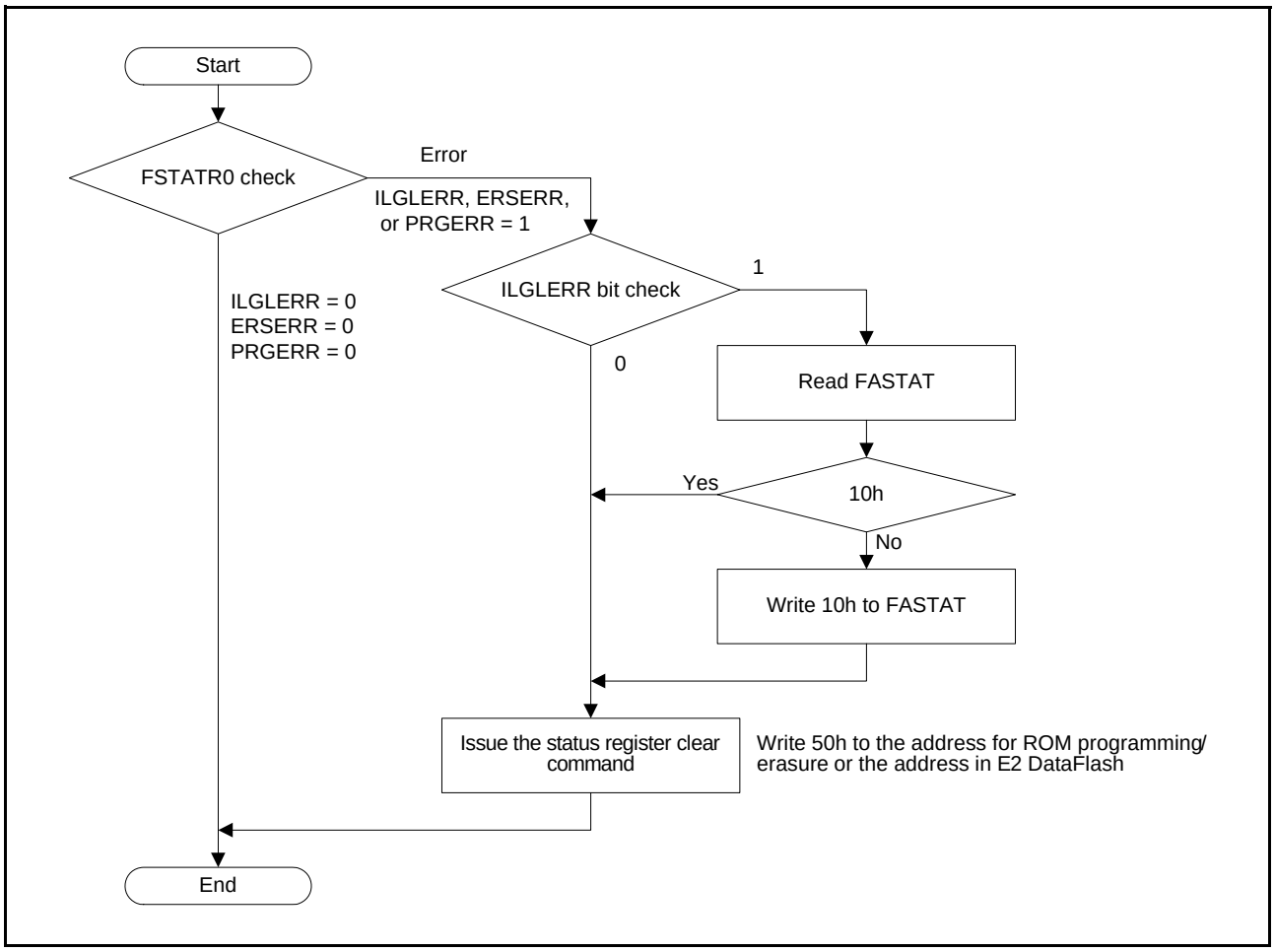


Figure 40.19 Processing to Check for Errors

(2) Processing to Confirm the Value of the FRDY Bit

A period of waiting for processing is required after an FCU command is issued. For details on the period of waiting, see section 41, Electrical Characteristics. When a timeout leads to the FSTATR0.FRDY bit not being set to 1, FRESETR must be used to initialize the FCU. For the initialization of the FCU, see section (3), Initializing the FCU.

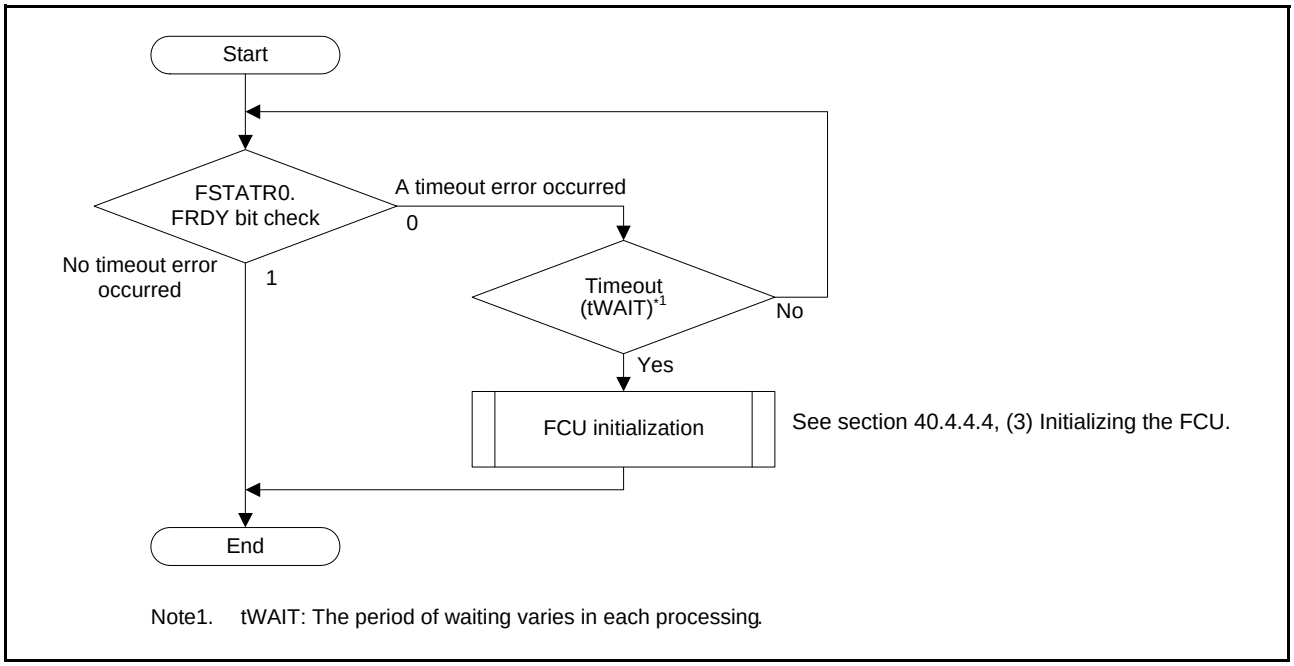


Figure 40.20 Processing to Confirm the Value of the FRDY Bit

(3) Initializing the FCU

If a timeout occurs after an FCU command has been issued, or the FSTATR0.FRDY bit is 1, FRESETR must be used to initialize the FCU. In either case, maintain the FRESETR.FRESET bit set to 1 for a period of tFCR (FCU reset time, see section 41, Electrical Characteristics). Disable reading from the ROM and E2 DataFlash memory during this period of keeping the FRESET bit set to 1. In addition, while the FRESET bit is 1, FCU commands are disabled because FENTRYR is initialized. Restart the processing from the start, as shown in Figure 40.10.

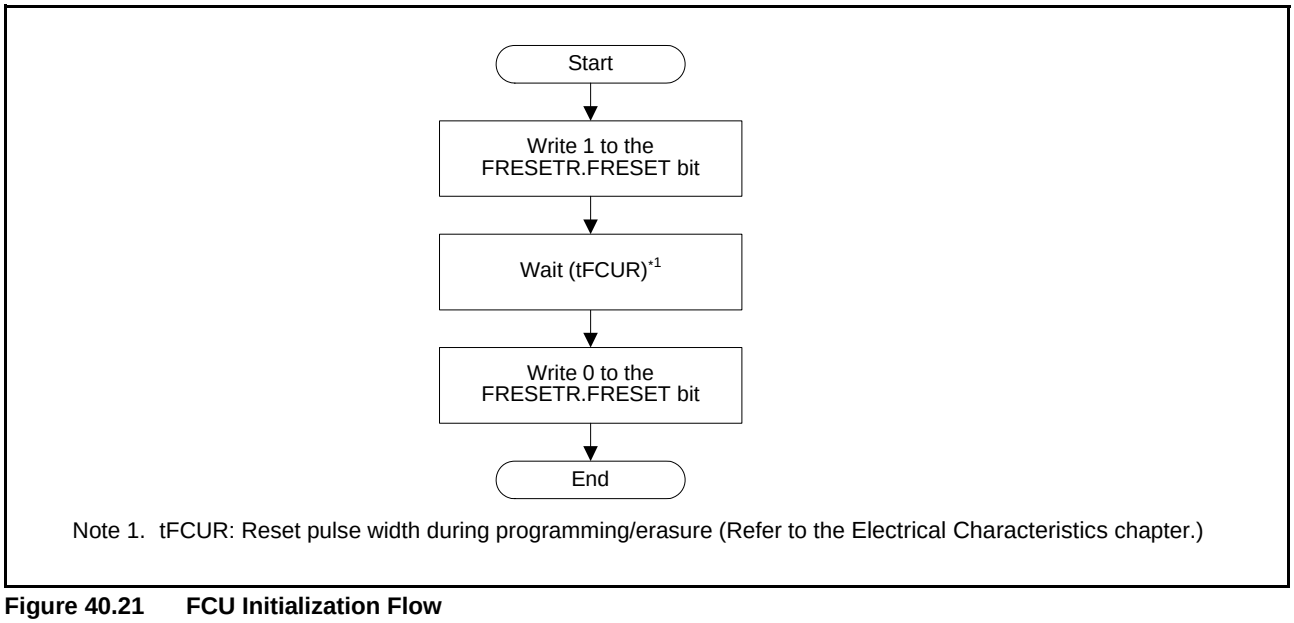


Figure 40.21 FCU Initialization Flow

40.5 Suspending Operation

The ROM/E2 DataFlash cannot be read during programming/erasure. The ROM/E2 DataFlash can be read by suspending the ROM/E2 DataFlash programming/erasure with the P/E suspend command. The P/E suspend command includes one programming mode and two erasure modes (suspension priority mode and erasure priority mode). The P/E resume command that resumes suspended programming/erasure processing is also provided.

40.5.1 Suspension during Programming

When issuing a P/E suspend command during the ROM/E2 DataFlash programming, the FCU suspends programming processing. Figure 40.22 shows the suspend operation of programming.
When receiving a programming-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start programming. If the FCU enters the state in which the P/E suspend command can be received after starting programming, it sets the FSTATR0.SUSRDY bit to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. If the FCU receives a P/E suspend command while a programming pulse is being applied, the FCU continues applying the pulse. After the specified pulse application time, the FCU finishes pulse application, and starts the programming suspend processing and sets the FSTATR0.PRGSPD bit to 1. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the FCU clears the FRDY and PRGSPD bits to 0 and resumes programming.

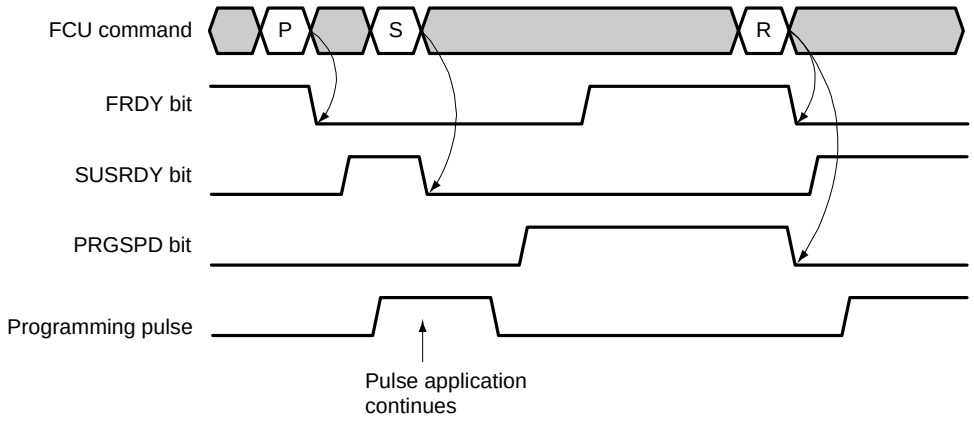


Figure 40.22 Suspension during Programming

40.5.2 Suspension during Erasure (Suspension Priority Mode)

This MCU has a suspension priority mode for the suspension of erasure.

Figure 40.23 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (FCPSR.ESUSPMD bit is 0).

When receiving an erasure-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start erasure. If the FCU enters the state in which the P/E suspend command can be received after starting erasure, it sets the FSTATR0.SUSRDY bit to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. When receiving a suspend command during erasure, the FCU starts the suspend processing and sets the FSTATR0.ERSSPD bit to 1 even if it is applying an erasure pulse. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the FCU clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.

The setting of the FCPSR.ESUSPMD bit affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has never been suspended in the past is being applied, the FCU suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed by a P/E resume command, the FCU continues applying erasure pulse A. After the specified pulse application time, the FCU finishes erasure pulse application and enters the erasure suspended state. When the FCU receives a P/E resume command next and erasure pulse B starts to be newly applied, and then the FCU receives a P/E suspend command again, the application of erasure pulse B is suspended. In suspension priority mode, delay due to suspension can be minimized because the application of an erasure pulse is suspended one time per pulse and priority is given to the suspend processing.

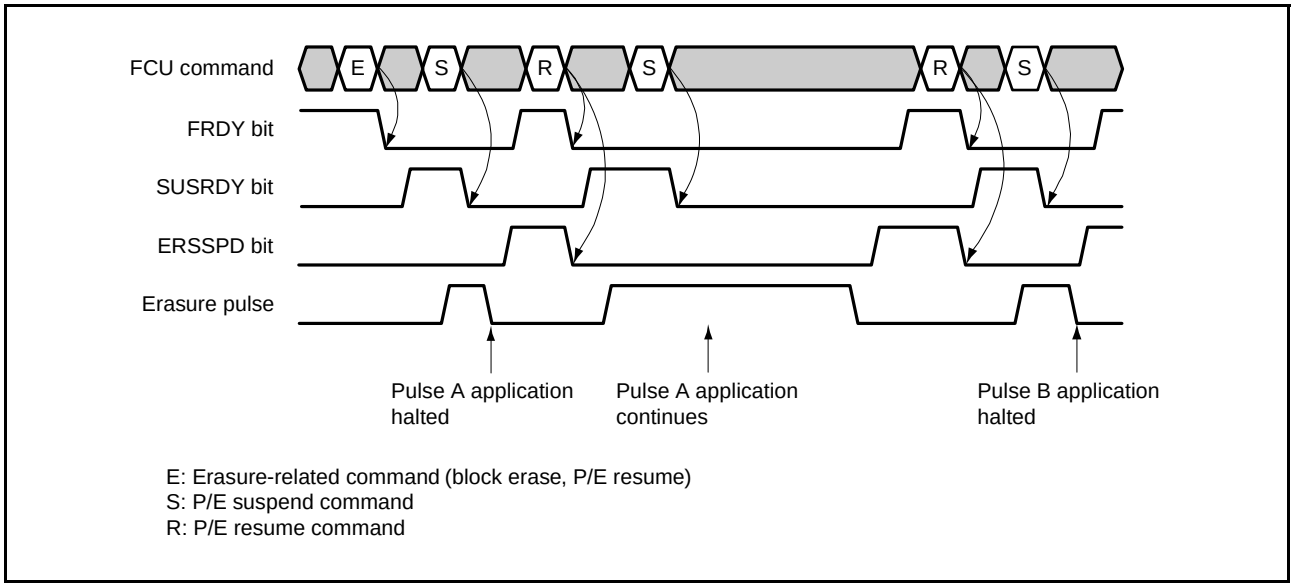


Figure 40.23 Suspension during Erasure (Suspension Priority Mode)

40.5.3 Suspension during Erasure (Erasure Priority Mode)

This MCU has a suspension priority mode for the suspension of erasure.
Figure 40.24 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.
If the FCU receives a P/E suspend command while an erasure pulse is being applied, the FCU definitely continues applying the pulse. In this mode, the required time for the whole erasure processing can be reduced as compared with the suspension priority mode because the reapplication of erasure pulses does not occur when a P/E resume command is issued.

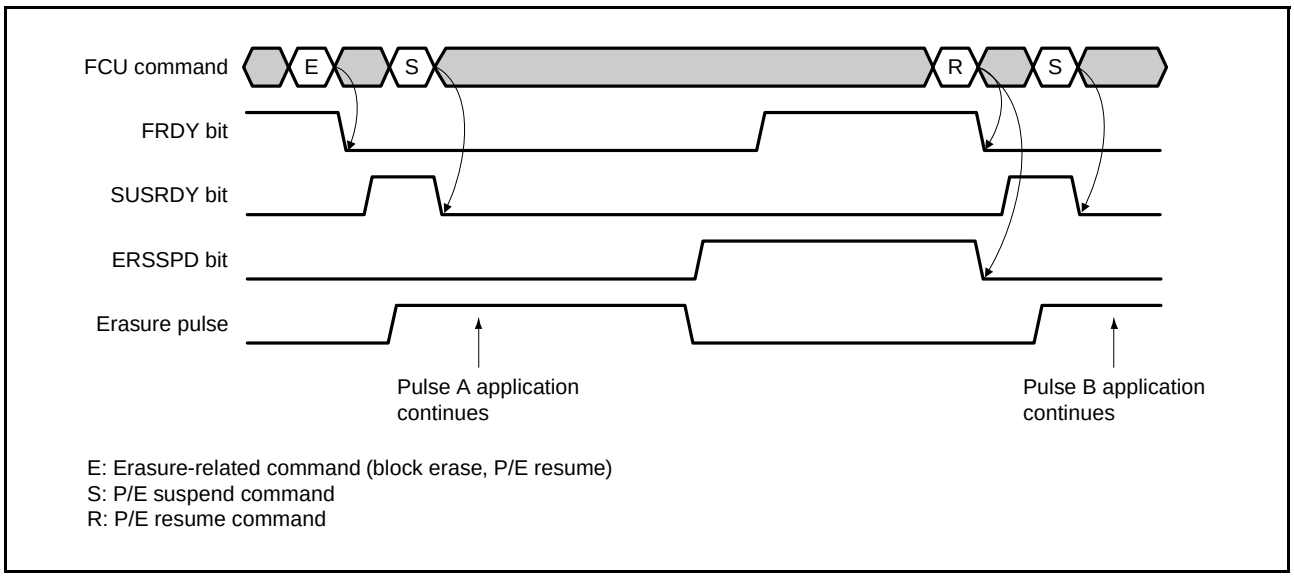


Figure 40.24 Suspension during Erasure (Erasure Priority Mode)

40.6 Protection

Protection against programming/erasure for the ROM/E2 DataFlash includes software protection and the command-locked state.

40.6.1 Software Protection

With the software protection, the ROM/E2 DataFlash programming/erasure is prohibited by the settings of the control registers or user area lock bit. When the software protection is violated and a ROM/E2 DataFlash programming/erasure-related command is issued, the FCU detects an error and the FASTAT.CMDLK bit is set to 1 (command-locked state).

(1) Protection through FWEPROR

If the FWEPROR.FLWE[1:0] bits are not set to 01b, programming cannot be performed in any of the modes.

(2) Protection through FENTRYR

When the FENTRYR.FENTRY3, FENTRY2, FENTRY1, and FENTRY0 bits are 0, ROM/E2 DataFlash read mode is selected. Because the FCU command cannot be received in ROM/E2 DataFlash read mode, ROM/E2 DataFlash programming/erasure is prohibited. When an FCU command is issued in ROM/E2 DataFlash read mode, the FCU detects an illegal command error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

(3) Protection through Lock Bit

Each block in the user area includes a lock bit. When the FPROTR.FPROTCN bit is 0, blocks whose lock bit is set to 0 are prohibited from being programmed/erased. To program or erase blocks whose lock bit is set to 0, set the FPROTCN bit to 1. When the lock bit protection is violated and a ROM programming/erasure-related command is issued, the FCU detects a programming/erasure error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

(4) Protection through DFLWEy

When the DBWEj (j = 00 to 15) bit in DFLWEy (y = 0, 1) is 0, programming and erasure of block DBj in the data area is disabled. If an attempt is made to program or erase block DBj while the DBWEj bit is 0, the FCU detects a programming/erasure protection error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

(5) Protection through DFLREy

When the DBREj (j = 00 to 15) bit in DFLREy (y = 0, 1) is 0, reading of block DBj in the data area is disabled. If an attempt is made to read block DBj while the DBREj bit is 0, the FCU detects a read protection error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 40.6.2, Command-Locked State).

40.6.2 Command-Locked State

With the command-locked state, the FCU detects malfunctions caused by FCU command issuance errors and prohibited access occurrences, and an FCU command is prohibited from being received.
When any bit from among the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FSTATR1.FCUERR bit, and the ROMAE, DFLAE, DFLRPE and DFLWPE bits in FASTAT), the FCU will be in the command-locked state (FASTAT.CMDLK bit is set to 1), so programming and erasure of the ROM/E2 DataFlash are prohibited. To clear the command-locked state, a status register clear command must be issued with FASTAT set to 10h. While the interrupt enable bit in FAEINT is 1, if the corresponding bit in FASTAT is set to 1, a flash interface error (FIFERR) interrupt occurs.
Table 40.12 lists the relationship between the error contents and status bit values (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.ROMAE, DFLAE, DFLRPE, and DFLWPE bits) at error detection.
If a command other than the suspend command is issued during programming/erasure and the FCU enters the command-locked state, it continues programming/erasure. In this state, it is impossible to issue a P/E suspend command and suspend programming/erasure. When a command is issued in the command-locked state, the ILGLERR bit is set to 1.

Table 40.12 Errors and Status Bits

Type	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE	DFLAE	DFLRPE	DFLWPE
FENTRYR setting error	FENTRYR setting is other than 0001h, 0002h, 0004h, 0008h, and 0080h.	1	0	0	0	0	0	0	0
	The FENTRYR setting at suspension disagrees with that at resumption	1	0	0	0	0	0	0	0
Illegal command error (Common to ROM/ E2 DataFlash)	Undefined code is specified in the first cycle of an FCU command	1	0	0	0	0	0	0	0
	Other than D0h is specified in the last cycle of a multi-cycle FCU command	1	0	0	0	0	0	0	0
	The peripheral clock is set to other than 1 to 100 MHz in PCKAR (an error is not detected if the setting is from 1 to 4 MHz or from 32 to 100 MHz)	1	0	0	0	0	0	0	0
	A command other than the suspend command is issued during programming/erasure	1	0	0	0	0	0	0	0
	A suspend command is issued during processing other than programming/ erasure	1	0	0	0	0	0	0	0
	A suspend command is issued in the suspended state	1	0	0	0	0	0	0	0
	A resume command is issued in other than the suspended state	1	0	0	0	0	0	0	0
	A programming/erasure-related command (programming/lock bit programming/block erase) is issued in the programming suspended state	1	0	0	0	0	0	0	0
	A block erase command is issued in the erasure suspended state	1	0	0	0	0	0	0	0
	A programming or lock bit programming command is issued to an erasure suspend target area in the erasure suspended state	1	0	0	0	0	0	0	0
	A command is issued when the FASTAT.CMDLK bit is 1 (in the command-locked state)	1	0/1	0/1	0/1	0/1	0	0	0
Illegal command error (ROM)	Other than 40h is specified in the second cycle of a programming command	1	0	0	0	0	0	0	0
Erasure error	An error occurs during erasure	0	1	0	0	0	0	0	0
	When the FPROTR.FPROTCN bit is 0, a block erase command is issued to an erasure block whose lock bit is set to 0	0	1	0	0	0	0	0	0
Illegal command error (E2 DataFlash)	Other than 01h is specified in the second cycle of a programming command	1	0	0	0	0	0	0	0
	When the FENTRYR.FENTRYD bit is 1, lock bit programming command is issued to the E2 DataFlash area when the FENTRYR.FENTRYD bit is 1	1	0	0	0	0	0	0	0
Programming error	An error occurs during programming	0	0	1	0	0	0	0	0
	When the FPROTR.FPROTCN bit is 0, a programming or lock bit programming command is issued to a block whose lock bit is set to 0	0	0	1	0	0	0	0	0
FCU error	An error occurs during FCU internal processing	0	0	0	1	0	0	0	0
ROM access violation	When the area n (n = 0 to 3) is in the ROM P/E mode (FENTRYR.FENTRYn bit = 1), an attempt is made to read a read address in the area n*1	1	0	0	0	1	0	0	0
	An FCU command was issued for the address for programming/erasure in the area n (n = 0 to 3) while the area n is in read mode (FENTRYR.FENTRYn bit = 0)	1	0	0	0	1	0	0	0
	An attempt is made to read the area n (n = 0 to 3) while in the ROM P/E mode (the FENTRYR register is set)	1	0	0	0	1	0	0	0
E2 DataFlash access error	An attempt is made to read the E2 DataFlash area while in the E2 DataFlash P/E normal mode (FENTRYR.FENTRYD bit = 1)	1	0	0	0	0	1	0	0
	An FCU command is issued for the E2 DataFlash area while FENTRYR.FENTRYD is 0	1	0	0	0	0	1	0	0
	An FCU command is issued for the E2 DataFlash area while the FENTRYR.FENTRYn bits (n = 0 to 3) are 1	1	0	0	0	0	1	0	0
E2 DataFlash read protect error	An attempt is made to read the E2 DataFlash area while it is protected against reading by the DFLREy (y = 0, 1) setting	1	0	0	0	0	0	1	0
E2 DataFlash programming protect error	A programming/erasure command is issued for the E2 DataFlash area while it is protected against programming and erasure by the DFLWEy (y = 0, 1) setting	1	0	0	0	0	0	0	1

Note 1. For the correspondence between ROM capacity and areas, see section 40.1.1, Configuration of the ROM Area.

40.7 User Boot Mode

If the low level is on the MD pin and the high level is on the PC7 pin at the time of release from the reset state, the chip starts in user boot mode. The reset vector at this time points to the address FF7F FFFCh of the user boot area. For other vector tables, refer to normal vector table (see section 14, Interrupt Controller (ICUb)).
In user boot mode, it is possible to perform programming using a given interface; user area or data area can be programmed or erased by issuing the FCU command. Note that programming to the user boot area should be performed in boot mode.

40.8 Boot Mode

40.8.1 System Configuration

In boot mode, the host sends control commands and data for programming, and the user area, data area, and user boot area are programmed or erased accordingly. An on-chip SCI handles transfer between the host and the 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, the program on the boot area is executed. This program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.
Figure 40.25 shows the system configuration for operations in boot mode.
Input and output pins associated with the ROM/E2 DataFlash are listed in Table 40.13.

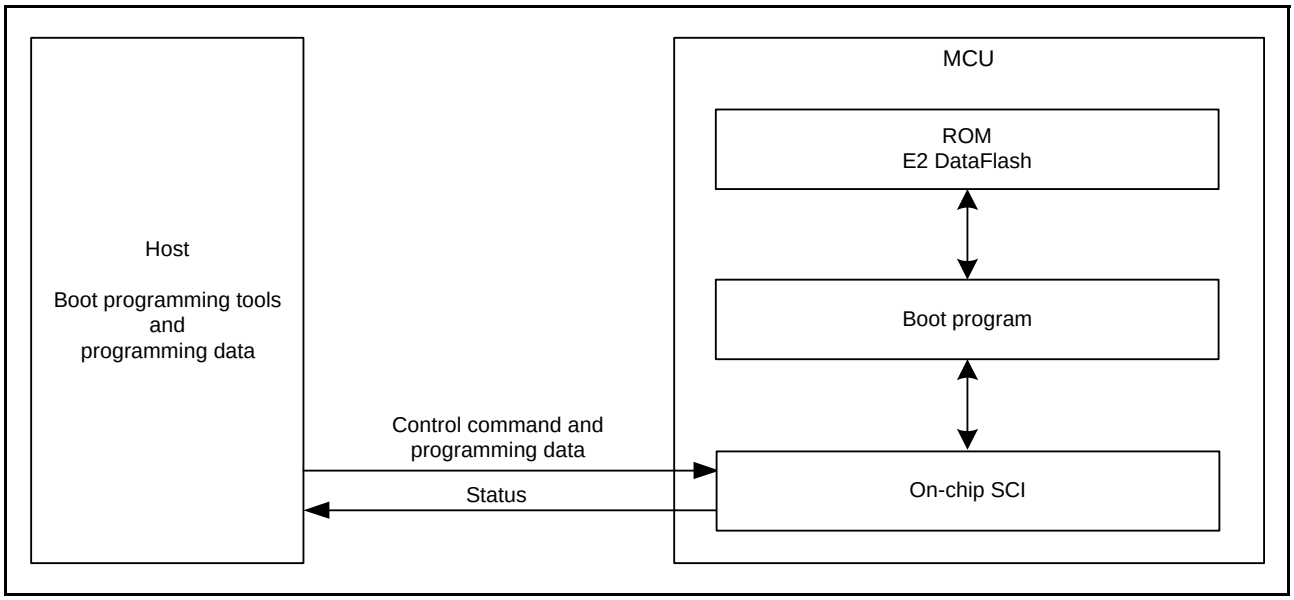


Figure 40.25 System Configuration for Operations in Boot Mode

Table 40.13 Input and Output Pins Associated with the ROM/E2 DataFlash

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode User boot mode	Selection of operation mode
PC7	Input		Selection of boot mode (SCI boot) or user boot mode
P30/RXD1	Input	Boot mode	For host communication (to receive data through SCI)
P26/TXD1	Output		For host communication (to transmit data through SCI)

40.8.2 State Transitions in Boot Mode

Figure 40.26 is a diagram of the state transitions in boot mode.

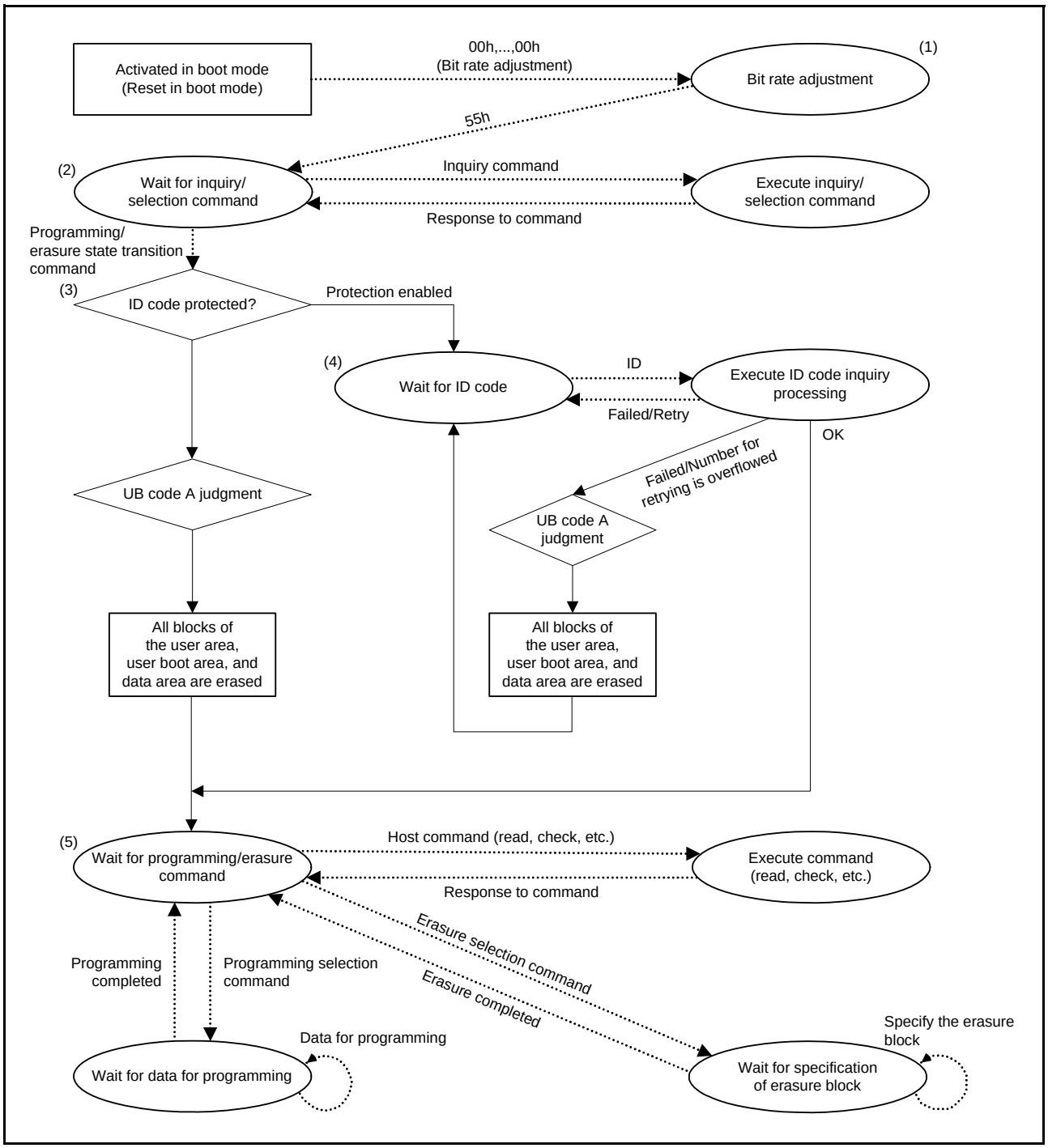


Figure 40.26 State Transitions in Boot Mode

(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 wait for a command for inquiry or selection. For details on matching of the bit rates, see section 40.8.3, Automatic Adjustment of the Bit Rate.

(2) Waiting for a Command for Inquiry or Selection

This state is for inquiries on the area size, the area configuration, the addresses where the areas start, the state of support, etc., and for selection of the device, clock mode, and bit rate. This MCU receives a programming/erasure state transition command issued by the host and then enters the state to determine whether ID code protection is enabled or disabled. For the inquiry/selection commands, see section 40.8.7, Inquiry/Selection Command Wait.

(3) Judging ID Code Protection

This state is for determining whether ID code protection is enabled or disabled. The control code and ID code written in ROM are used to determine whether ID code protection is enabled or disabled. When enabled, the state of waiting for the ID code is entered. When disabled, the user area and data area are completely erased, and the wait for programming and erasure commands is entered. For details on the control code and ID code, see section 40.8.4, ID Code Protection (Boot Mode).

(4) Waiting for an ID Code

This state is for waiting for the control code and ID code to be sent from the host. The control code and ID code sent by the host are compared with the code stored in ROM, and the state of waiting for programming and erasure commands is entered if the two match. If they do not match, the next transition is back to the state of waiting for an ID code. However, if the ID codes fail to match three times in a row and also the state of protection is authentication method 1, the ROM is completely erased, and the state of waiting for an ID code is entered again. Turn the power off and start all over. For details on the control code and ID code, see section 40.8.4, ID Code Protection (Boot Mode).

(5) Waiting for a Command for Programming or Erasure

In this state, programming and erasure proceed in accordance with commands from the host. In response to the reception of a command, this MCU enters the wait for the data to use in programming, the wait for specification of the erasure block to be erased, or the state of executing the processing of commands, such as read and check.
When this MCU receives a programming selection command, it enters the state of waiting for the data to use in programming. After the host has issued the programming selection command, the process continues with the address where programming is to start and then the data for programming. Setting of FFFF FFFFh as the address where programming is to start indicates the completion of programming, and the next transition is from the wait for the data to use in programming to the wait for programming and erasure commands.
When this MCU receives an erasure selection command, it enters the state of waiting for specification of the erasure block to be erased. After the host has issued the erasure selection command, the process continues with the number of the erasure block to be erased. Setting of FFh as the number of the erasure block indicates the completion of erasure, and the next transition is from the wait for specification of the erasure block to the wait for programming and erasure commands. Since the user area, user boot area and data area are all completely erased during the interval between booting up in boot mode and transition to the wait for programming and erasure commands, execution of erasure is not necessary unless data newly programmed in boot mode is to be erased without a further reset.
Other than the programming and erasure commands, commands for execution in this state include those for checksum of the user area and user boot area, blank checking, reading from memory, and acquiring status information.

40.8.3 Automatic Adjustment of the Bit Rate

When this MCU is booted up in boot mode, 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 or 19,200 bps. This MCU calculates the host's SCI bit rate from the measured periods at low level, adjusts its own bit rate accordingly, 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 E6h to the host, and if successful reception of 55h by this MCU is not possible, it responds by sending FFh to the host.

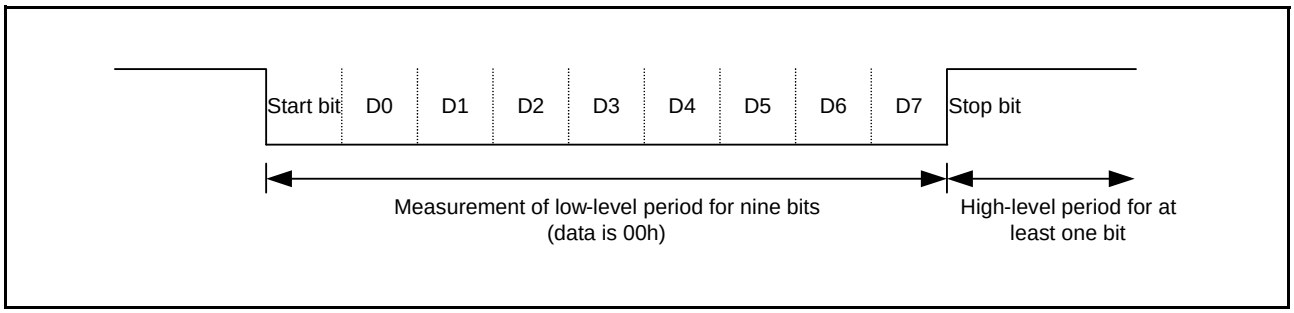


Figure 40.27 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

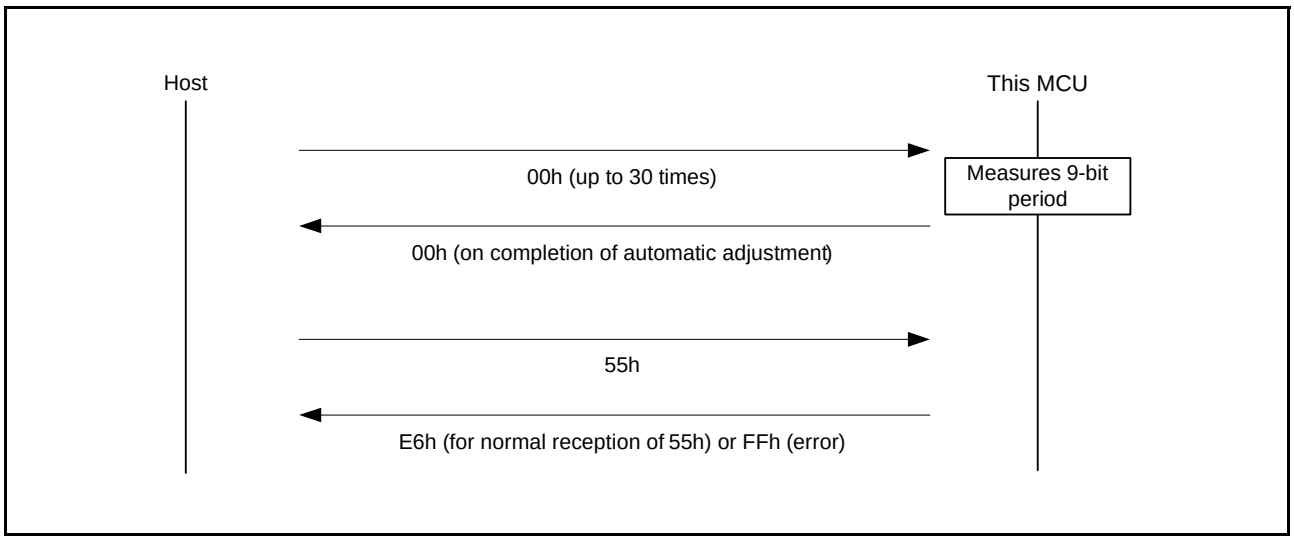


Figure 40.28 Sequence of Transfer between Host and this MCU

Since the bit rate of this MCU depends on the bit rate of SCI communications and the frequency of this MCU's peripheral clock, adjustment to match the bit rate will not be possible under some conditions. Accordingly, ensure that SCI communication is under the conditions given in Table 40.14.

Table 40.14 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host	Frequency Range for EXTAL Signal
9,600 bps	4 to 20 MHz*1
19,200 bps	8 to 20 MHz*1

Note 1. The minimum frequency of the resonator is 8 MHz.

40.8.4 ID Code Protection (Boot Mode)

This function is used to prohibit reading/programming/erasure from the host such as the PC.
After automatic adjustment of the bit rate when booting up in boot mode, the ID code transmitted from the host and the control and ID codes written to the ROM are used to determine disabling or enabling of ID code protection. When ID code protection is enabled, the code sent from the host is compared with the control code and ID code in the ROM to determine whether they match, and reading/programming/erasure will be enabled only when the two match.
The control code and ID code in the ROM consists of four 32-bit words. Figure 40.29 shows the configuration of the control code and ID code. The ID code should be set in 32-bit units.

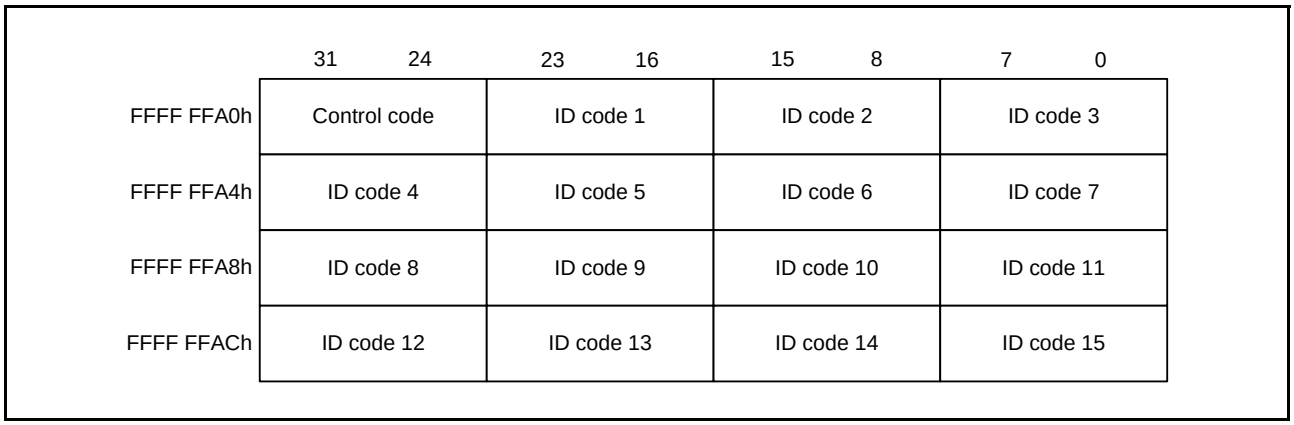


Figure 40.29 Configuration of Control Code and ID Code in ROM

(1) Control Code

The control code determines whether ID code protection is enabled or disabled and the method of authentication to use with the host. Table 40.15 lists how the control code determines the method of authentication.

Table 40.15 Specifications for ID Code Protection

Control Code	ID Code	State of Protection	Operations at the Time of SCI Connection
45h	As desired	Protection enabled (authentication method 1)	Matching ID code: The command wait is entered. Non-matching ID code: The ID code protection wait is entered again. However, if a non-matching ID code is received three times in a row, all blocks are erased.
52h	Sequences other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 2)	Matching ID code: The command wait is entered. Non-matching ID code: The ID code protection wait is entered again.
	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 3)	Judged to be a non-matching ID code.
Other than above	—	Protection disabled	All blocks are erased.

(2) ID Code

The ID code can be set to any desired value. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh (from the ID code 1 field), there is no determination of matching and the ID code is considered to be non-matching. Accordingly, reading, programming, and erasure from the host are prohibited.

(3) Program Example for ID Code Setting

The following assembler directives set up a control code of 45h and an ID code of 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh (from the ID code 1 field).

```
.SECTION  ID_CODE,CODE
.ORG  0FFFFFFA0h
.LWORD  45010203h
.LWORD  04050607h
.LWORD  08090A0Bh
.LWORD  0C0D0E0Fh
```

40.8.5 UB Code A

For the UB Code A, see section 7.3, UB Codes.

40.8.6 Configuration of Commands and Responses

Communications between the host and this MCU in boot mode consist of commands sent by the host and responses sent as replies from this MCU.
The notation “SUM” in descriptions of commands means the checksum, in the case of the total of the bytes transmitted by this MCU, and byte data for which the total becomes 00h is indicated. “Size” indicates the size of the command from the byte where it starts, and is the number of bytes transmitted or received other than the SUM byte.

Moreover, if the host has sent an undefined command, this MCU returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response	80h	Command

40.8.7 Inquiry/Selection Command Wait

Table 40.16 lists the commands available in the inquiry/selection command wait. The boot program status inquiry command can also be used in the programming/erasure command wait. The other commands can only be used in the inquiry/selection command wait.

Table 40.16 Inquiry/Selection Commands

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the series name
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the number of clock modes and their values
Clock mode selection	Notifies the selected clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot area information inquiry	Inquires regarding the number of user boot area and the start and end addresses
User area information inquiry	Inquires regarding the number of user area and the start and end addresses
Block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
Data area inquiry	Inquires regarding the availability of data area
Data area information inquiry	Inquires regarding the number of data areas and the start and end addresses
New bit rate selection	Modifies the bit rate of SCI communications between the host and this MCU
Programming/erasure state transition	Enters the state for determining ID code protection
Boot program status inquiry	Inquires regarding the processing state

In the inquiry/selection command wait, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up this MCU according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, this MCU returns a response indicating a command error. Figure 40.30 shows an example of the procedure to use commands in the inquiry/selection command wait.

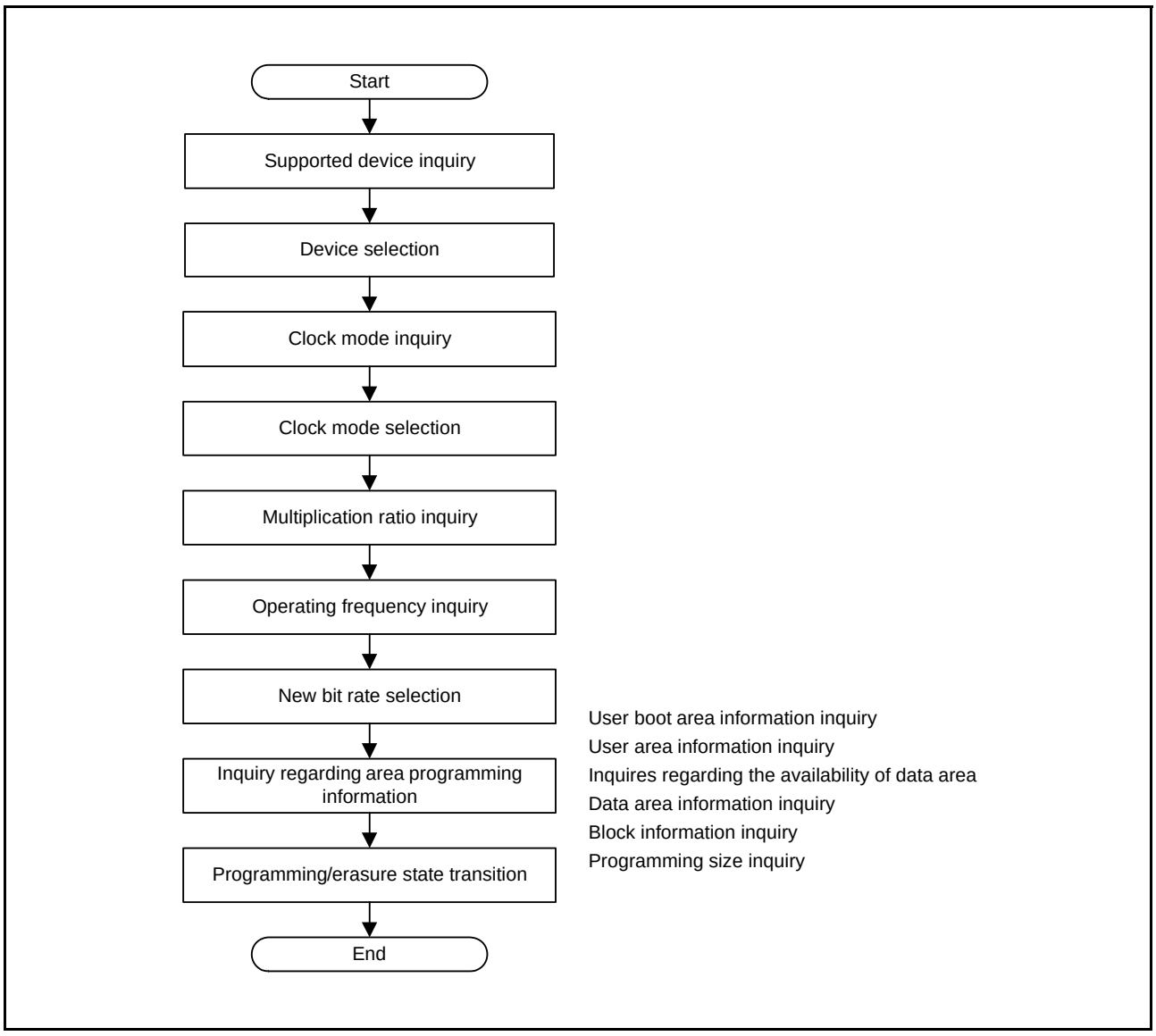


Figure 40.30 Example of Procedure to Use Inquiry/Selection Commands

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, this MCU returns the information concerning the devices supported by the boot program for boot mode. If the supported device inquiry command comes after the host has selected a device, this MCU only returns the information concerning the selected device. In response to supported device inquiry commands, this MCU transmits two sets of device information in turn, one in little endian and the other in big endian.

Command	20h			
Response	30h	Size	Device count	
	Character count	Device code (little endian is specified)		Series name
	Character count	Device code (big endian is specified)		Series name
	SUM			

Size (1 byte):	Total number of bytes in the device count, character count, device code, and series name fields
Device count (1 byte):	Number of device types supported by the boot program for boot mode
Character count (1 byte):	Number of characters included in the device code and series name fields
Device code (4 bytes):	Chip recognition code
Series name (n bytes):	ASCII code for the supported device
SUM (1 byte):	Checksum

(2) Device Selection

In response to a device selection command sent from the host, this MCU checks if the selected device is supported. When the selected device is supported, this MCU and returns a response (06h). If the selected device is not supported or the sent command is illegal, this MCU returns an error response (90h).

Select the device code with the endian specification from the two sets of device information transmitted in response to a supported device inquiry command in accord with the written data.

Command	10h	Size	Device code	SUM
Response	06h			
Error response	90h	Error		

Size (1 byte):	Number of characters in the device code field (fixed at 4)
Device code (4 bytes):	ASCII code for the series name of the chip (same code as the response to the supported device inquiry command)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error (illegal command) 21h: Device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, this MCU returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, this MCU only returns the information concerning the selected clock mode.

Command	21h	
Response	31h	Size
	Mode	
	SUM	

Size (1 byte): Total number of bytes in the mode count and mode fields

Mode (1 byte): Supported clock mode (for example, 01h indicates clock mode 1)

SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, this MCU checks if the selected clock mode is supported. When the selected mode is supported, this MCU specifies this clock mode for use and returns a response (06h). If the selected mode is not supported or the sent command is illegal, this MCU returns an error response (91h). Be sure to issue a clock mode selection command only after issuing a device selection command. Even when 00h or 01h has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	11h	Size	Mode	SUM
Response	06h			
Error response	91h	Error		

Size (1 byte): Number of characters in the mode field (fixed at 1)

Mode (1 byte): Clock mode (same mode as the response to the clock mode inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code
11h: Checksum error (illegal command)
22h: Clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, this MCU returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

22h

Response

32h	Size	Clock type count		
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
SUM				

Size (1 byte):	Total number of bytes in the clock type count, multiplication ratio count, and multiplication ratio fields
Clock type count (1 byte):	Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
Multiplication ratio count (1 byte):	Number of supported multiplication/division ratios (for example, 04h indicates that four multiplication ratios are supported for the system clock (multiplied by 1, multiplied by 2, multiplied by 4, and multiplied by 8)
Multiplication ratio (1 byte):	A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4) A negative value indicates a division ratio (for example, FEh = –2 = divided by 2)
SUM (1 byte):	Checksum

(6) Operating Frequency Inquiry

In response to an operating frequency inquiry command sent from the host, this MCU returns the minimum and maximum operating frequencies for each clock.

Command

23h

Response

33h	Size	Clock type count
Minimum frequency		Maximum frequency
Minimum frequency		Maximum frequency
SUM		

Size (1 byte):	Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields
Clock type count (1 byte):	Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
Minimum frequency (2 bytes):	Minimum value of the operating frequency (for example, 07D0h indicates 20.00 MHz). This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.
Maximum frequency (2 bytes):	Maximum value of the operating frequency This value is represented in the same format as the minimum frequency
SUM (1 byte):	Checksum

(7) User Boot Area Information Inquiry

In response to a user boot area information inquiry command sent from the host, this MCU returns the number of user boot areas and their addresses.

Command	24h		
Response	34h	Size	Area count
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of user boot areas (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a user boot area
Area end address (4 bytes):	End address of a user boot area
SUM (1 byte):	Checksum

(8) User Area Information Inquiry

In response to a user area information inquiry command sent from the host, this MCU returns the number of user areas and their addresses.

Command	25h		
Response	35h	Size	Area count
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of user areas (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a user area
Area end address (4 bytes):	End address of a user area
SUM (1 byte):	Checksum

(9) Block Information Inquiry

In response to a block information inquiry command sent from the host, this MCU returns the number of total erasure blocks in the user area and data area, and their addresses.

Command	26h		
Response	36h	Size	Block count
	Block start address		
	Block end address		
	Block start address		
	Block end address		
	. . .		
	Block start address		
	Block end address		
	SUM		

Size (2 bytes):	Total number of bytes in the block count, block start address, and block end address fields
Block count (1 byte):	Number of erasure blocks in the user area
Block start address (4 bytes):	Start address of an erasure block
Block end address (4 bytes):	End address of an erasure block
SUM (1 byte):	Checksum

(10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, this MCU returns the programming size.

Command	27h			
Response	37h	Size	Programming size	SUM

Size (1 byte):	Number of characters included in the programming size field (fixed at 2)
Programming size (2 bytes):	Programming unit (bytes)
SUM (1 byte):	Checksum

(11) Data Area Inquiry

In response to a data area inquiry command sent from the host, this MCU returns the information concerning the availability of data areas.

Command	2Ah			
Response	3Ah	Size	Area availability	SUM

Size (1 byte):	Number of characters in the area availability field (fixed at 1)
Area availability (1 byte):	Availability of data areas (fixed at 21h) 21h: Data area is available
SUM (1 byte):	Checksum

(12) Data Area Information Inquiry

In response to a data area information inquiry command sent from the host, this MCU returns the number of data area counts and their addresses.

Command	2Bh		
Response	3Bh	Size	Area count
	Area start address		
	Area end address		
	SUM		

Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of data area counts (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a data area
Area end address (4 bytes):	End address of a data area
SUM (1 byte):	Checksum

The information concerning the block configuration in the data area is included in the response to the block information inquiry command (see section 40.8.7, Inquiry/Selection Command Wait).

(13) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, this MCU checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, this MCU returns a response (06h) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, this MCU returns an error response (BFh). Upon reception of response 06h, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host's bit rate to the new one. After that, the host sends confirmation data (06h) in the new bit rate, and this MCU returns a response (06h) for the confirmation data.
Be sure to issue a new bit rate selection command only after a clock mode selection command.

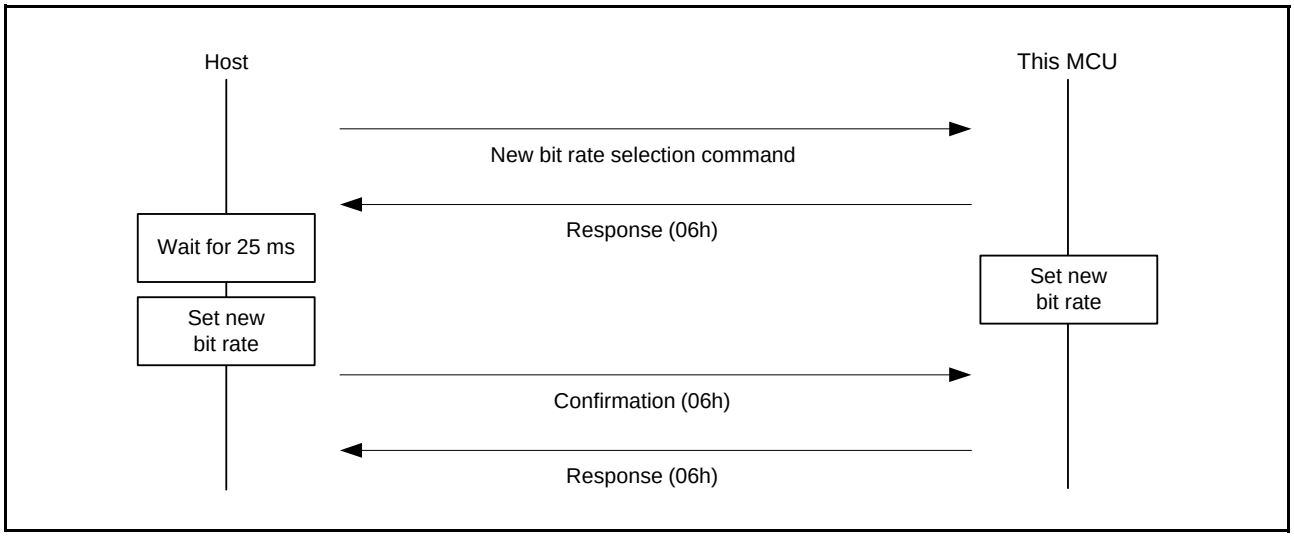


Figure 40.31 New Bit Rate Selection Sequence

Command	3Fh	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			
Response	06h			
Error response	BFh	Error		
Confirmation	06h			
Response	06h			

Size (1 byte):	Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields
Bit rate (2 bytes):	New bit rate (for example, 00C0h indicates 19200 bps) 1/100 of the new bit rate value should be specified.
Input frequency (2 bytes):	Frequency input to this MCU (for example, 04E2h indicates 12.50 MHz) This value should be calculated by multiplying the input frequency value to two decimal places by 100.
Clock type count (1 byte):	Number of clock types (fixed: 02h indicates two clock types; that is, a system clock and a peripheral clock)
Multiplication ratio 1 (1 byte):	Multiplication/division ratio of the input frequency to obtain the system clock (ICLK) A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4) A negative value indicates a division ratio (for example, FEh = –2 = divided by 2)
Multiplication ratio 2 (1 byte):	Multiplication/division ratio of the input frequency to obtain the peripheral clock (PCLKB) This value is represented in the same format as multiplication ratio 1
SUM (1 byte):	Checksum
Error:	Error code 11h: Checksum error 24h: Bit rate selection error 25h: Input frequency error 26h: Multiplication ratio error 27h: Operating frequency error

- Bit rate selection error

A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of this MCU within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (f_{EX}), multiplication ratio 2 ($M_{P\phi}$), the bit rate register (BRR) setting (N) in the SCI, and the CKS[1:0] bit value (n) in the serial mode register (SMR).

$$\text{Error (\%)} = \left\{ \frac{f_{EX} \times M_{P\phi} \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$$

- Input frequency error

An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

- Multiplication ratio error

A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

- Operating frequency error

An operating frequency error occurs when this MCU cannot operate at the operating frequencies selected through a new bit rate selection command. This MCU calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating frequency inquiry command.

(14) Programming/Erasure State Transition

In response to a programming/erasure state transition command sent from the host, this MCU determines whether ID code protection is enabled or disabled using the control code and ID code written in the ROM. When ID code protection is enabled, this MCU returns a response (16h) and waits for the ID code. When ID code protection is disabled, this MCU erases the entire area of each of the user area, user boot area and data area. After completing entire erasure, this MCU returns a response (26h) and waits for a programming/erasure command. If this MCU has failed to complete erasure due to an error, it returns an error response (C0h, and 51h).

Do not issue a programming/erasure state transition command before the device selection, clock mode selection, and new bit rate selection commands.

Command	40h	
Response	ACK	
Error response	C0h	51h

ACK (1 byte): ACK code
26h: ID code protection is disabled
16h: ID code protection is enabled

(15) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, this MCU returns its current status. The boot program status inquiry command can be issued in both the inquiry/selection command wait and programming/erasure command wait.

Command	4Fh			
Response	5Fh	Size	Status	Error

Size (1 byte): Total number of bytes in the status and error fields (fixed at 2)

Status (1 byte): Current status of this MCU (see Table 40.17)

Error (1 byte): Error status of this MCU (see Table 40.18)

Table 40.17 Status Code

Code	Description
11h	Waiting for device selection
12h	Waiting for clock mode selection
13h	Waiting for bit rate selection
1Fh	Waiting for transition to programming/erasure command wait (bit rate has been selected)
31h	Erasing the user area/user boot area
3Fh	Waiting for a programming/erasure command
4Fh	Waiting for reception of programming data
5Fh	Waiting for erasure block specification

Table 40.18 Error Code

Code	Description
00h	No error
11h	Checksum error
21h	Device code error
22h	Clock mode error
24h	Bit rate selection error
25h	Input frequency error
26h	Multiplication ratio error
27h	Operating frequency error
29h	Block number error
2Ah	Address error
2Bh	Data size error
51h	Erasure error
52h	Incomplete erasure error
53h	Programming error
54h	Selection error
80h	Command error
FFh	Bit rate adjustment confirmation error

40.8.8 ID Code Wait State

Table 40.19 shows the command available in the ID code wait state.

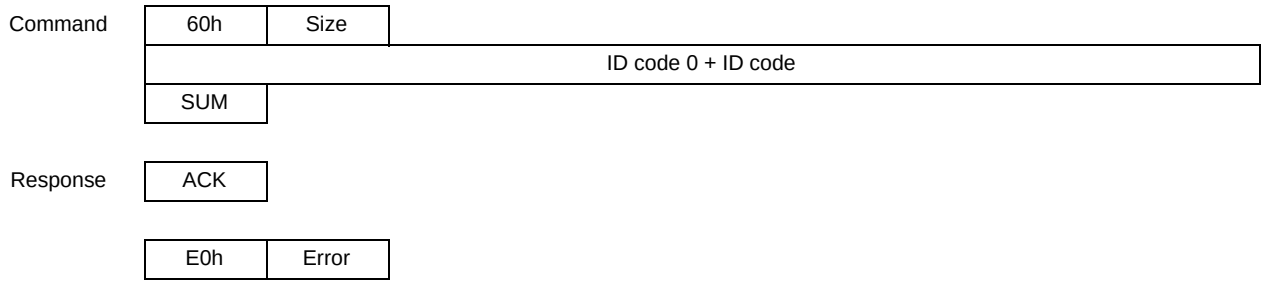
Table 40.19 ID Code Check Command

Command Name	Function
ID code check	Performs the ID code check

If the host has sent an undefined command, this MCU returns a response indicating a command error. For the contents of a command error, see section 40.8.7, Inquiry/Selection Command Wait.

(1) ID Code Check

In response to an ID code check command sent from the host, this MCU compares the code sent from the host with the control code and ID code in the ROM and returns the result.



Size (1 byte): Number of bytes in the ID code field (fixed at 16)

ID code (16 bytes): ID code 0 (1 byte) + ID code (15 bytes)

SUM (1 byte): Checksum

ACK (1 byte): ACK code
26h: Returns the response for a programming/erasure state transition command

Error (1 byte): Error code
11h: Checksum error
61h: ID code mismatch
63h: ID code mismatch (erasure error)
An error has occurred during erasure triggered by an ID code mismatch.

40.8.9 Programming/Erasure Command Wait

Table 40.20 lists the commands available in the programming/erasure command wait.

Table 40.20 Programming/Erasure Commands

Command Name	Function
User boot area programming selection	Selects the program for user boot area programming
User/data area programming selection	Selects the program for user/data area programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erase	Erases block data
Memory read	Reads data from memory
User boot area checksum	Performs checksum verification for the user boot area
User area checksum	Performs checksum verification for the user area
Data area checksum	Performs checksum verification for the data area
User boot area blank check	Checks whether the user boot area is blank
User area blank check	Checks whether the user area is blank
Data area blank check	Checks whether the data area is blank
Read lock bit status	Reads from the lock bit
Lock bit program	Writes to the lock bit
Lock bit enable	Enables the lock bit protection
Lock bit disable	Disables the lock bit protection
Boot program status inquiry	Inquires regarding the state of the this MCU

If the host has sent an undefined command, this MCU returns a response indicating a command error. For the contents of a command error, see section 40.8.7, Inquiry/Selection Command Wait.
To program the ROM, issue a programming selection command (user area/data programming selection, user boot area programming selection) and then a 256-byte programming command from the host. To program the DataFlash, issue a user/data area programming selection command and then a 256-byte programming command specifying a data area address as the programming address. Upon reception of a programming selection command, this MCU enters the programming data wait state (see section 40.8.2, State Transitions in Boot Mode). In response to a 256-byte programming command sent from the host in this state, this MCU starts programming the ROM/E2 DataFlash. When the host sends a 256-byte programming command specifying FFFF FFFFh as the programming start address, this MCU detects it as the end of programming and enters the programming/erasure command wait.
To erase the ROM/E2 DataFlash, issue an erasure selection command and then a block erase command from the host. Upon reception of an erasure selection command, this MCU enters the erasure block selection wait state (see section 40.8.2, State Transitions in Boot Mode). In response to a block erase command sent from the host in this state, this MCU erases the specified block in the ROM/E2 DataFlash. When the host sends a block erase command specifying FFh as the block number, this MCU detects it as the end of erasure and enters the programming/erasure command wait.
To read data from the data area, select the user area through a memory read command specifying a data area address as the read address.

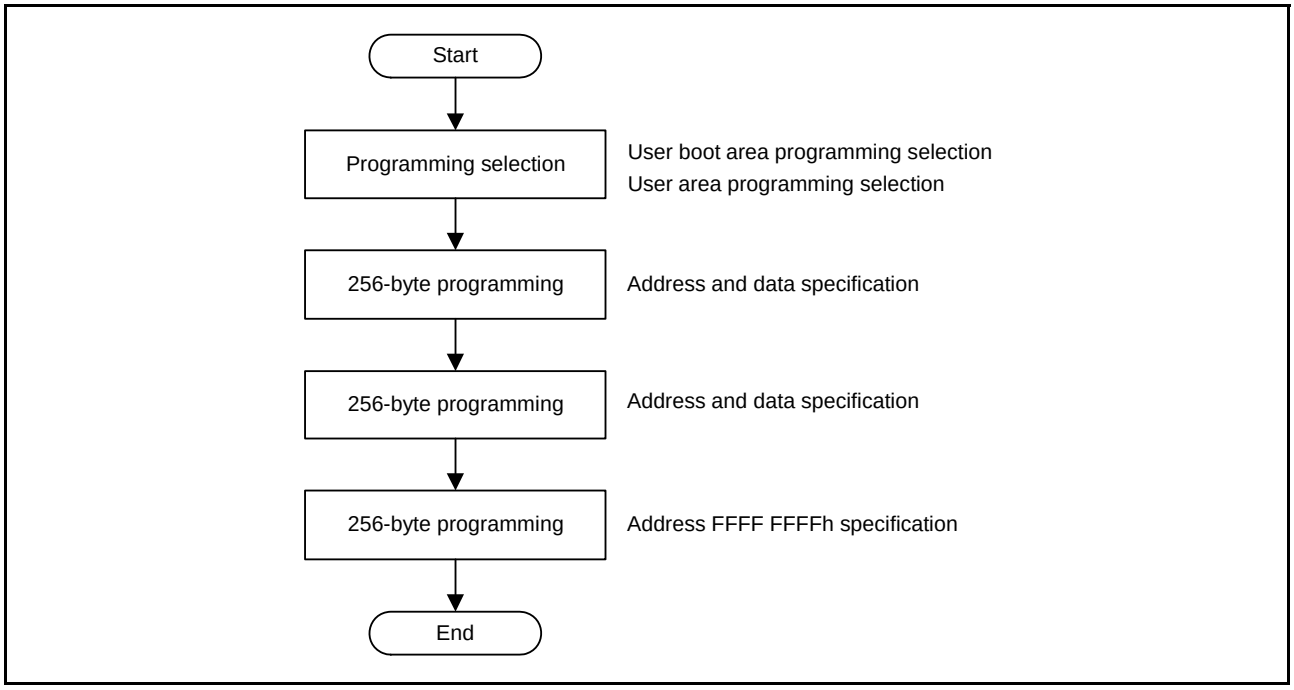


Figure 40.32 Procedure for ROM/E2 DataFlash Programming in Boot Mode

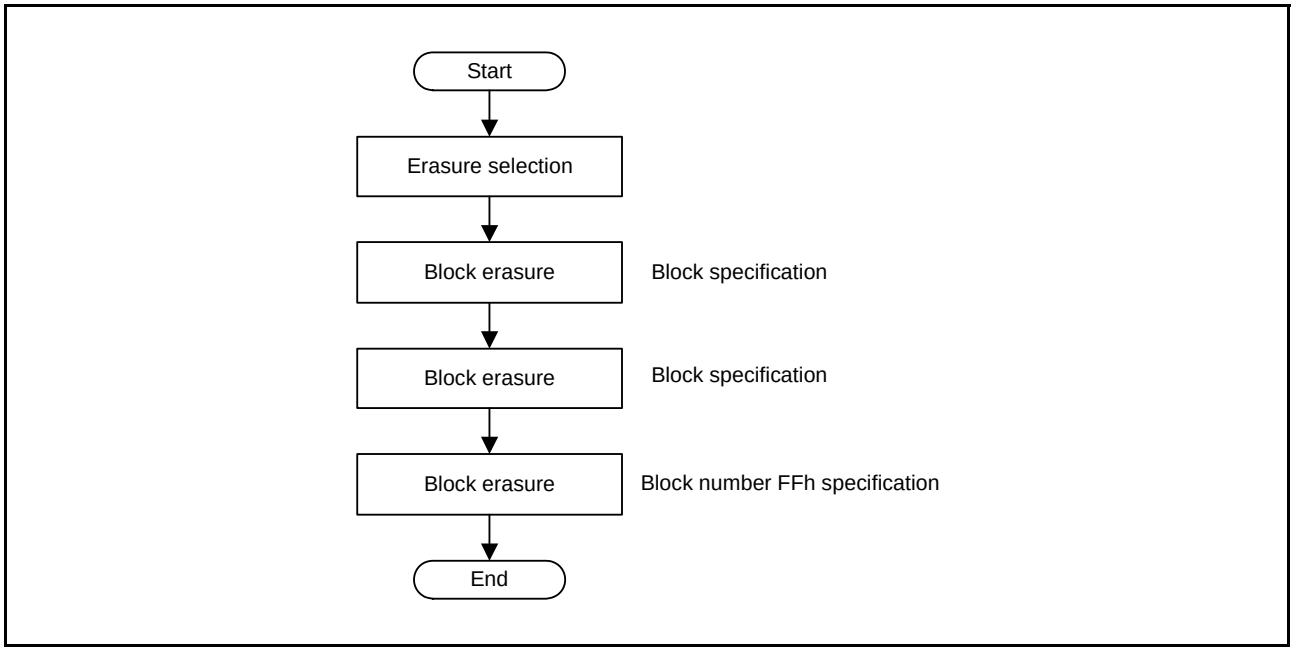


Figure 40.33 Procedure for ROM/E2 DataFlash Erasure in Boot Mode

Each command is described in detail below. The “command” in the description indicates a command sent from the host to this MCU and the “response” indicates a response sent from this MCU to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by this MCU becomes 00h.

(1) User Boot Area Programming Selection

In response to a user boot area programming selection command sent from the host, this MCU selects the program for user boot area programming and waits for programming data.

Command	42h
Response	06h

(2) User/Data Area Programming Selection

In response to a user/data area programming selection command sent from the host, this MCU selects the program for user area programming and waits for programming data. To program the E2 DataFlash, issue a user/data area programming selection command.

Command	43h
Response	06h

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, this MCU programs the ROM/E2 DataFlash. After completing ROM/E2 DataFlash programming successfully, this MCU returns a response (06h). If an error has occurred during ROM programming, this MCU returns an error response (D0h).

Command	50h	Programming address		
	Data	Data	. . .	Data
	SUM			
Response	06h			
Error response	D0h	Error		

Programming address (4 bytes):	Target address of programming To execute programming, a 256-byte boundary address should be specified. To terminate programming, FFFF FFFFh should be specified.
Data (256 bytes):	Programming data FFh should be specified for the bytes that do not need to be programmed. When terminating programming, no data needs to be sent (only the programming address and SUM should be sent in that order).
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area) 53h: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, this MCU selects the erasure program and waits for erasure block specification.

Command	48h
Response	06h

(5) Block Erasure

In response to a block erase command sent from the host, this MCU erases the ROM/E2 DataFlash. When erasing the user boot area, 80h should be specified as the block number. After completing ROM/E2 DataFlash erasure successfully, this MCU returns a response (06h). If an error has occurred during ROM/E2 DataFlash erasure, this MCU returns an error response (D8h).

Command	58h	Size	Block	SUM
Response	06h			
Error response	D8h	Error		

Size (1 byte):	Number of bytes in the block specification field (fixed at 1)
Block (1 byte):	Block number whose data is to be erased To specify user boot area, 80h should be specified. To terminate erasure, FFh should be specified.
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 29h: Block number error (an incorrect block number is specified) 51h: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, this MCU reads data from the ROM/E2 DataFlash. After completing ROM/E2 DataFlash reading successfully, this MCU returns the data stored in the address specified by the memory read command. If this MCU has failed to read the ROM/E2 DataFlash, this MCU returns an error response (D2h).

Command	52h	Size	Area	Read start address
	Reading size			SUM

Response	52h	Reading size		
	Data	Data	. . .	Data
	SUM			

Error response	D2h	Error

Size (1 byte):	Total number of bytes in the area, read start address, and reading size fields
Area (1 byte):	Target area to be read 00h: User boot area 01h: User area, data area
Read start address (4 bytes):	Start address of the area to be read
Reading size (4 bytes):	Size of data to be read (bytes)
SUM (1 byte):	Checksum
Data (1 byte):	Data read from the ROM/E2 DataFlash
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error • The value specified for area selection is neither 00h nor 01h. • The specified read start address is outside the selected area. 2Bh: Data size error • 00h is specified for the reading size. • The reading size is larger than the area. • The end address calculated from the read start address and the reading size is outside the selected area.

(7) User Boot Area Checksum

In response to a user boot area checksum command sent from the host, this MCU sums the user boot area data in byte units and returns the result (checksum).

Command	4Ah			
Response	5Ah	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the user boot area data
SUM (1 byte):	Checksum (for the response data)

(8) User Area Checksum

In response to a user area checksum command sent from the host, this MCU sums the user area data in byte units and returns the result (checksum).

Command	4Bh			
Response	5Bh	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the user area data The user area also stores the key code for debugging function authentication. Note that the checksum includes this key code value.
SUM (1 byte):	Checksum (for the response data)

(9) Data Area Checksum

In response to a data area checksum command sent from the host, this MCU sums the data area data in byte units and returns the result (checksum).

Command	61h			
Response	71h	Size	Area checksum	SUM

Size (1 byte):	Number of bytes in the area checksum field (fixed at 4)
Area checksum (4 bytes):	Checksum of the data area
SUM (1 byte):	Checksum (for the response data)

(10) User Boot Area Blank Check

In response to a user boot area blank check command sent from the host, this MCU checks whether the user boot area is completely erased. When the user boot area is completely erased, this MCU returns a response (06h). If the user boot area has an unerased area, this MCU returns an error response (CCh, 52h).

Command	4Ch	
Response	06h	
Error response	CCh	52h

(11) User Area Blank Check

In response to a user area blank check command sent from the host, this MCU checks whether the user area is completely erased. When the user area is completely erased, this MCU returns a response (06h). If the user area has an unerased area, this MCU returns an error response (CDh, 52h).

Command	4Dh	
Response	06h	
Error response	CDh	52h

(12) Data Area Blank Check

In response to a data area blank check command sent from the host, this MCU checks whether the data area is completely erased. When the data area is completely erased, this MCU returns a response (06h). If the data area has an unerased area, this MCU returns an error response (E2h, 52h).

Command	62h	
Response	06h	
Error response	E2h	52h

(13) Read Lock Bit Status

In response to a read lock bit status command sent from the host, this MCU reads data from the lock bit. After completing the lock bit reading successfully, this MCU returns the data stored in the address specified by the read lock bit status command. If this MCU has failed to read the lock bit, this MCU returns an error response (F1h).

Command	71h	Size	Area	A15 to A8	A23 to A16	A31 to A24	SUM

A15 to A8 (1 byte): The last address in the specified block (bits 15 to 8)
A23 to A16 (1 byte): The last address in the specified block (bits 23 to 16)
A31 to A24 (1 byte): The last address in the specified block (bits 31 to 24)

Response	Status	
Error response	F1h	Error

Size (1 byte): Total number of bytes in the area, A15 to A8, A23 to A16, and A31 to A24 (fixed at 4 in this MCU)

Area (1 byte): Target area to be read
01h: User area

A15 to A8 (1 byte): A15 to A8 of the last address in the specified block (bits 8 to 15)
A23 to A16 (1 byte): A23 to A16 of the last address in the specified block (bits 16 to 23)
A31 to A24 (1 byte): A31 to A24 of the last address in the specified block (bits 24 to 31)

SUM (1 byte): Checksum
Status (1 byte): Bit 6 locked at 0
Bit 6 unlocked at 1

Error (1 byte): Error code
11h: Checksum error
2Ah: Address error (the specified address is not in the target area)

(14) Lock Bit Program

In response to a lock bit program command sent from the host, this MCU writes to a lock bit and locks the specified block. After completing the lock bit blocking successfully, this MCU returns a response (06h). If this MCU has failed to lock, this MCU returns an error response (F7h).

Command	77h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
Response	06h						
Error response	F7h	Error					

Size (1 byte):	Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in this MCU)
Area (1 byte):	Target area to be locked 01h: User area
Third highest order address (1 byte):	Third highest order address at the specified block's end address (8 to 15 bits)
Second highest order address (1 byte):	Second highest order address at the specified block's end address (16 to 23 bits)
Highest order address (1 byte):	Highest order address at the specified block's end address (24 to 31 bits)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area) 53h: Lock cannot be done due to a programming error

(15) Lock Bit Enable

In response to a lock bit enable command sent from the host, this MCU enables a lock bit.

Command	7Ah
Response	06h

(16) Lock Bit Disable

In response to a lock bit disable command sent from the host, this MCU disables a lock bit.

Command	75h
Response	06h

(17) Boot Program Status Inquiry

For details, refer to section 40.8.7, Inquiry/Selection Command Wait.

40.9 ID Code Protection on Connection of the On-Chip Debugger

This function is used to prohibit connection with the on-chip debugger. When connecting an on-chip debugger, the control code and ID code that have been written to the ROM are used to determine whether ID code protection on connection of the on-chip debugger is enabled or disabled and to judge ID code protection on connection of the on-chip debugger. When the ID code protection is enabled, the code sent from the on-chip debugger is compared with the control code and ID code in the ROM to determine whether they match. If they match, connection with the on-chip debugger is allowed. If they do not match, the on-chip debugger cannot be connected. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, and 74h (from the first to the seventh field of the ID code), this is considered to be a non-match, judgment of the ID code does not proceed, and connection of the on-chip debugger is prohibited. Furthermore, if all bytes of the control code and ID code have the value FFh, there is no determination of matching, the ID code is considered to match, and connection of the on-chip debugger is allowed. See Figure 40.29 for the configuration of ID codes in flash memory.

Table 40.21 Specifications for ID Code Protection on Connection of the On-Chip Debugger

Control Code	ID Code	State of Protection	Operations at On-Chip Debugger Connection
FFh	FFh, …, FFh (all bytes FFh)	Protection disabled	The ID code matches, and connection to the on-chip debugger is permitted.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, …, FFh	Protection enabled	The ID code is non-matching, and connection to the on-chip debugger is prohibited.
Other than above	Other than above	Protection enabled	Matching ID code: Authentication of the on-chip debugger is ended and connection with the on-chip debugger is permitted. Non-matching ID code: The ID code protection waiting state is entered again.

40.10 ROM Code Protection

ROM code protection is a facility for prohibiting a PROM programmer from reading from or programming flash memory. The ROM code in flash memory is a 32-bit code. Figure 40.34 shows the configuration of the ROM code. Set the ROM code in 32-bit units.
For release from ROM code protection, erase the EB00 block of the user area that contains the ROM code in boot mode or by user programming.

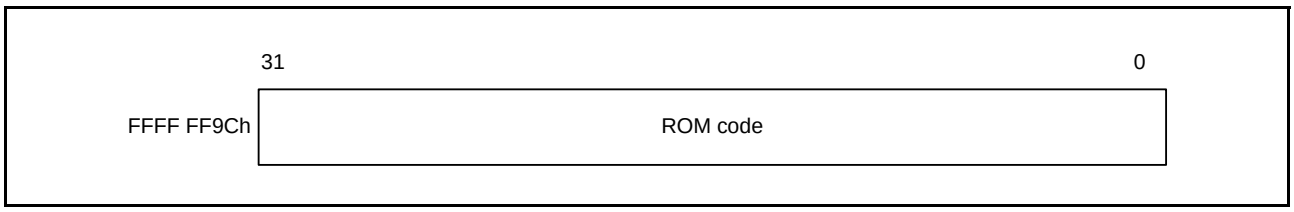


Figure 40.34 Configuration of ROM Code

Table 40.22 Specifications for ROM Code Protection

ROM Code	State of Protection	Operations at the Time of Connection with the PROM Programmer
0000 0000h	Protection enabled (ROM code protection 1)	Access (both reading and writing) to the user area and the user boot area are prohibited.
0000 0001h	Protection enabled (ROM code protection 2)	Reading from the user area and the user boot area are prohibited.
Other than above	Protection disabled	Access (both reading and writing) to the user area and the user boot area are permitted.

40.11 Usage Notes (Common to the ROM/E2 DataFlash Memory)

(1) Areas where Programming or Erasure is Suspended

Data in areas where programming or erasure is suspended are undefined. To avoid malfunctions due to the reading of undefined data, prevent the reading of data and execution of code from areas where programming or erasure is currently suspended.

(2) Suspending Programming or Erasure

If you use the programming/erasure suspension command to suspend the processing of programming or erasure, be sure to use the resume command so that the processing is completed.

(3) Prohibition of Reprogramming

Two or more programming operations cannot be performed for the same address range. If an address range that has already been programmed is to be programmed again, be sure to erase the area in advance of the programming.

(4) Reset during Programming, Erasure, or Blank Check

In case of the generation of a reset by the signal on the RES# pin during programming and erasure or blank checking, only de-assert the reset signal after it has remained within the range of operating voltage stipulated in the electrical characteristics over the period tRESWF (see section 41, Electrical Characteristics).
In case of resetting of the FCU by the FRESETR.FRESET bit during programming and erasure or blank checking, maintain the reset state over the period tFCUR (see section 41, Electrical Characteristics).
During resetting of the FCU, do not read ROM that is the target of programming and erasure or blank checking.
Resets by the watchdog timer and watchdog timer and software resets during programming and erasure or blank checking can be used without securing the above periods.

(5) Prohibition of Non-Maskable Interrupts during Programming or Erasure

If a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, WDT underflow, refresh error, IWDT underflow/refresh error, voltage-monitoring 1 interrupt or voltage-monitoring 2 interrupt) occurs during programming or erasure, as this will lead to fetching of the vector from the ROM, and the data read out will be undefined. Therefore, avoid a non-maskable interrupt being generated during programming or erasing in the ROM (this prohibition only applies to the ROM).

(6) Interrupt Vector Assignment During Programming, Erasure, or Blank Check

The generation of interrupts during programming, erasure, or blank check may lead to the fetching of vectors from the ROM. To prevent access to the ROM area due to the generation of interrupts, set the interrupt table register (INTB) of the CPU so that the destination for the fetching of interrupt vector is an area outside the ROM.

(7) Programming and Erasure in Low-Speed Operating Modes 1 and 2

Do not program or erase the flash memory after low-speed operating mode 1 or 2 has been selected in the operating voltage control register (OPCCR).

(8) Abnormal End during Programming, Erasure or Blank Check

In cases where programming and erasure or blank checking is not completed due to the operating voltage rising above the range stipulated in the electrical characteristics, a reset, the FCU being reset by using the FRESETR.FRESET bit, error detection leading to the flash memory being placed in the command-locked state, or one of the prohibitions under item (9), the lock bit may become 0 (indicating the protected state).
In such cases, erase the lock bit by issuing a block-erase command while the FPROTR.FPROTCN is 1. Redo programming that was not completed normally after that.

(9) Actions Prohibited during Programming, Erasure, or Blank Check

During programming, erasure, or blank check, high voltage is applied to the flash memory. In order to prevent damaging the flash memory, the following must be observed.

- Switch the clock source to the LOCO by setting the ELSR28 register and do not generate this event.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not enable clock source switching at returning from sleep mode by setting the RSTCKCR.RSTCKEN bit.
- Do not set a frequency that is different from the FlashIF clock (FCLK) in the PCKAR register.
- Do not allow the input voltage to be below the operating voltage of this MCU.
- Do not change the value of the FWEPROR.FLWE[1:0] bits.
- Do not change the operating mode by the setting of the SYSCR0.ROME bit.
- Do not change the clock-source by setting of the SCKCR3.CKSEL[2:0] bits.
- Do not change the frequency dividing ratio for the FlashIF clock (FCLK).
- Do not make transition to all-module clock-stop mode, software standby mode, or deep software standby mode.

(10) Notes on Flash Programming in Boot Mode

The main clock must be being input for programming of the flash memory in boot mode. In the case of boot mode, a crystal oscillator within the range indicated in section 41.3.1, Clock Timing, must be connected between the XTAL and EXTAL pins.

(11) Handling of the EXTAL Pin in Boot Mode

When operation is in boot mode, provide a clock signal through connection of an external input or crystal oscillator to the EXTAL pin.

40.12 Usage Notes (for E2 DataFlash)

(1) Protection of Data Area Immediately after a Reset

As the initial values of DFLREy and DFLWEy (y = 0, 1) are 0000h, programming, erasure, and reading of the data area are disabled after a reset. To read data from the data area, set DFLREy appropriately before accessing the data area. To program or erase the data area, set DFLWEy appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data area without setting the registers, the FCU detects the error and the FASTAT.CMDLK bit becomes 1 (the command-locked state).

(2) Reading in Low-Speed Operating Mode 2

The E2 DataFlash memory is not readable in low-speed operating mode 2. Place the chip in another mode if reading is to proceed.

41. Electrical Characteristics

41.1 Absolute Maximum Ratings

Table 41.1 Absolute Maximum Ratings
Conditions: VSS = AVSS0 = VREFL = VREFL0 = 0 V

Item	Symbol	Value	Unit
Power supply voltage	VCC	–0.3 to +6.5	V
Analog power supply voltage	AVCC0*1	–0.3 to +6.5	V
Reference power supply voltage	VREFH0*1	–0.3 to AVCC0 + 0.3	V
	VREFH*1	–0.3 to +6.5	V
Input voltage (except for port 4, ports 03, 05, and 07)	V_{in}	–0.3 to VCC + 0.3	V
Input voltage (port 4, port 07)	V_{in}	–0.3 to AVCC0 + 0.3	V
Input voltage (ports 03, 05)	V_{in}	–0.3 to VREFH + 0.3	V
Analog input voltage (ports 4, E)	V_{AN}	–0.3 to AVCC0 + 0.3	V
Operating temperature	T_{opr}	–40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.
To preclude any malfunctions due to noise interferences, insert capacitors of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, and between the VREFH0 and VREFL0 pins. Place capacitors of 0.1 µF or so as close to every power pin and use the shortest and heaviest possible traces.

Note 1. When neither the A/D converter nor the D/A converter is in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.

41.2 DC Characteristics

Table 41.2 DC Characteristics (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus)	V_{IH}	VCC × 0.7	—	VCC + 0.3	V	
	Port 4, port 07		AVCC0 × 0.8	—	AVCC0 + 0.3		
	Ports 03, 05		VREFH × 0.8	—	VREFH + 0.3		
	Port L5		VCC × 0.8	—	3.9		Conditions 1
	Except for RIIC input pin, port 4, ports 03, 05, 07, port L5		VCC × 0.8	—	VCC + 0.3		
	RIIC input pin (except for SMBus)	V_{IL}	–0.3	—	VCC × 0.3		
	Port 4, port 07		–0.3	—	AVCC0 × 0.2		
	Port 03, 05		–0.3	—	VREFH × 0.2		
	Port L5		–0.3	—	VCC × 0.2		Conditions 1
	Except for RIIC input pin, port 4, ports 03, 05, 07, port L5		–0.3	—	VCC × 0.2		
	RIIC input pin (except for SMBus)	ΔV_T	VCC × 0.05	—	—		
	Port 4, port 07		AVCC0 × 0.06	—	—		
	Ports 03, 05		VREFH × 0.06	—	—		
	Port L5		VCC × 0.06	—	—		Conditions 1
	Except for RIIC input pin, port 4, ports 03, 05, 07, port L5		VCC × 0.06	—	—		
Input level voltage (except for schmitt trigger input pins)	MD, EMLE	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL, WAIT#, TCK, RSPI input pin		VCC × 0.8	—	VCC + 0.3		
	D0 to D15		VCC × 0.7	—	VCC + 0.3		
	RIIC input pin (SMBus)		2.1	—	VCC + 0.3		Conditions 1
	CEC input pin		2.0	—	3.9		Conditions 1
	MD, EMLE	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL, WAIT#, TCK, RSPI input pin		–0.3	—	VCC × 0.2		
	D0 to D15		–0.3	—	VCC × 0.3		
	RIIC input pin (SMBus)		–0.3	—	0.8		Conditions 1
	CEC input pin		–0.3	—	0.8		Conditions 1
	CEC input pin	ΔV_T	—	0.3	—		Conditions 1

Table 41.3 DC Characteristics (2)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Test Conditions
Input leakage current	RES#, MD input pin, P35/NMI, EXTAL, port 4	$\|I_{in}\|$	—	—	1.0	μA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	Ports 12, 13, 16, 17, 20, 21, C0, C1	$\|I_{TSI}\|$	—	—	5.0	μA	V_{in} = 0 V, VCC
	Except for ports 12, 13, 16, 17, 20, 21, C0, C1		—	—	1.0		
	Port L5		—	—	1.8		VCC = 0 V, V_{in} = 3.9 V
Input capacitance	All input pins (except for ports 12, 13, 16, 17, 20, 21, C0, C1)	C_{in}	—	—	15	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	Ports 12, 13, 16, 17, 20, 21, C0, C1		—	—	30		

Table 41.4 DC Characteristics (3)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	VCC				Unit	Test Conditions
			2.7 to 3.6V		4.0 to 5.5V			
			Min.	Max.	Min.	Max.		
Input pull-up MOS current	All ports (except for ports 03, 05, ports 35 to P37, port 4, port L5)	I_p	–200	–10	–400	–50	μA	V_{in} = 0 V

Table 41.5 DC Characteristics (4)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item					Symbol	Typ.	Max.	Unit	Test Conditions
Supply current*1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 54 MHz	I_{CC}	20	—	mA	ICLK = 54 MHz PCLKB = 27 MHz PCLKD = 54 MHz FCLK = 27 MHz BCLK = 54 MHz
			All peripheral operation: Normal*3	ICLK = 54 MHz		24	—		
			All peripheral operation: Max.*3	ICLK = 54 MHz		—	55		
		Sleep mode	No peripheral operation	ICLK = 54 MHz		15.5	—		
			All peripheral operation: Normal	ICLK = 54 MHz		19.5	—		
		All-module clock stop mode				14	—		
		Increase during BGO operation*4				12	—		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation.
Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation.
Note 4. This is the increase if data is programmed to or erasing from the ROM or E2 DataFlash during program execution.

Table 41.6 DC Characteristics (5)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item					Symbol	Typ.	Max.	Unit	Test Conditions
Supply current*1	Low-speed operating mode 1	Normal operating mode	No peripheral operation*2	ICLK = 1 MHz	I_{CC}	4	—	mA	
			All peripheral operation: Normal*3	ICLK = 1 MHz		4.2	—		
			All peripheral operation: Max.*3	ICLK = 1 MHz		—	15		
		Sleep mode	No peripheral operation	ICLK = 1 MHz		3.8	—		
			All peripheral operation: Normal	ICLK = 1 MHz		4.0	—		
		All-module clock stop mode				3.7	—		
	Low-speed operating mode 2	Normal operating mode	No peripheral operation*4	ICLK = 125 kHz		0.4	—		
			All peripheral operation: Normal*5	ICLK = 125 kHz		0.5	—		
			All peripheral operation: Max.*5	ICLK = 125 kHz		—	8*6		
		Sleep mode	No peripheral operation	ICLK = 125 kHz		0.3	—		
			All peripheral operation: Normal	ICLK = 125 kHz		0.4	—		
		All-module clock stop mode				0.28	—		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is the main clock.
Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is the main clock.
Note 4. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is LOCO.
Note 5. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is LOCO.
Note 6. Value when the main clock continues oscillating at 13.5 MHz.

Table 41.7 DC Characteristics (6)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item				Symbol	Typ.	Max.	Unit	Test Conditions
Supply power*1	Software standby mode			I_{CC}	40	1000	µA	
	Deep software standby mode	RAM power supplied			22	200		
		RAM power not supplied	Power-on reset circuit low power consumption function disabled		21	60		
			Power-on reset circuit low power consumption function enabled		6.2	28		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.

Table 41.8 DC Characteristics (7)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item	Symbol	Typ.	Max.	Unit	Test Conditions
Permissible total consumption power*1	Pd	—	360	mW	

Note 1. Total power dissipated by the entire chip (including output currents)

Table 41.9 DC Characteristics (8)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion	I_{AVCC0}	—	1.9	4.2	mA	Conditions 1
			—	2.5	4.2	mA	Conditions 2
	Waiting for A/D conversion		—	0.1	4	μA	
	During D/A conversion (per channel)	I_{VREFH}*1	—	0.3	1	mA	Conditions 1
			—	0.46	1	mA	Conditions 2
	Waiting for A/D, D/A conversion (all units)*2	—	—	23	40	μA	
Reference power supply current	During A/D conversion	I_{VREFH0}	—	0.44	1.5	mA	Conditions 1
			—	0.66	1.5	mA	Conditions 2
	Waiting for A/D conversion		—	0.1	1	μA	

Note: The values for A/D conversion apply when the sample and hold circuit is not in use.
Note 1. The reference power supply current is included in the power supply current value for D/A conversion.
Note 2. The value is the total value of I_{AVCC0} and I_{VREFH}.

Table 41.10 DC Characteristics (9)

Conditions: VCC = AVCC0 = 0 to 5.5 V, VREFH = 0 to AVCC0, VREFH0 = 0 to AVCC0,
VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
VCC rising gradient	SrVCC	—	—	20	ms/V	At cold start

Table 41.11 Permissible Output Currents

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item			Symbol	Max.	Unit
Permissible output low current (average value per 1 pin)		Normal output mode	I_{OL}	2.0	mA
Permissible output low current (maximum value per 1 pin)		Normal output mode		4.0	mA
Permissible output low current (total)	Total of all output pins		ΣI_{OL}	80	mA
Permissible output high current (average value per 1 pin)		Normal output mode	I_{OH}	–2.0	mA
Permissible output high current (maximum value per 1 pin)		Normal output mode		–4.0	mA
Permissible output high current (total)	Total of all output pins		ΣI_{OH}	–80	mA

Table 41.12 Output Values of Voltage (1)

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output pins (other than RIIC)	Normal output	V_{OL}	—	0.5	V	I_{OL} = 1.0 mA
		High-drive output		—	0.5		I_{OL} = 2.0 mA
	RIIC pins			—	0.4		I_{OL} = 3.0 mA
				—	0.6		I_{OL} = 6.0 mA
	CEC pins			—	0.6		I_{OL} = 2.1 mA
Output high	All output pins	Normal output	V_{OH}	VCC – 0.5	—	V	I_{OH} = –1.0 mA
		High-drive output		VCC – 0.5	—		I_{OH} = –2.0 mA

Table 41.13 Output Values of Voltage (2)

Conditions: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output pins (other than RIIC)	Normal output	V_{OL}	—	0.8	V	I_{OL} = 2.0 mA
		High-drive output		—	0.8		I_{OL} = 4.0 mA
	RIIC pins			—	0.4		I_{OL} = 3.0 mA
				—	0.6		I_{OL} = 6.0 mA
Output high	All output pins	Normal output	V_{OH}	VCC – 0.8	—	V	I_{OH} = –2.0 mA
		High-drive output		VCC – 0.8	—		I_{OH} = –4.0 mA

41.3 AC Characteristics

Table 41.14 Operation Frequency Value (High-Speed Operating Mode)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)	f_{max}	—	—	54	MHz
	FlashIF clock (FCLK)*1		—	—	32	
	Peripheral module clock (PCLKB)		—	—	32	
	Peripheral module clock (PCLKD)*2		—	—	54	
	External bus clock (BCLK)		—	—	54	
	BCLK pin output		—	—	27	

Note 1. The lower-limit frequency of FCLK is 4 MHz during programming or erasing of the flash memory.
Note 2. The lower-limit frequency of PCLKD is 1 MHz when the A/D converter is in use.

Table 41.15 Operation Frequency Value (Low-Speed Operating Mode 1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)	f_{max}	—	—	1	MHz
	FlashIF clock (FCLK)*1		—	—	1	
	Peripheral module clock (PCLKB)		—	—	1	
	Peripheral module clock (PCLKD)*2		—	—	1	
	External bus clock (BCLK)		—	—	1	
	BCLK pin output		—	—	1	

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The lower-limit frequency of PCLKD is 1 MHz when the A/D converter is in use.

Table 41.16 Operation Frequency Value (Low-Speed Operating Mode 2)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)	f_{max}	—	—	143.75	kHz
	FlashIF clock (FCLK)*1		—	—	143.75	
	Peripheral module clock (PCLKB)		—	—	143.75	
	Peripheral module clock (PCLKD)*2		—	—	143.75	
	External bus clock (BCLK)		—	—	143.75	
	BCLK pin output		—	—	143.75	

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The A/D converter cannot be used.

41.3.1 Clock Timing

Table 41.17 BCLK Timing

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
BCLK pin output cycle time	t_{Bcyc}	37	—	—	ns	Figure 41.1
BCLK pin output high pulse width*1	t_{CH}	5	—	—	ns	
BCLK pin output low pulse width*1	t_{CL}	5	—	—	ns	
BCLK pin output rising time	t_{Cr}	—	—	5	ns	
BCLK pin output falling time	t_{Cf}	—	—	5	ns	

Note 1. When the EXTAL external clock input is used with divided by 1 to output from the BCLK pin, the above should be satisfied with a duty cycle of 45 to 55%.

Table 41.18 Clock Timing

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time		t_{EXcyc}	50 (62.5)*1	—	—	ns	Figure 41.2
EXTAL external clock input high pulse width		t_{EXH}	20 (25)*1	—	—	ns	
EXTAL external clock input low pulse width		t_{EXL}	20 (25)*1	—	—	ns	
EXTAL external clock rising time		t_{EXr}	—	—	5	ns	
EXTAL external clock falling time		t_{EXf}	—	—	5	ns	
EXTAL external clock input wait time*2		t_{EXWT}	1	—	—	ms	
Main clock oscillator oscillation frequency*3		f_{MAIN}	8	—	20 (16)*1	MHz	
Main clock oscillation stabilization time (crystal)		$f_{MAINOSC}$	—	—	*3	ms	Figure 41.3
Main clock oscillation stabilization wait time (crystal)		$f_{MAINOSCWT}$	—	—	*4	ms	
LOCO clock cycle time		$t_{LOCOCYC}$	6.96	8	9.4	µs	
LOCO clock cycle time		$t_{LOCOCYC}$	7.27	8	8.89	µs	T_a = 0 to +60°C
LOCO clock oscillation frequency		f_{LOCO}	106.25	125	143.75	kHz	
LOCO clock oscillation frequency		f_{LOCO}	112.5	125	137.5	kHz	T_a = 0 to +60°C
LOCO clock oscillation stabilization wait time		t_{LOCOWT}	—	—	20	µs	Figure 41.4
PLL input frequency		f_{PLLIN}	4	—	20	MHz	
PLL circuit oscillation frequency		t_{LOCOWT}	104	—	200	MHz	
PLL clock oscillation stabilization time	PLL operation started after main clock oscillation has settled	t_{PLL1}	—	—	500	µs	Figure 41.5
PLL clock oscillation stabilization wait time		t_{PLLWT1}	—	—	*5	ms	
PLL clock oscillation stabilization time	PLL operation started before main clock oscillation has settled	t_{PLL2}	—	—	$t_{MAINOSC}$ + t_{PLL1}	ms	Figure 41.6
PLL clock oscillation stabilization wait time		t_{PLLWT2}	—	—	*5	ms	

Note 1. The values in parentheses indicate when the MONFCR register is set to a value other than A5h (noise filter enabled) while CECMCLK is selected as the CEC operating clock and RCRMCLK is selected as the RCR operating clock.
Note 2. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating).
Note 3. When using a main clock, request the oscillator manufacturer to evaluate the oscillator. For the oscillation stabilization time, refer to the evaluation results obtained from the oscillator manufacturer.
Note 4. The number of cycles n selected by the value of the MOSCWTCR.MSTS[4:0] bits determines the main-clock oscillation stabilization waiting time in accord with the formula below.

$$t_{MAINOSCWT} = t_{MAINOSC} + \frac{n + 16384}{f_{MAIN}}$$

Note 5. The number of cycles n selected by the value of the PLLWTCR.PSTS[4:0] bits determines the PLL-clock oscillation stabilization waiting time in accord with the formula below.

$$t_{PLLWT1} = t_{PLL1} + \frac{n + 131072}{f_{PLL}}$$

$$t_{PLLWT2} = t_{PLL2} + \frac{n + 131072}{f_{PLL}} = t_{MAINOSC} + t_{PLL1} + \frac{n + 131072}{f_{PLL}}$$

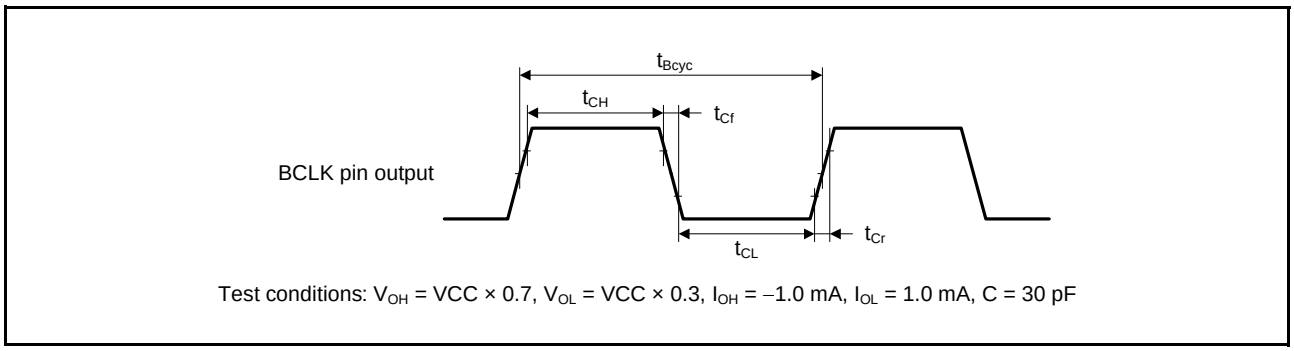


Figure 41.1 BCLK Pin Output Timing

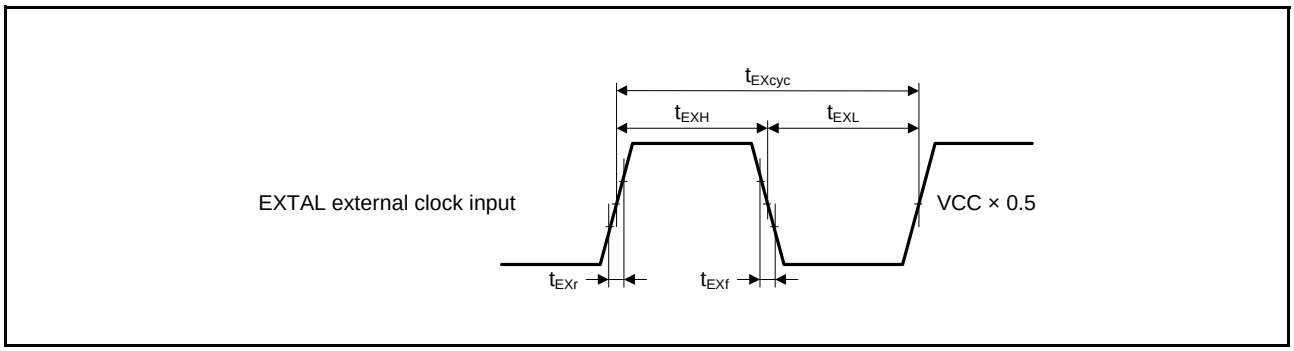


Figure 41.2 EXTAL External Clock Input Timing

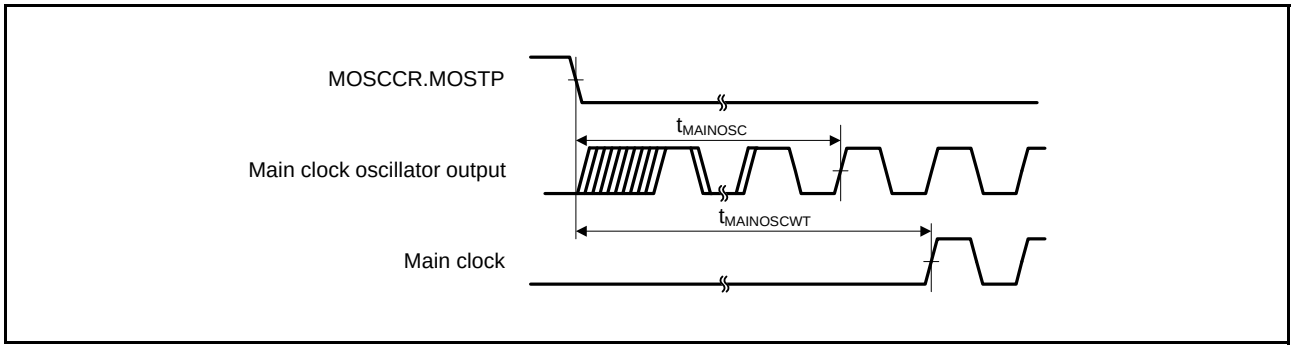


Figure 41.3 Main Clock Oscillation Start Timing

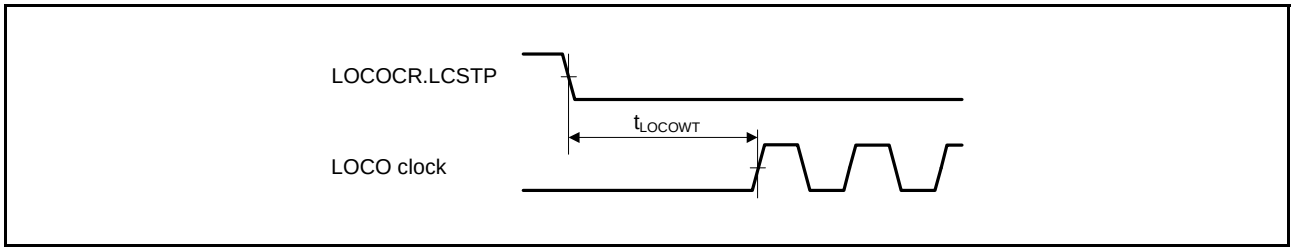


Figure 41.4 LOCO Clock Oscillation Start Timing

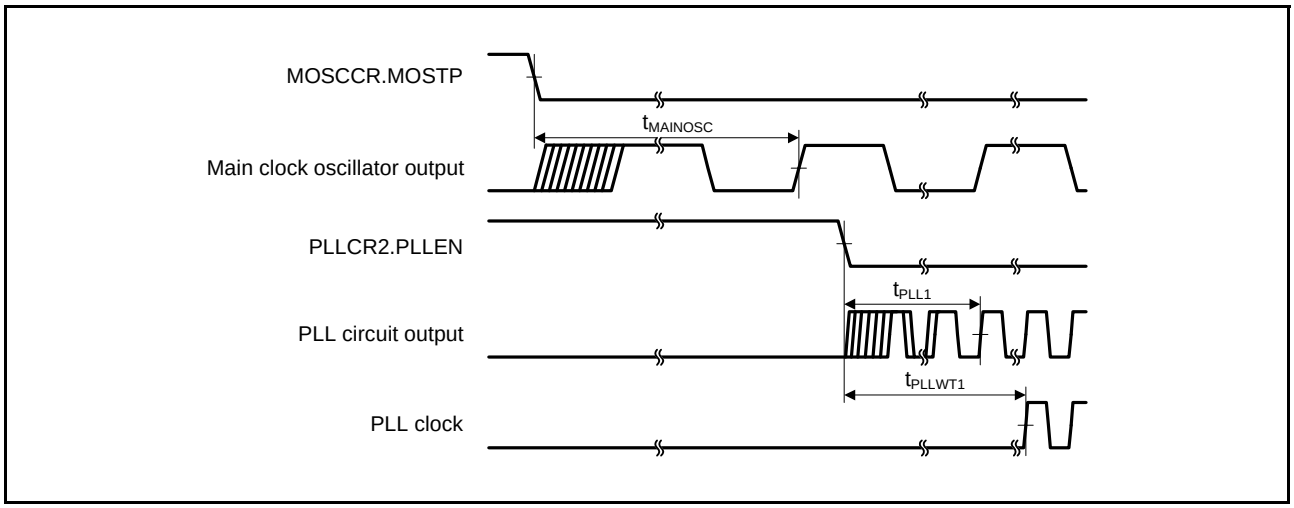


Figure 41.5 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)

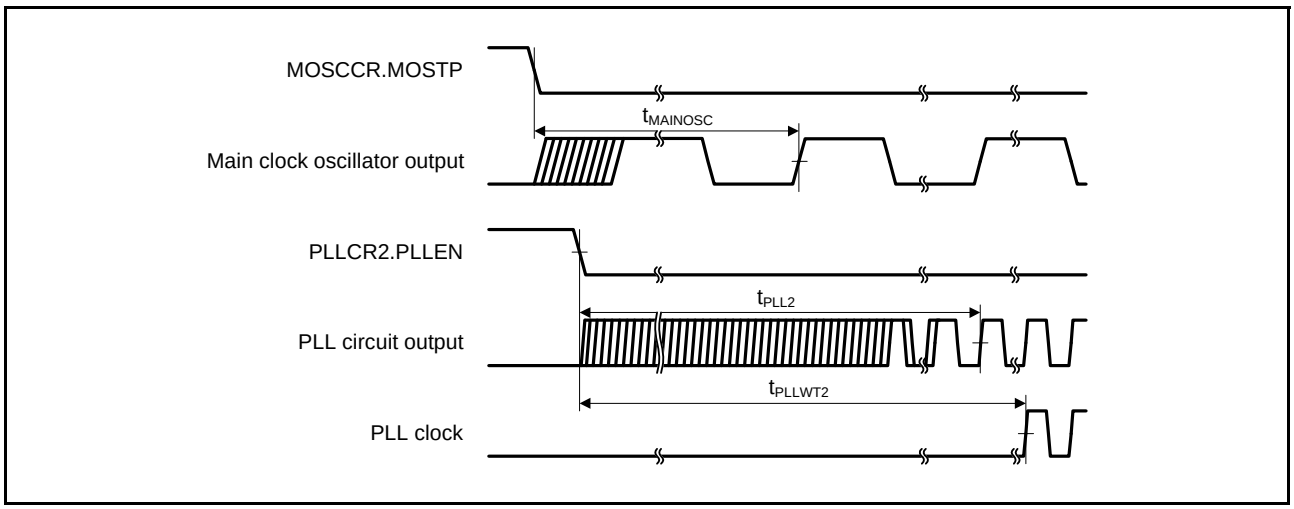


Figure 41.6 PLL Clock Oscillation Start Timing (PLL is Operated before Main Clock Oscillation Has Settled)

41.3.2 Reset Timing

Table 41.19 Reset Timing

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	2	—	—	ms	Figure 41.7
	Deep software standby mode	t_{RESWD}	1	—	—	ms	Figure 41.8
	Software standby mode, low-speed operating modes 1 and 2	t_{RESWS}	1	—	—	ms	
	Programming or erasure of the ROM or E2 DataFlash memory or blank checking of the E2 DataFlash memory	t_{RESWF}	200	—	—	µs	
	Other than above	t_{RESW}	200	—	—	µs	
Wait time after RES# cancellation		t_{RESWT}	59	—	60	t_{CYC}	Figure 41.7
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		t_{RESW2}	112	—	120	t_{CYC}	

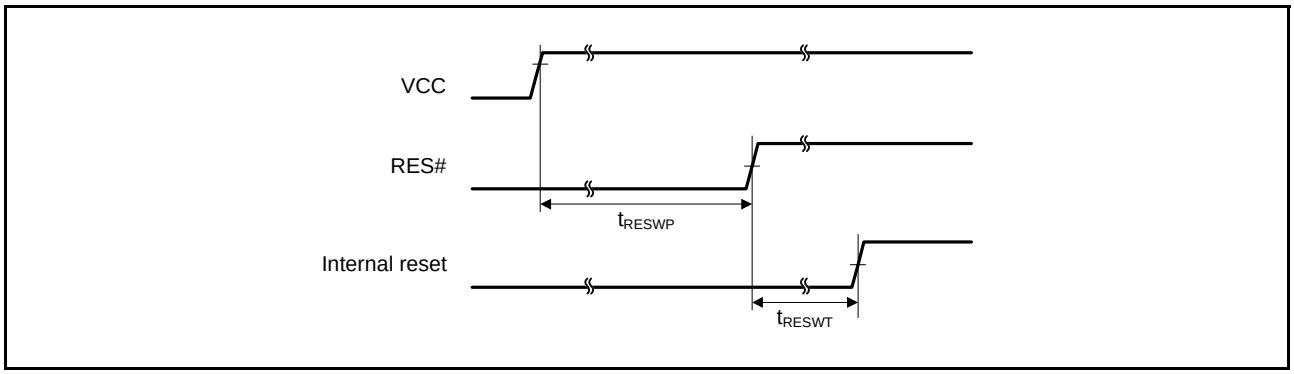


Figure 41.7 Reset Input Timing at Power-On

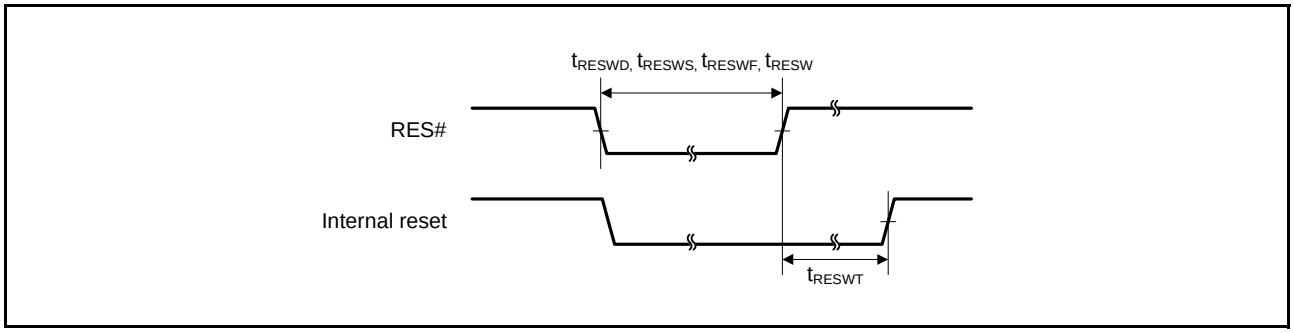


Figure 41.8 Reset Input Timing

41.3.3 Timing of Recovery from Low Power Consumption Modes

Table 41.20 Timing of Recovery from Low Power Consumption Modes

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time after cancellation of software standby mode	Crystal resonator connected to main clock oscillator	Main clock oscillator operating	t_{SBYMC}	10	—	—	ms	Figure 41.9
		Main clock oscillator and PLL circuit operating	t_{SBYPC}	10	—	—	ms	
	External clock input to main clock oscillator	Main clock oscillator operating	t_{SBYEX}	1	—	—	ms	
		Main clock oscillator and PLL circuit operating	t_{SBYPE}	1	—	—	ms	
	Low-speed on-chip oscillator or IWDT-dedicated on-chip oscillator operating		t_{SBYLO}	—	—	800	µs	
Recovery time after cancellation of deep software standby mode			t_{DSBY}	—	—	1	ms	Figure 41.10
Wait time after cancellation of deep software standby mode			t_{DSBYWT}	45	—	46	t_{CYC}	

Note: The wait time varies depending on the state in which each oscillator was when the WAIT instruction was executed. The recovery time when multiple oscillators are operating is the same period as that when the oscillator which requires the longest time of all operating oscillators to recover is operating alone.

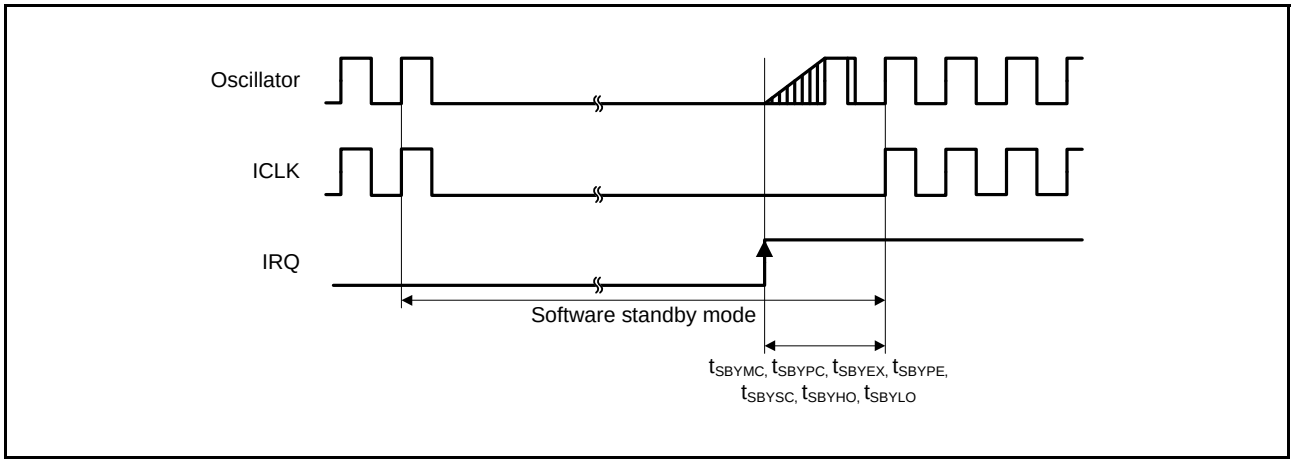


Figure 41.9 Software Standby Mode Cancellation Timing

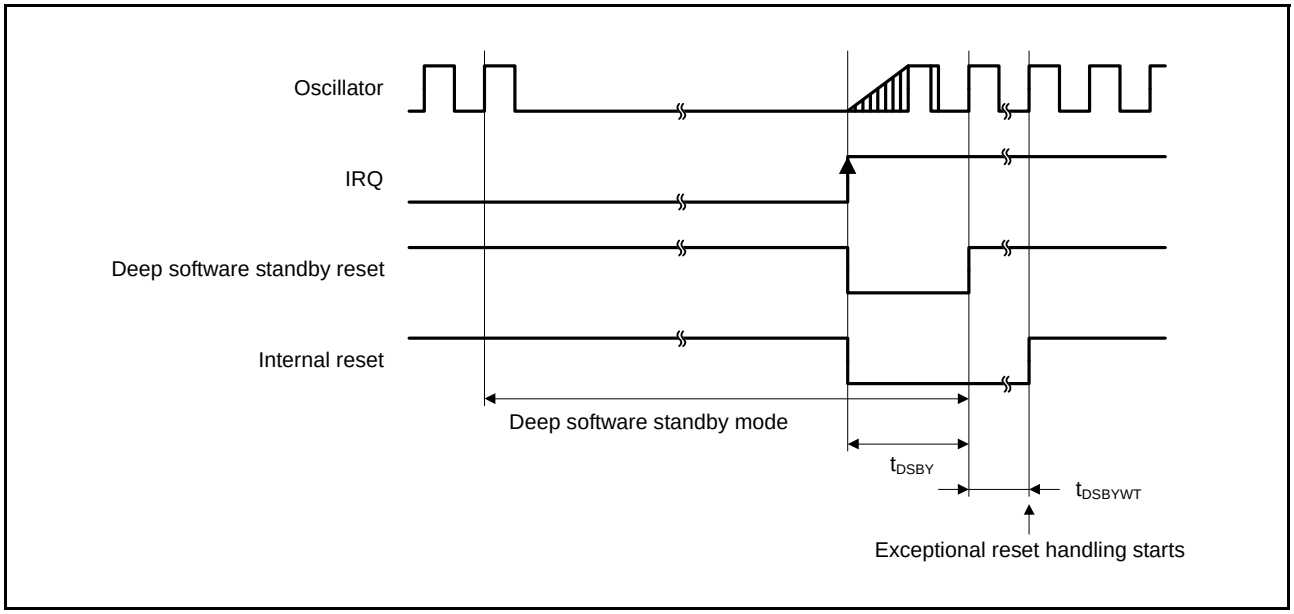


Figure 41.10 Deep Software Standby Mode Cancellation Timing

41.3.4 Control Signal Timing

Table 41.21 Control Signal Timing

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
NMI pulse width	t_{NMIW}	200	—	—	ns	$t_{c\ (PCLKB)}$× 2 ≤ 200ns, Figure 41.11
		$t_{c\ (PCLKB)}$ × 2	—	—	ns	$t_{c\ (PCLKB)}$× 2 > 200ns, Figure 41.11
IRQ pulse width	t_{IRQW}	200	—	—	ns	$t_{c\ (PCLKB)}$× 2 ≤ 200ns, Figure 41.12
		$t_{c\ (PCLKB)}$ × 2	—	—	ns	$t_{c\ (PCLKB)}$× 2 > 200ns, Figure 41.12

Note: 200 ns minimum in deep software standby and software standby modes.

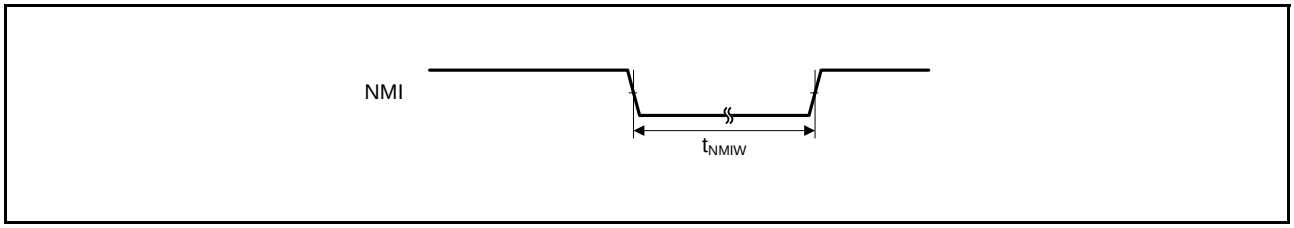


Figure 41.11 NMI Interrupt Input Timing

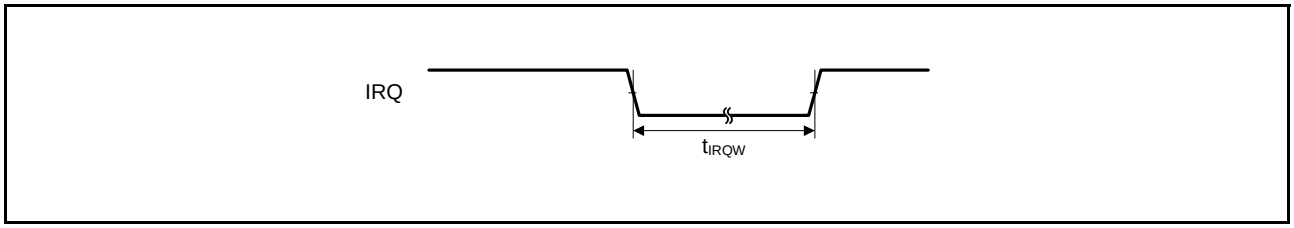


Figure 41.12 IRQ Interrupt Input Timing

41.3.5 Bus Timing

Table 41.22 Bus Timing (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
fBCLK ≤ 54 MHz (BCLK pin output frequency ≤ 27 MHz), VOH = VCC × 0.5, V_{OL} = VCC × 0.5, I_{OH} = –1.0 mA,
I_{OL} = 1.0 mA, CL = 30 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	30	ns	Figure 41.13 to Figure 41.16
Byte control delay time	t_{BCD}	—	30	ns	
CS# delay time	t_{CSD}	—	30	ns	
RD# delay time	t_{RSD}	—	30	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	—	30	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	20	—	ns	Figure 41.17
WAIT# hold time	t_{WTH}	0	—	ns	

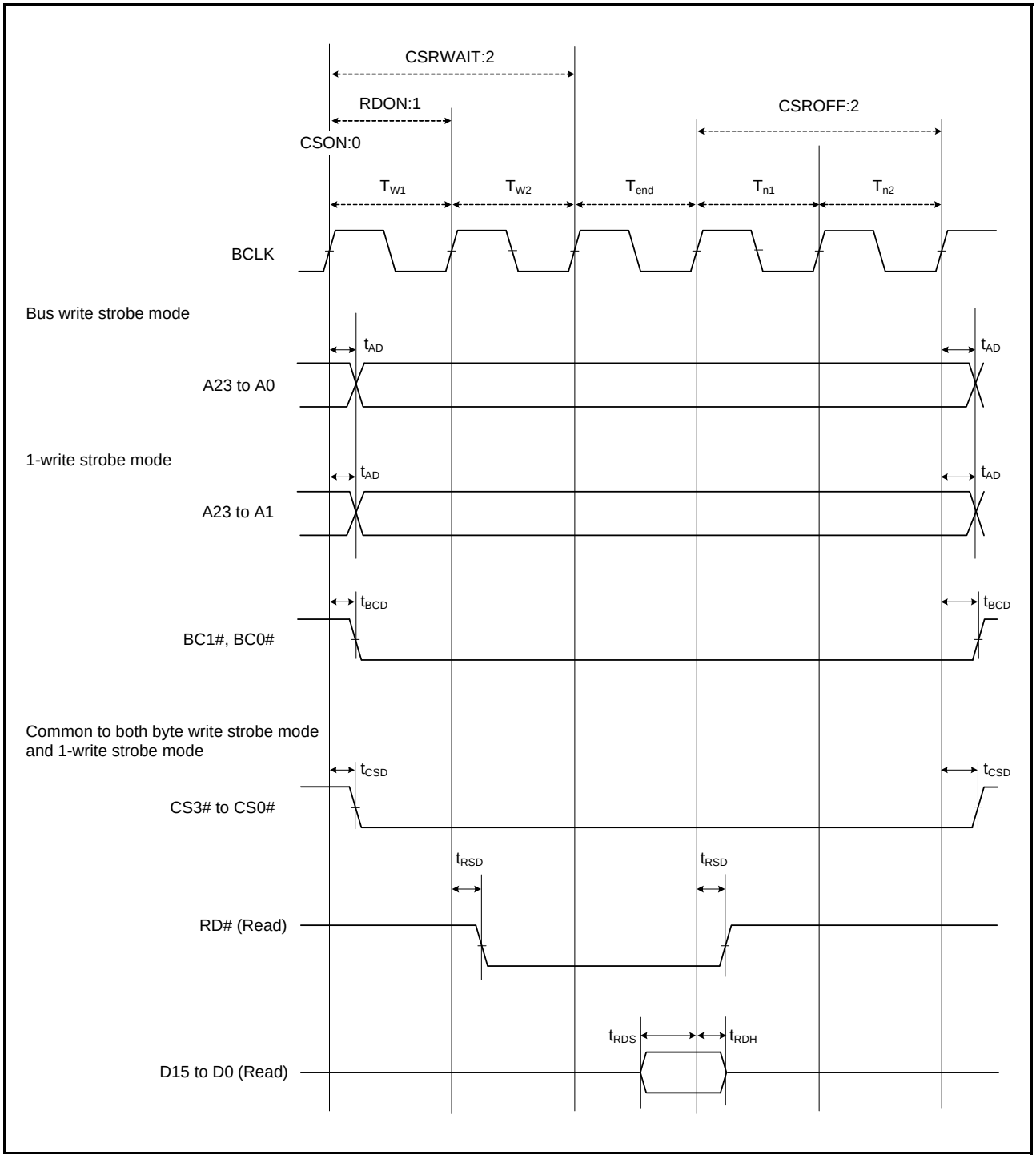
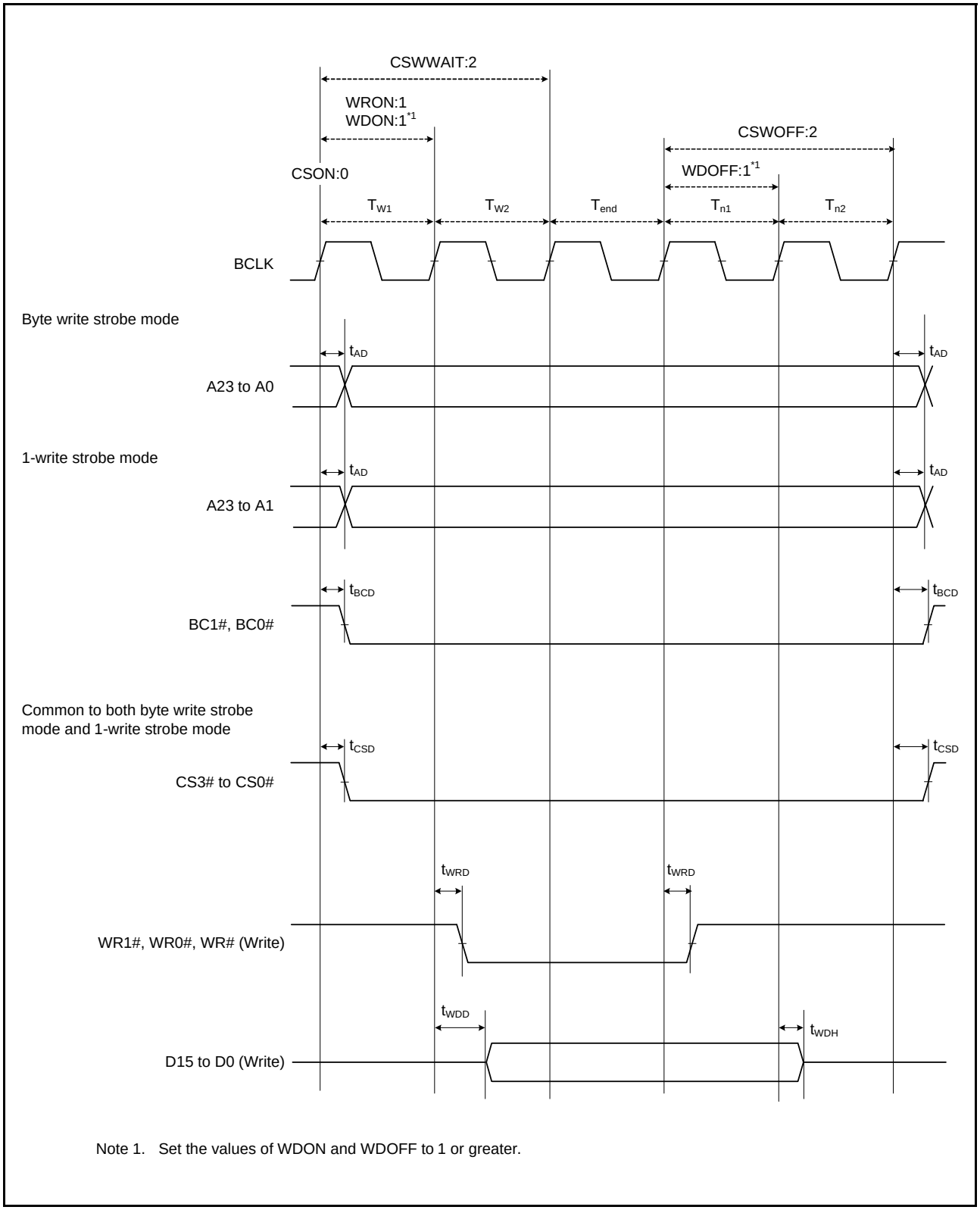


Figure 41.13 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)



Note 1. Set the values of WDON and WDOFF to 1 or greater.

Figure 41.14 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

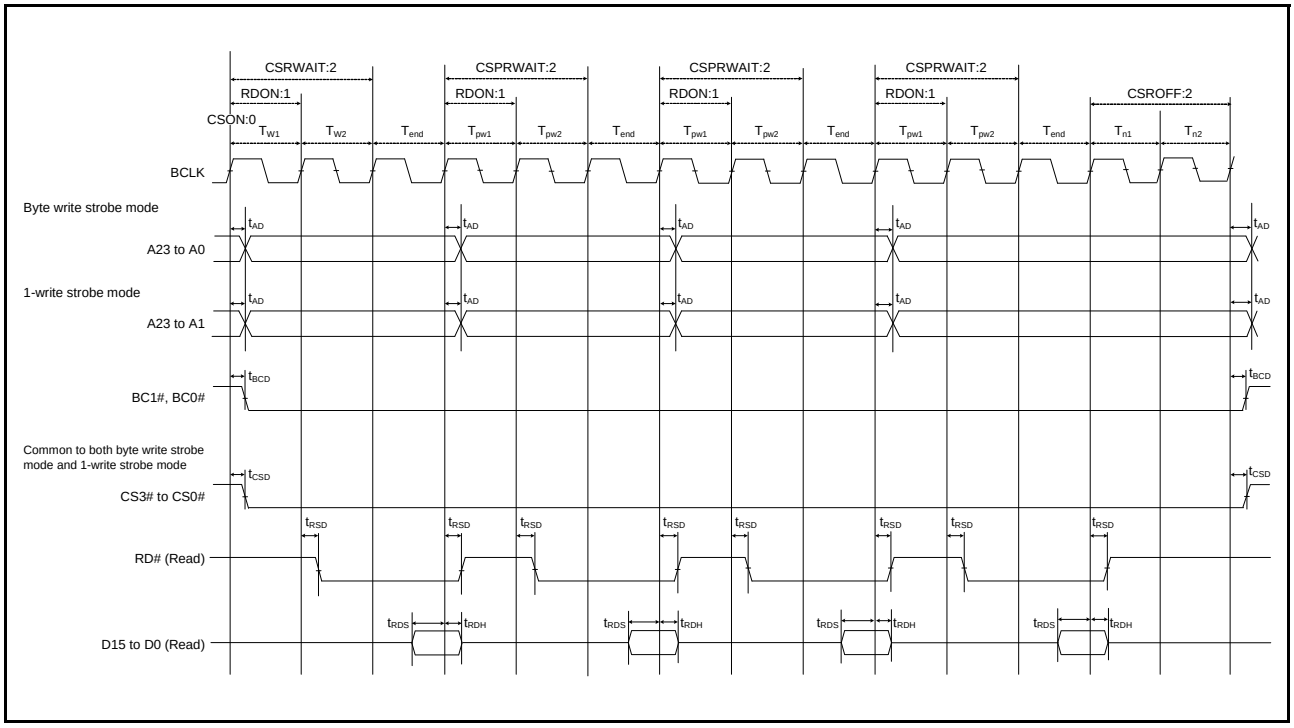
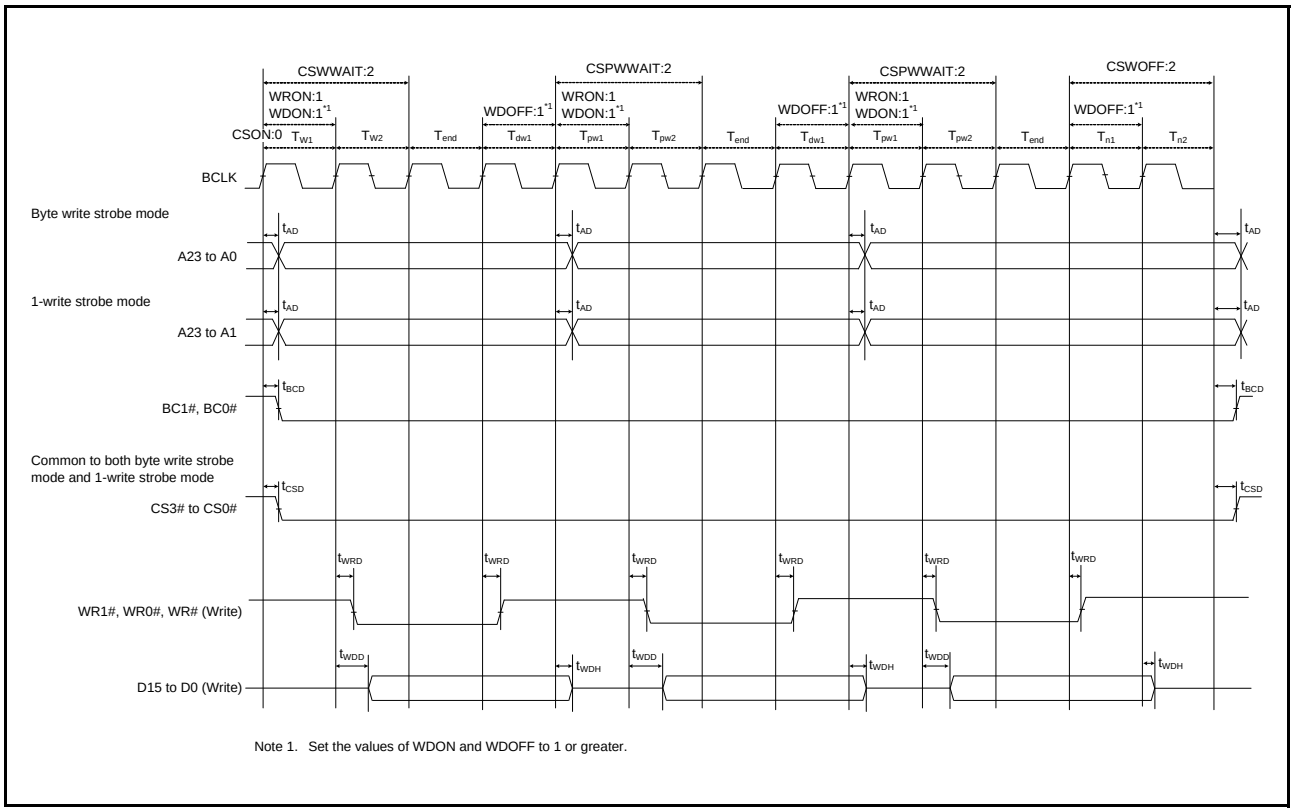


Figure 41.15 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)



Note 1. Set the values of WDON and WDOFF to 1 or greater.

Figure 41.16 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

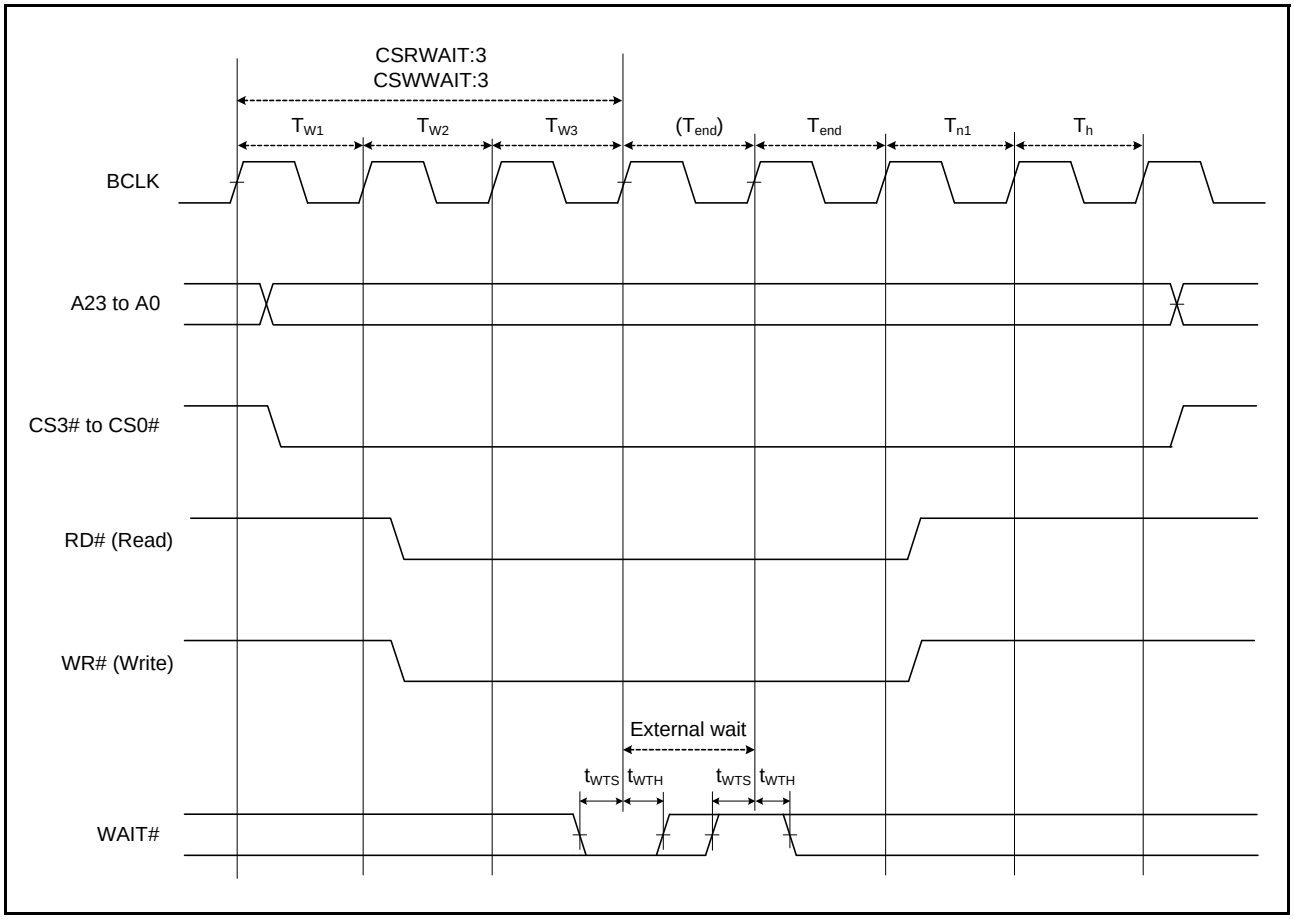


Figure 41.17 External Bus Timing/External Wait Control

Table 41.23 Bus Timing (Multiplexed Bus)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
fBCLK ≤ 54 MHz (BCLK pin output frequency ≤ 27 MHz), VOH = VCC × 0.5, V_{OL} = VCC × 0.5, I_{OH} = –1.0 mA,
I_{OL} = 1.0 mA, CL = 30 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	30	ns	Figure 41.18, Figure 41.19
Byte control delay time	t_{BCD}	—	30	ns	
CS# delay time	t_{CSD}	—	30	ns	
RD# delay time	t_{RSD}	—	30	ns	
ALE delay time	t_{ALED}	—	30	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	—	30	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	20	—	ns	Figure 41.17
WAIT# hold time	t_{WTH}	0	—	ns	

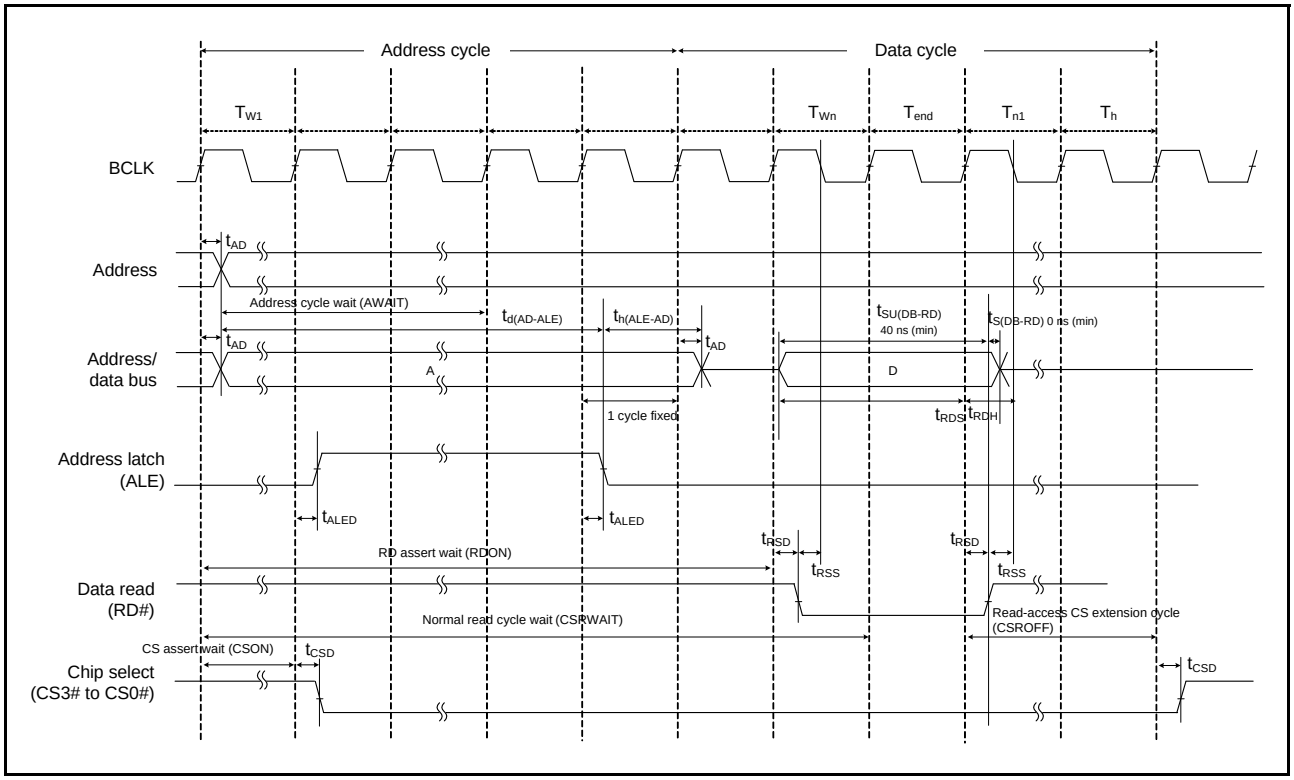


Figure 41.18 Example of Operation in Read Access over the External Bus (Multiplexed)

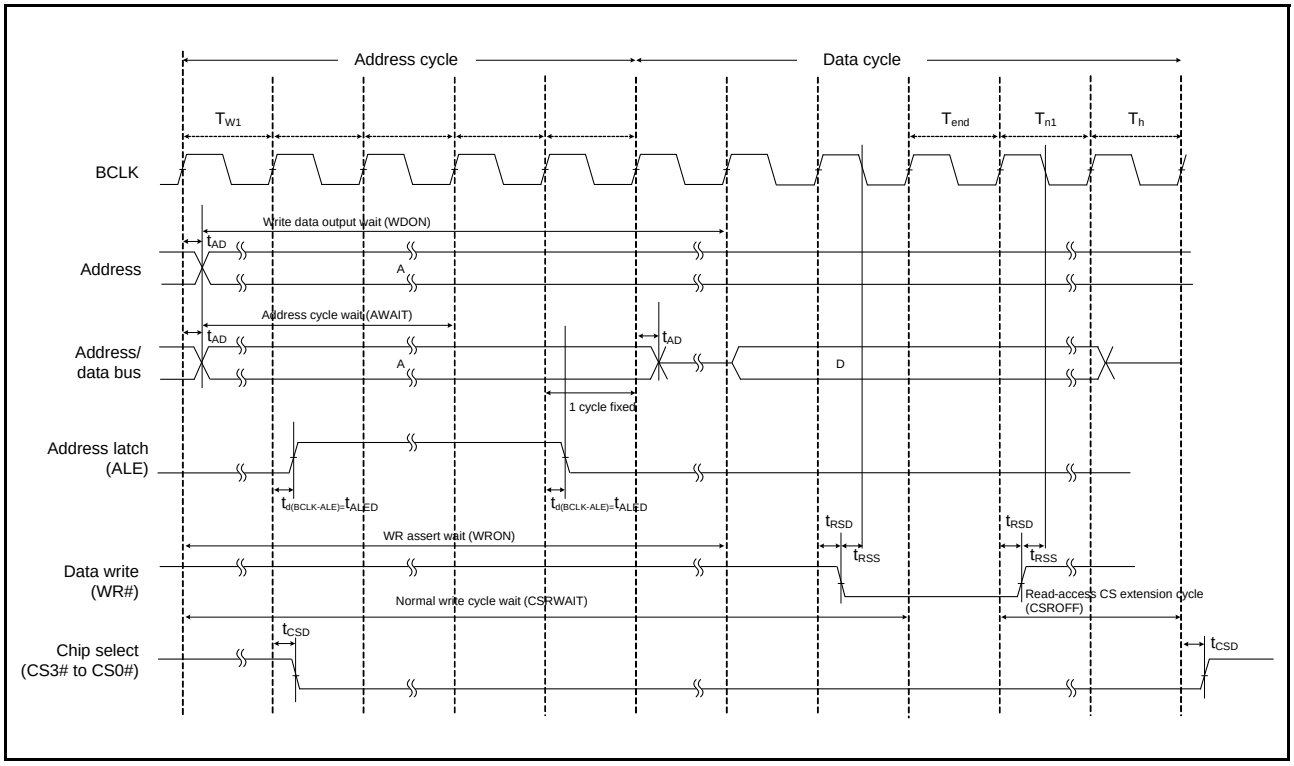


Figure 41.19 Example of Operation in Write Access over the External Bus (Multiplexed)

41.3.6 Timing of On-Chip Peripheral Modules

Table 41.24 Timing of On-Chip Peripheral Modules (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width		t_{PRW}	1.5	—	t_{Pcyc}	Figure 41.20
MTU/TPU	Input capture input pulse width	Single-edge setting	t_{TICW}	1.5	—	t_{Pcyc}	Figure 41.21
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—	t_{Pcyc}	Figure 41.22
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		
POE	POE# input pulse width		t_{POEW}	1.5	—	t_{Pcyc}	Figure 41.23
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{Pcyc}	Figure 41.24
		Both-edge setting		2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 41.25
		Clock synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	20	ns	
	Input clock fall time		t_{SCKf}	—	20	ns	
	Output clock cycle	Asynchronous	t_{Scyc}	16	—	t_{Pcyc}	Figure 41.26
		Clock synchronous		4	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	20	ns	
	Output clock fall time		t_{SCKf}	—	20	ns	
	Transmit data delay time	Clock synchronous	t_{TXD}	—	40	ns	
	Receive data setup time	Clock synchronous	t_{RXS}	40	—	ns	
	Receive data hold time	Clock synchronous	t_{RXH}	40	—	ns	
A/D converter	Trigger input pulse width		t_{TRGW}	1.5	—	t_{Pcyc}	Figure 41.27
CAC	CACREF input pulse width	$t_{Pcyc} \leq t_{cac}$*2	t_{CACREF}	4.5 t_{cac} + 3 t_{Pcyc}	—	ns	
		$t_{Pcyc} > t_{cac}$*2		5 t_{cac} + 6.5 t_{Pcyc}	—	ns	

Note 1. t_{Pcyc}: PCLKB cycle
Note 2. t_{cac}: CAC count clock source cycle

Table 41.25 Timing of On-Chip Peripheral Modules (2)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	4096	t_{Pcyc}	Figure 41.28
		Slave			8	4096		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock rise/ fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input			—	1	µs	
	Data input setup time	Master		t_{SU}	15	—	ns	Figure 41.29 to Figure 41.34
		Slave			$20 - t_{Pcyc}$	—		
	Data input hold time	Master	PCLKB set to a division ratio other than divided by 2	t_H	t_{Pcyc}	—	ns	
			PCLKB set to divided by 2		0	—		
		Slave			$20 + 2 \times t_{Pcyc}$	—		
	SSL setup time	Master		t_{LEAD}	1	8	t_{SPcyc}	
		Slave			4	—	t_{Pcyc}	
	SSL hold time	Master		t_{LAG}	1	8	t_{SPcyc}	
		Slave			4	—	t_{Pcyc}	
	Data output delay time	Master		t_{OD}	—	18	ns	
		Slave			—	$3 \times t_{Pcyc} + 40$		
	Data output hold time	Master		t_{OH}	0	—	ns	
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
		Slave			$4 \times t_{Pcyc}$	—		
	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}	—	4	t_{Pcyc}	Figure 41.33, Figure 41.34
	Slave output release time			t_{REL}	—	3	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLKB cycle

Table 41.26 Timing of On-Chip Peripheral Modules (3)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C
When high-drive output is selected by the drive capacity control register

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	t_{SPcyc}	4	65536	t_{Pcyc}	Figure 41.28
	SCK clock cycle input (slave)		8	65536		
	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}	40	—	ns	Figure 41.29 to Figure 41.34
	Data input hold time	t_H	40	—	ns	
	SS input setup time	t_{LEAD}	6	—	t_{Pcyc}	
	SS input hold time	t_{LAG}	6	—	t_{Pcyc}	
	Data output delay time	t_{OD}	—	40	ns	
	Data output hold time	t_{OH}	–10	—	ns	
	Data rise/fall time	t_{Dr}, t_{Df}	—	20	ns	
	SS input rise/fall time	t_{SSLr}, t_{SSLf}	—	20	ns	
	Slave access time	t_{SA}	—	5	t_{Pcyc}	Figure 41.33, Figure 41.34
	Slave output release time	t_{REL}	—	5	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLKB cycle

Table 41.27 Timing of On-Chip Peripheral Modules (4)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.(*1, *2)	Max.	Unit	Test Conditions
RIIC (Standard-mode)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 41.35
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	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	—	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	20 + 0.1C_b	300	ns	
	SCL, SDA input fall time	t_{Sf}	20 + 0.1C_b	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference count clock (IICφ) cycle

Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bits = 1.

Note 2. C_b indicates the total capacity of the bus line.

Table 41.28 Timing of On-Chip Peripheral Modules (5)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item		Symbol	Min.(*1, *2)	Max.*3	Unit	Test Conditions
Simple IIC (Standard-mode)	SCL, SDA input rise time	t_{Sr}	—	1000	ns	Figure 41.35
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA 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	
	SCL, SDA capacitive load	C_b	—	400	pF	
Simple IIC (Fast-mode)	SCL, SDA input rise time	t_{Sr}	20 + 0.1C_b	300	ns	
	SCL, SDA input fall time	t_{Sf}	20 + 0.1C_b	300	ns	
	SCL, SDA 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	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bits = 1.
Note 2. C_b indicates the total capacity of the bus line.
Note 3. t_{Pcyc}: PCLKB cycle

Table 41.29 Timing of On-Chip Peripheral Modules (6)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions
CEC fall time	t_{cf}	—	50	µs	C_b = 1600 pF, R_b = 27 kΩ
					C_b = 7700 pF, R_b = 3 kΩ

Note 1. Cb: Communication line load capacitance; Rb: Communication line external pull-up resistance

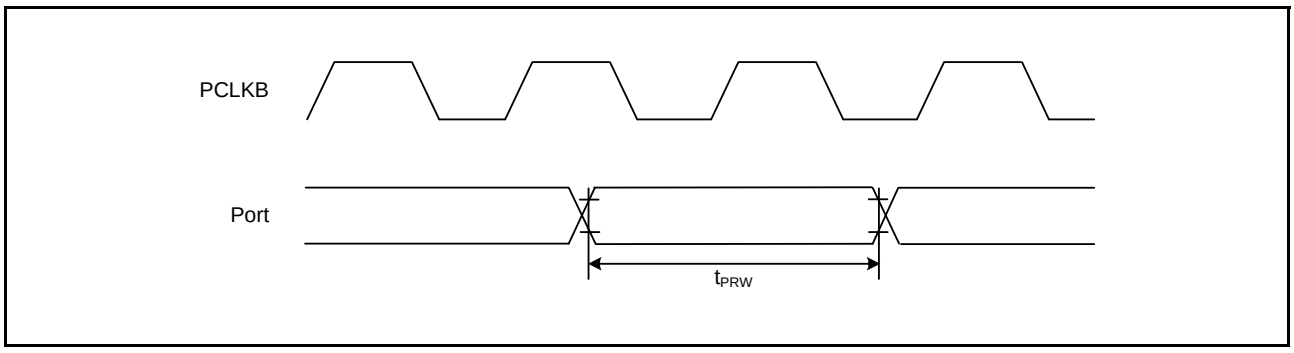


Figure 41.20 I/O Port Input Timing

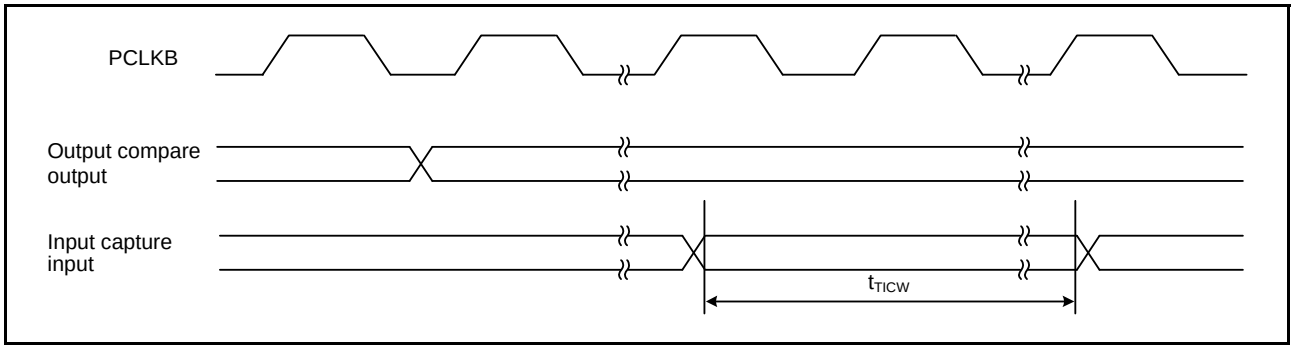


Figure 41.21 MTU/TPU Input/Output Timing

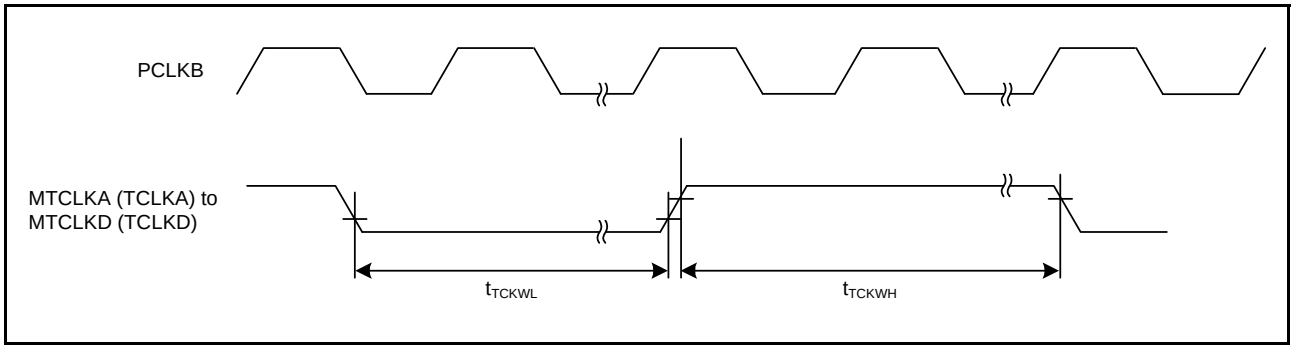


Figure 41.22 MTU/TPU Clock Input Timing

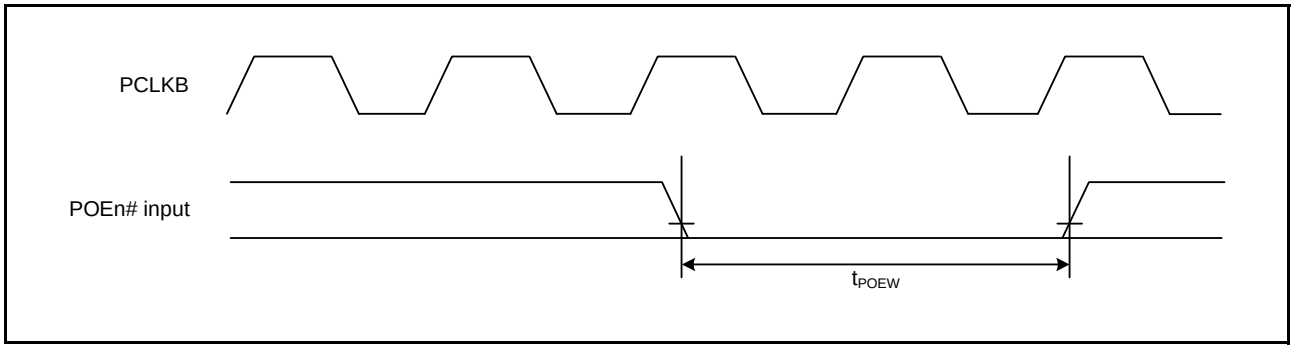


Figure 41.23 POE# Input Timing

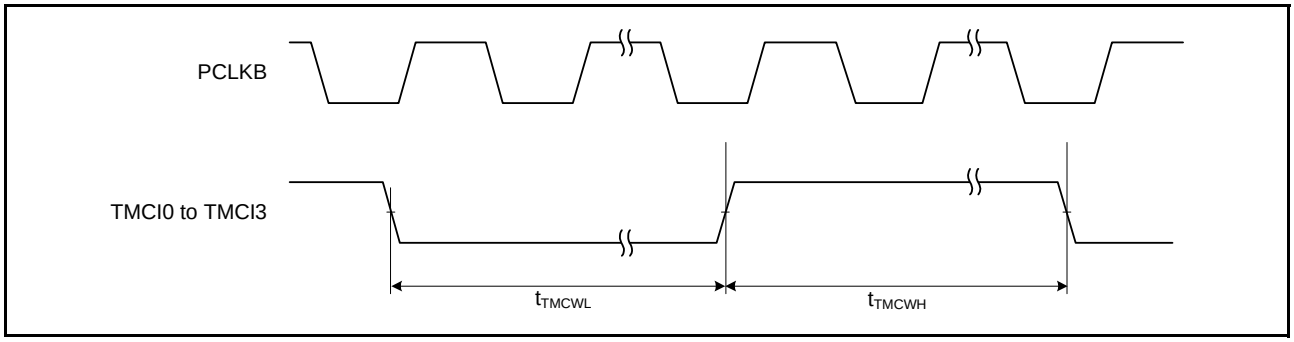


Figure 41.24 8-Bit Timer Clock Input Timing

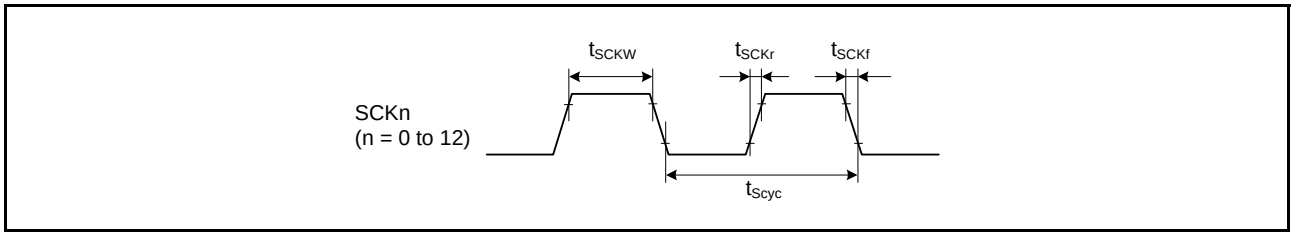


Figure 41.25 SCK Clock Input Timing

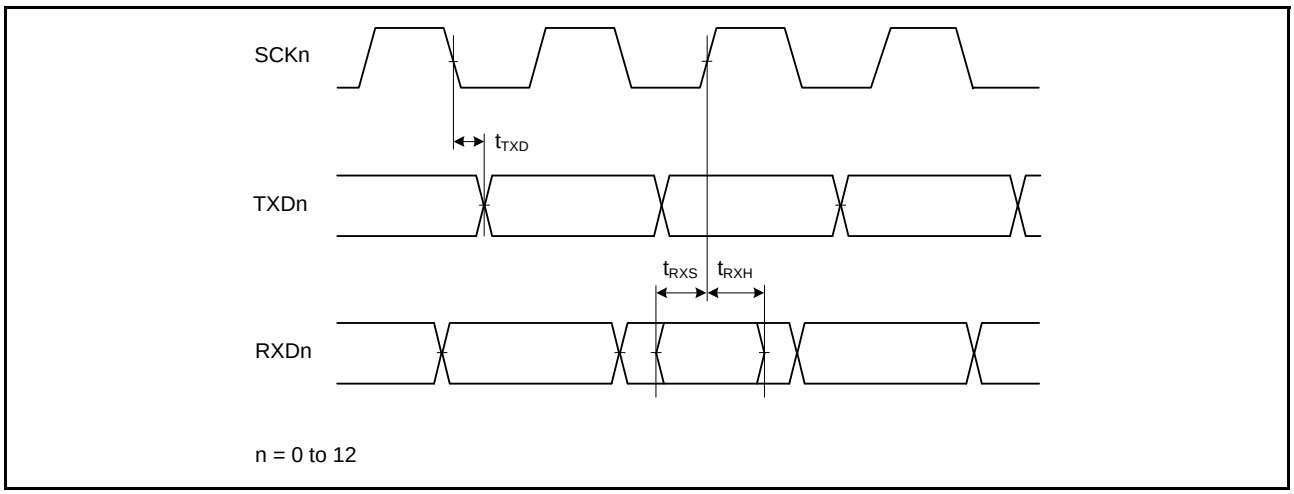


Figure 41.26 SCI Input/Output Timing: Clock Synchronous Mode

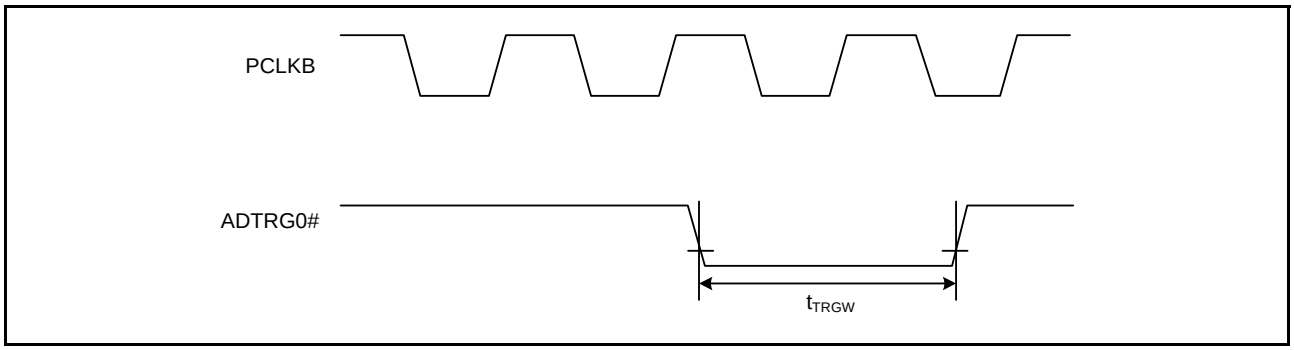


Figure 41.27 A/D Converter External Trigger Input Timing

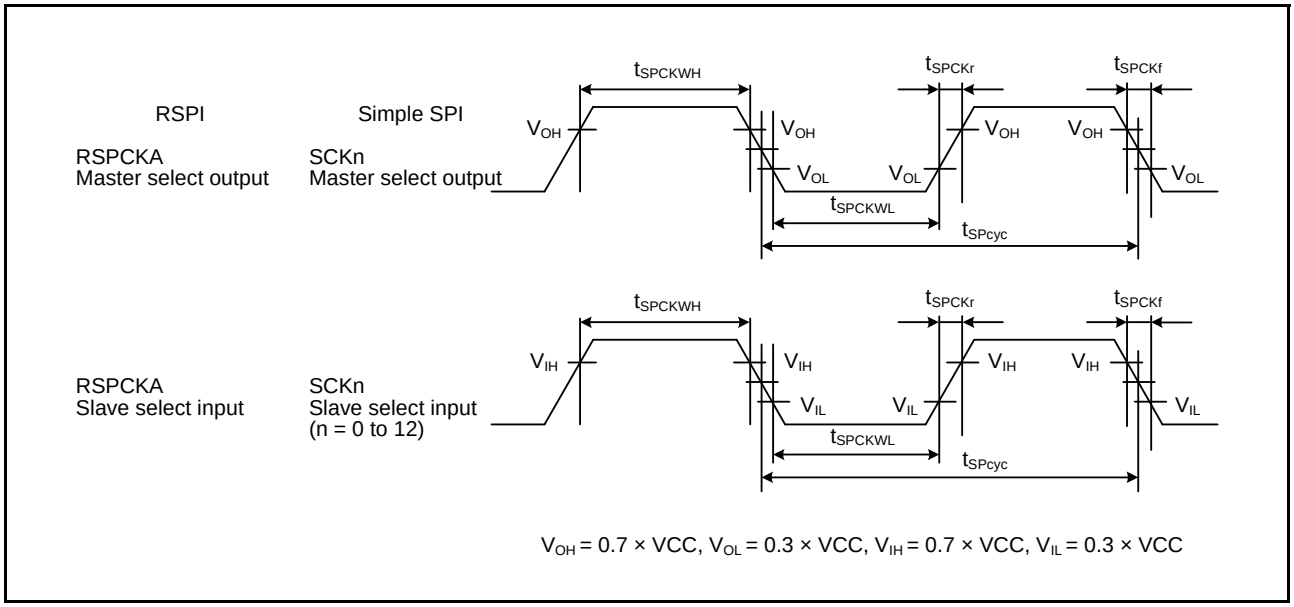


Figure 41.28 RSPI Clock Timing and Simple SPI Clock Timing

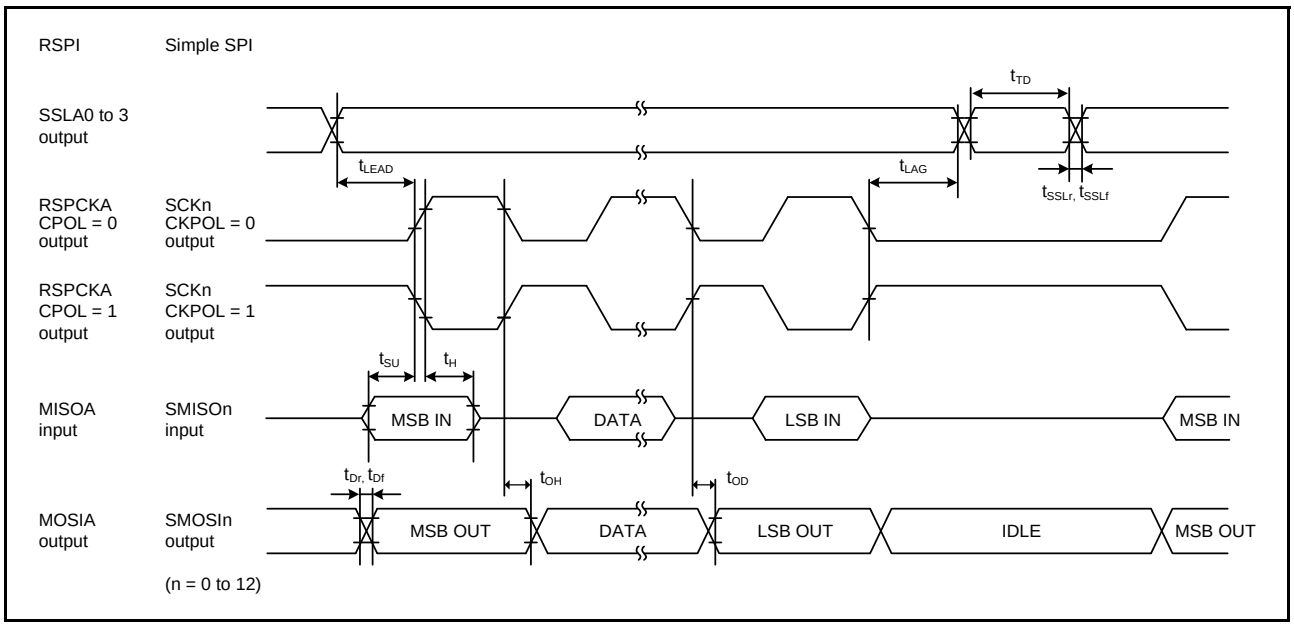


Figure 41.29 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 1)

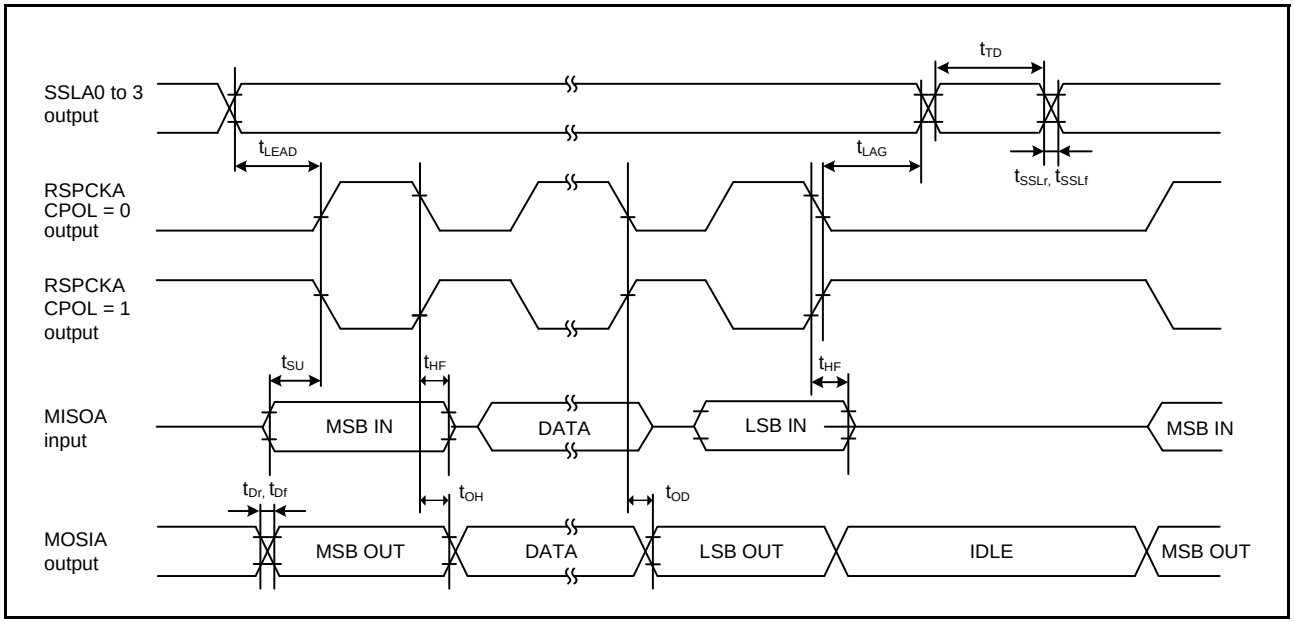


Figure 41.30 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Divided by 2)

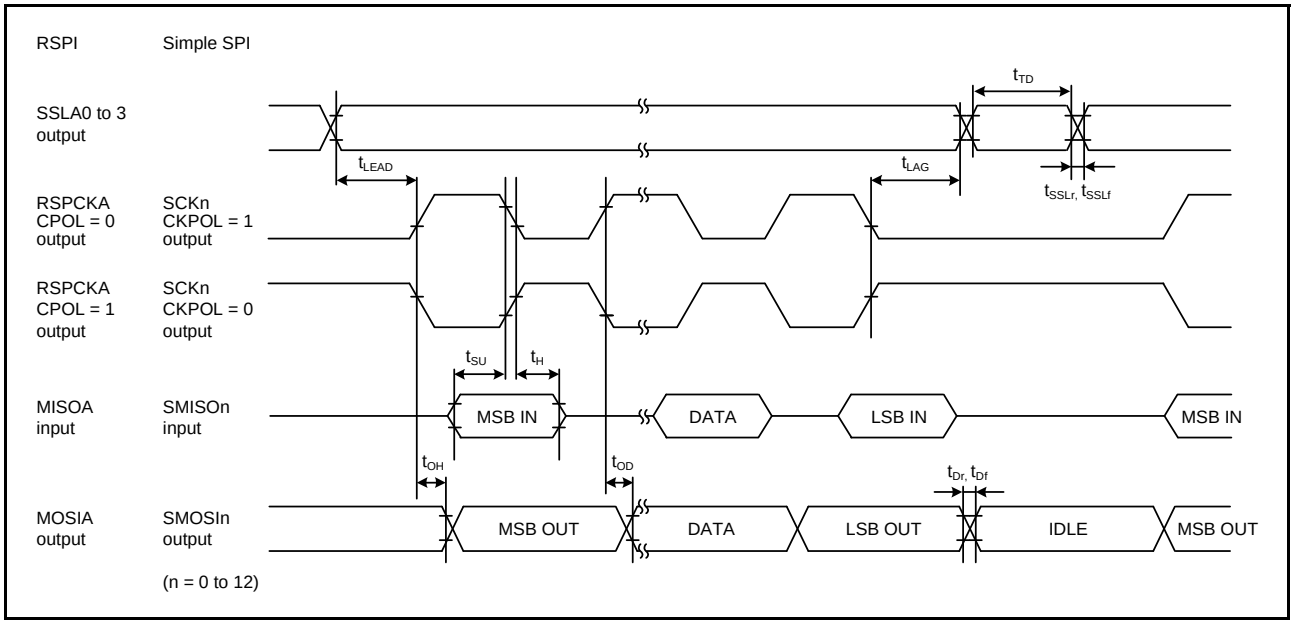


Figure 41.31 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 0)

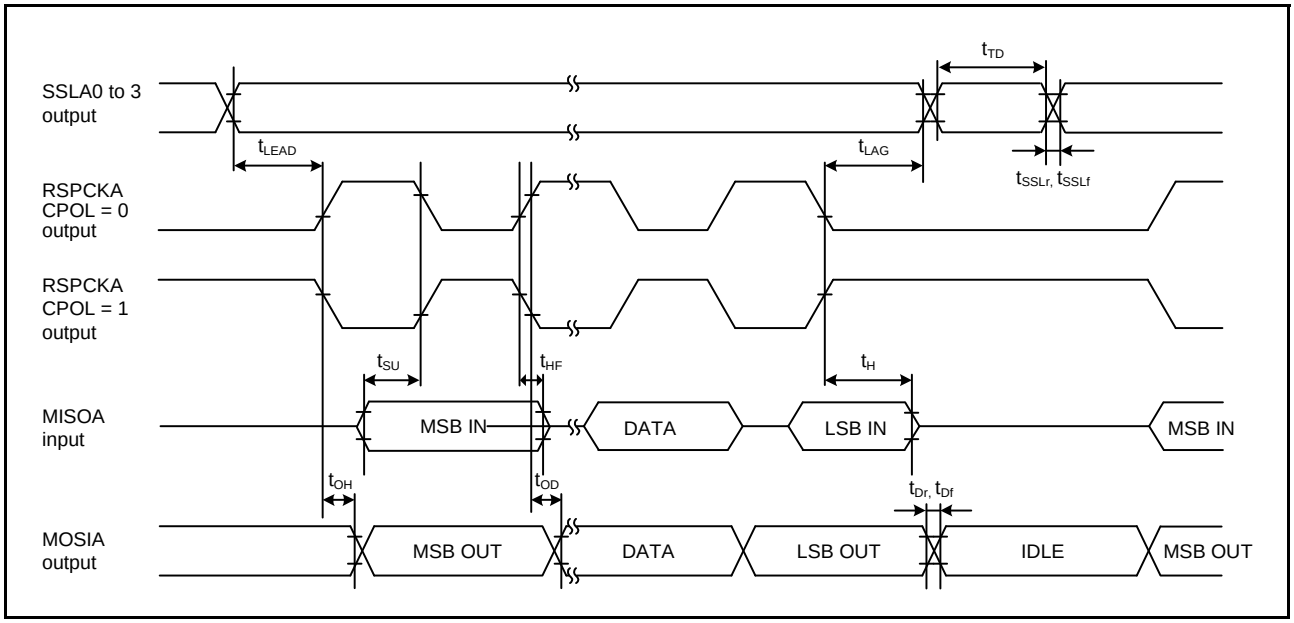


Figure 41.32 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Divided by 2)

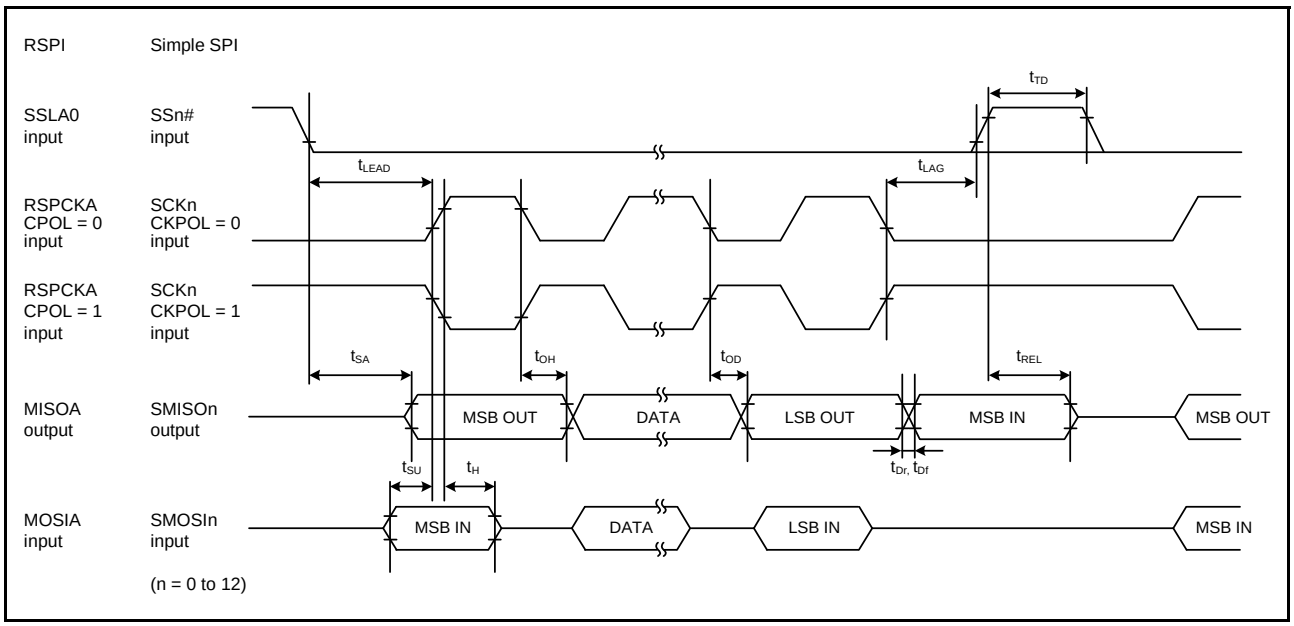


Figure 41.33 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)

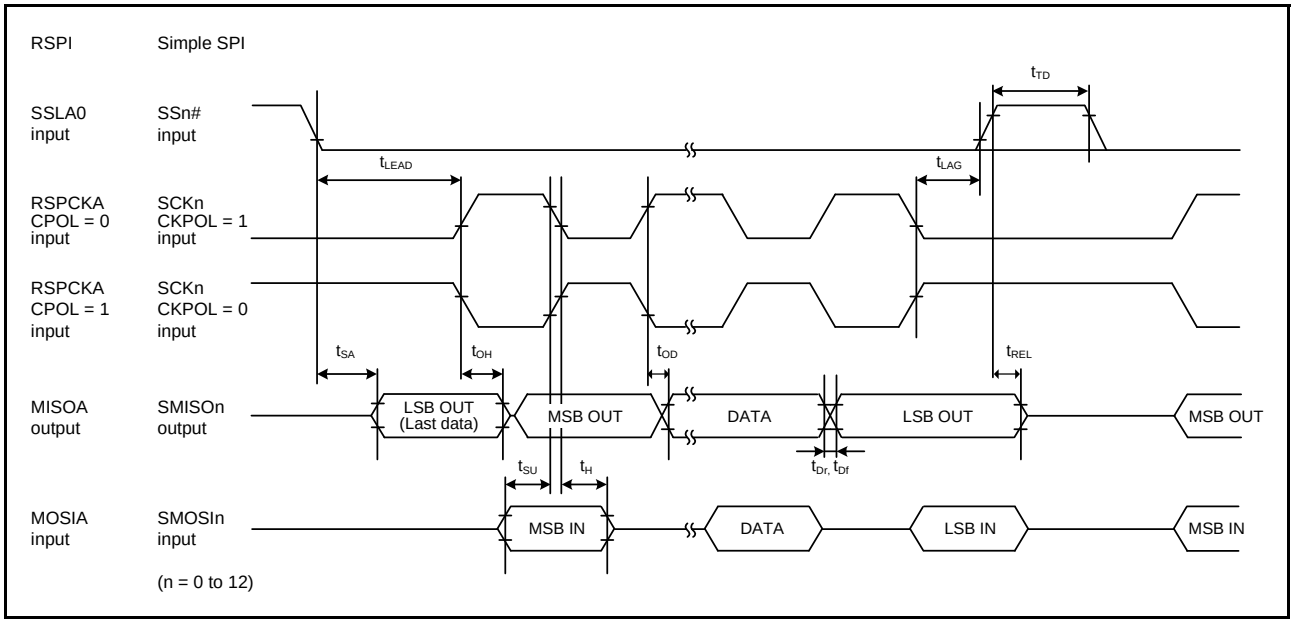
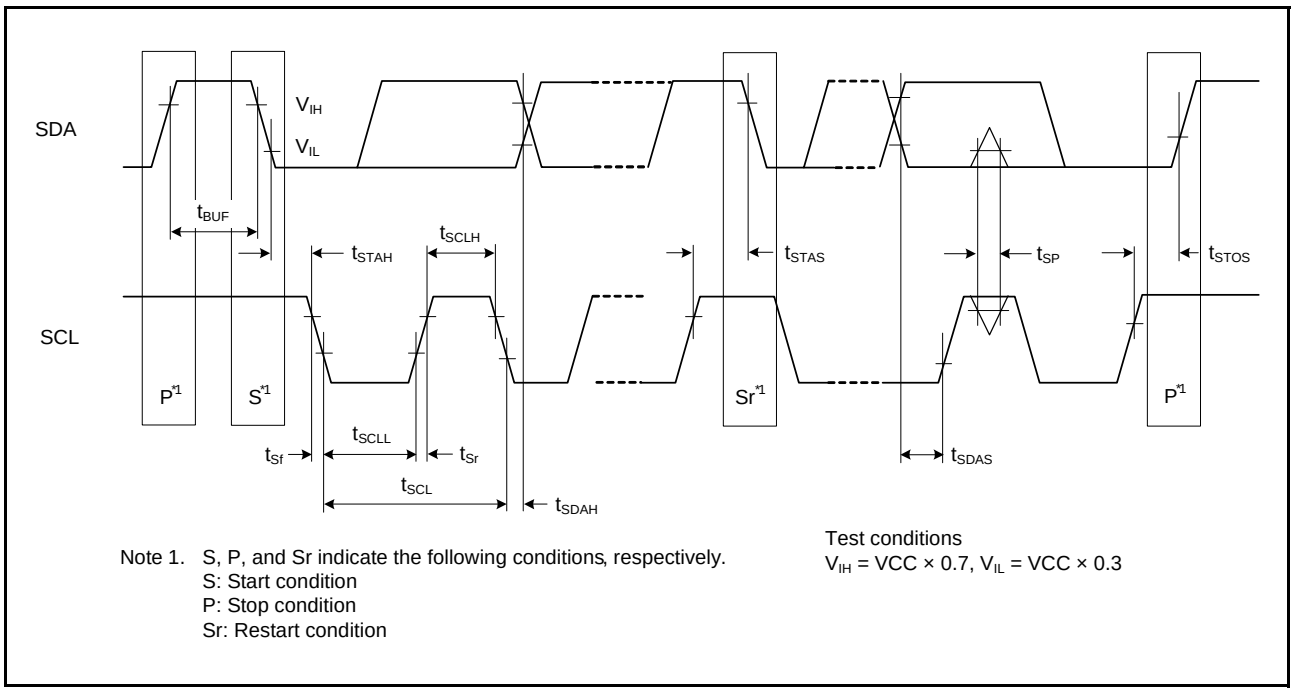


Figure 41.34 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3

Figure 41.35 RIIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing

41.4 A/D Conversion Characteristics

Table 41.30 A/D Conversion Characteristics

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
A/D conversion clock frequency (fPCLKD)		1	—	54	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (AD clock = 54 MHz)	Permissible signal source impedance = 1.0 kΩ	1 (0.4)*2	—	—	µs	High-precision channel
	Permissible signal source impedance = 1 kΩ, AVCC ≥ 4.0 V	1.9 (1.3)*2	—	—		Normal-precision channel
	Permissible signal source impedance = 1 kΩ, AVCC ≥ 2.7 V	2.5 (1.9)*2	—	—		
Analog input capacitance		—	—	30	pF	
Offset error		—	±2.0	±6.0	LSB	High-precision channel (SH used)
		—	±1.5	±3.0		High-precision channel (SH not used)
		—	±2.0	±7.5		Normal-precision channel
Full-scale error		—	±2.0	±6.0	LSB	High-precision channel (SH used)
		—	±1.5	±3.0		High-precision channel (SH not used)
		—	±2.0	±7.5		Normal-precision channel
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±4.0	±8.0	LSB	High-precision channel (SH used)
		—	±2.0	±6.0		High-precision channel (SH not used)
		—	±2.5	±8.0		Normal-precision channel
DNL differential nonlinearity error		—	±2.0	±6.0	LSB	High-precision channel (SH used)
		—	±1.5	±3.0		High-precision channel (SH not used)
		—	±2.0	±4.0		Normal-precision channel
INL integral nonlinearity error		—	±2.0	±6.0	LSB	High-precision channel (SH used)
		—	±1.5	±3.0		High-precision channel (SH not used)
		—	±2.0	±4.0		Normal-precision channel

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note: When using the channel-dedicated sample-and-hold circuit, use the AN000 to AN002 analog input voltage (V_{AN}) that satisfies all the following conditions: 0.25 V ≤ V_{AN} ≤ AVCC0 - 0.25 V, V_{AN} ≤ VREFH0, and AVCC0 ≥ 2.7 V.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

Table 41.31 Sampling Time

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Typ.	Unit	Test Conditions
Sampling time	High-precision channel	Ts	0.2 + 0.16 × R0 (KΩ)	µs	Figure 41.36
	Normal-precision channel		0.3 + 0.16 × R0 (KΩ)		

Table 41.32 A/D Converter Channel Classification

Classification	Target Channel	Channel-Dedicated Sample-and-Hold Circuit	Condition
High-accuracy channels	AN000 to AN002	Used	AVCC0 = VREFH0 = 2.7 to 3.6V (3-V package) AVCC0 = VREFH0 = 4.0 to 5.5V (5-V package) AVSS0 = VREFL0 = 0V 0.25V ≤ V_{AN} ≤ AVCC0 – 0.25V V_{AN} ≤ VREFH0
		Not used	AVCC0 = VREFH0 = 2.7 to 3.6V (3-V package) AVCC0 = VREFH0 = 4.0 to 5.5V (5-V package) AVSS0 = VREFL0 = 0V 0V ≤ V_{AN} ≤ VREFH0
	AN003 to AN007	—	
Normal-accuracy channels	AN008 to AN015	—	

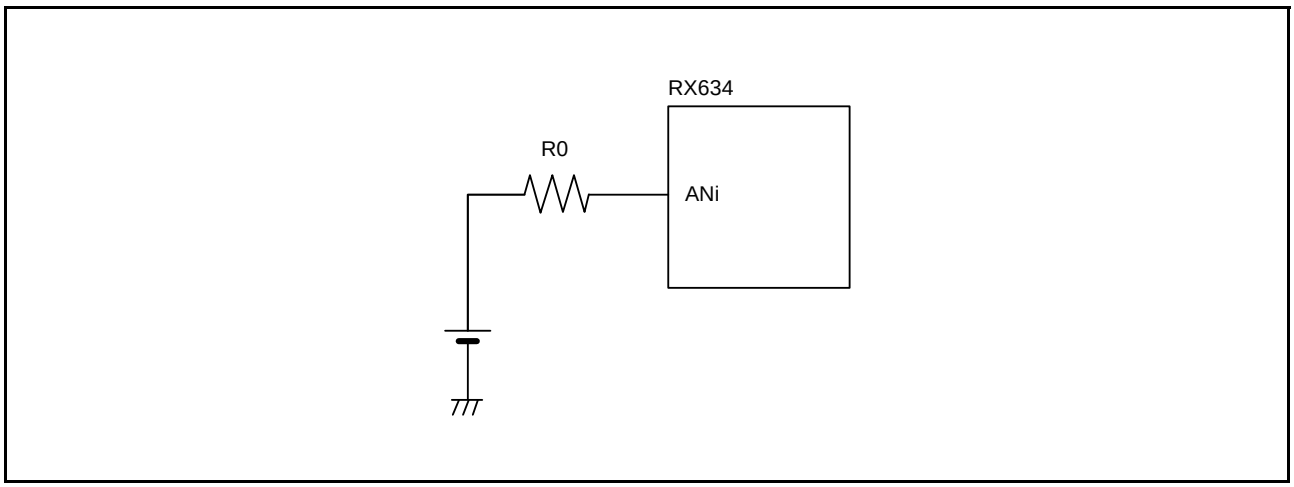


Figure 41.36 Internal Equivalent Circuit of Analog Input Pin

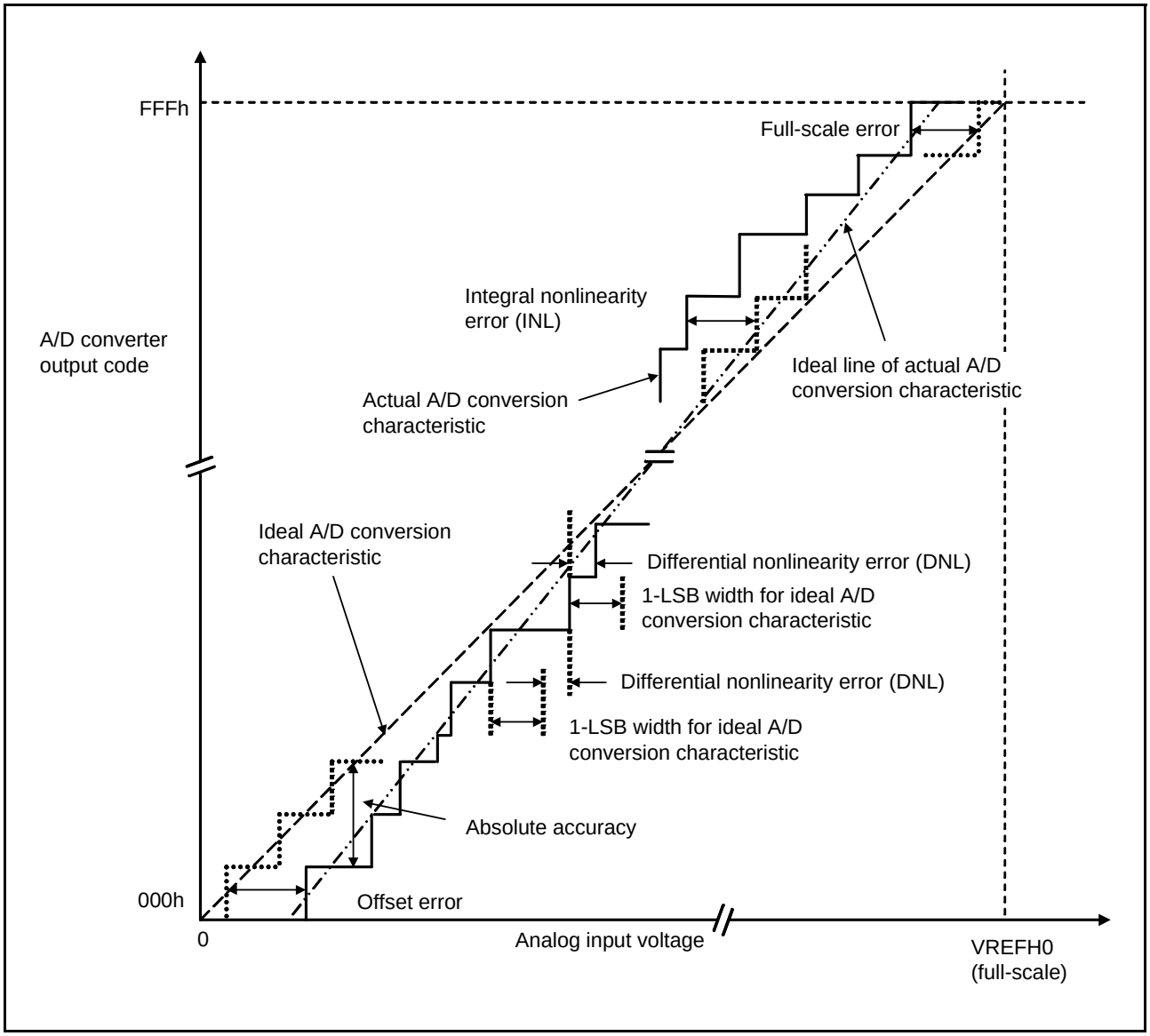


Figure 41.37 Illustration of A/D Converter Characteristic Terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage (VREFH0 = 5.12 V), then 1-LSB width becomes 1.25 mV, and 0 mV, 1.25 mV, 2.5 mV, ... are used as analog input voltages.

If analog input voltage is 10 mV, absolute accuracy = ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh though an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actually output code.

Offset error

Offset error is the difference between a transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

41.5 D/A Conversion Characteristics

Table 41.33 D/A Conversion Characteristics (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V,
T_a = –40 to +85°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	10	Bit	
Conversion time	—	—	±3.0	µs	20-pF capacitive load
Absolute accuracy	—	±3.0	±5.0	LSB	4-MΩ resistive load
	—	—	±4.0	LSB	8-MΩ resistive load
RO output resistance	—	3.6	—	kΩ	

41.6 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 41.34 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (1)

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	2.46	2.58	2.7	V	Figure 41.38
	Voltage detection circuit (LVD0)	V_{DET0}	2.7	2.82	2.94		Figure 41.39
	Voltage detection circuit (LVD1)*1	V_{DET1_8}	2.75	2.90	3.05		Figure 41.40
		V_{DET1_9}	2.70	2.85	3.00		
		V_{DET1_A}	2.73	2.88	3.03		
	Voltage detection circuit (LVD2)*2	V_{DET2_8}	2.75	2.90	3.05		Figure 41.41
		V_{DET2_9}	2.70	2.85	3.00		
		V_{DET2_A}	2.73	2.88	3.03		
Internal reset time	Power-on reset (POR)	t_{POR}		9.7		ms	Figure 41.38
	Voltage detection circuit (LVD0)	t_{LVO0}		9.7			Figure 41.39
	Voltage detection circuit (LVD1)	t_{LVO1}		0.9			Figure 41.40
	Voltage detection circuit (LVD2)	t_{LVO2}		0.9			Figure 41.41
Minimum VCC down time*3		t_{VOFF}	200	—	—	µs	Figure 41.39 to Figure 41.41
Response delay time		t_{DET}			200	µs	
LVD operation stabilization time (after LVD is enabled)		$Td_{(E-A)}$			3	µs	Figure 41.40, Figure 41.41
Hysteresis width (LVD1 and LVD2)		V_{LVH}		80		mV	

Note 1. # in the symbol $V_{DET1\_\#}$ denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.
Note 2. # in the symbol $V_{DET2\_\#}$ denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.
Note 3. 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.

Table 41.35 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (2)

Conditions: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	3.6	3.8	4.0	V	Figure 41.38
	Voltage detection circuit (LVD0)	V_{DET0}	4.0	4.2	4.4		Figure 41.39
	Voltage detection circuit (LVD1)*1	V_{DET1_8}	4.59	4.77	4.95		Figure 41.40
		V_{DET1_9}	4.05	4.23	4.41		
		V_{DET1_A}	4.32	4.50	4.68		
	Voltage detection circuit (LVD2)*2	V_{DET2_8}	4.59	4.77	4.95		Figure 41.41
		V_{DET2_9}	4.05	4.23	4.41		
		V_{DET2_A}	4.32	4.50	4.68		
Internal reset time	Power-on reset (POR)	t_{POR}		9.7		ms	Figure 41.38
	Voltage detection circuit (LVD0)	t_{LVO0}		9.7			Figure 41.39
	Voltage detection circuit (LVD1)	t_{LVO1}		0.9			Figure 41.40
	Voltage detection circuit (LVD2)	t_{LVO2}		0.9			Figure 41.41
Minimum VCC down time*3		t_{VOFF}	200	—	—	µs	Figure 41.39 to Figure 41.43
Response delay time		t_{DET}			200	µs	
LVD operation stabilization time (after LVD is enabled)		$Td_{(E-A)}$			3	µs	Figure 41.40, Figure 41.41
Hysteresis width (LVD1 and LVD2)		V_{LVH}		80		mV	

Note 1. # in the symbol $V_{DET1\_\#}$ denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.
Note 2. # in the symbol $V_{DET2\_\#}$ denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.
Note 3. 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.

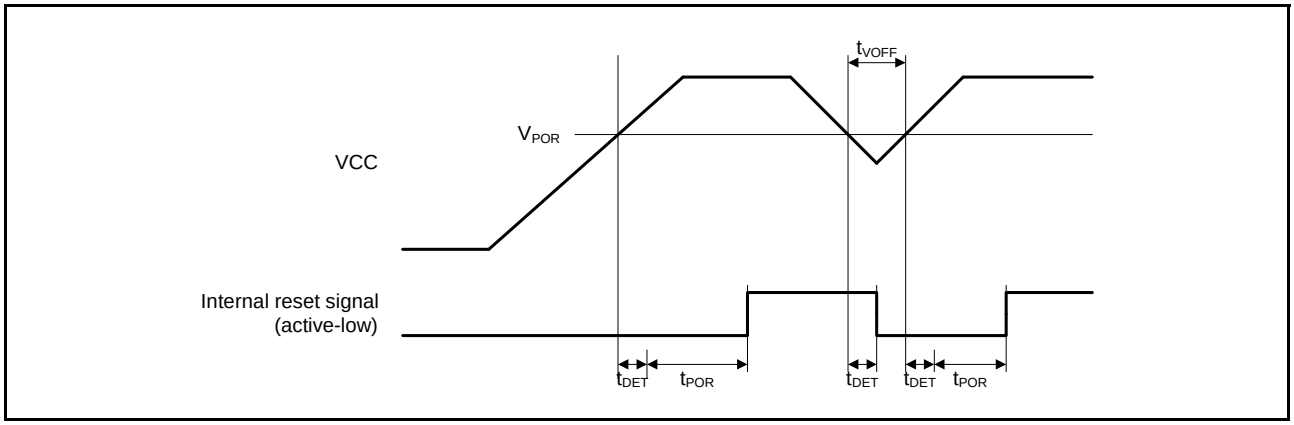


Figure 41.38 Power-on Reset Timing

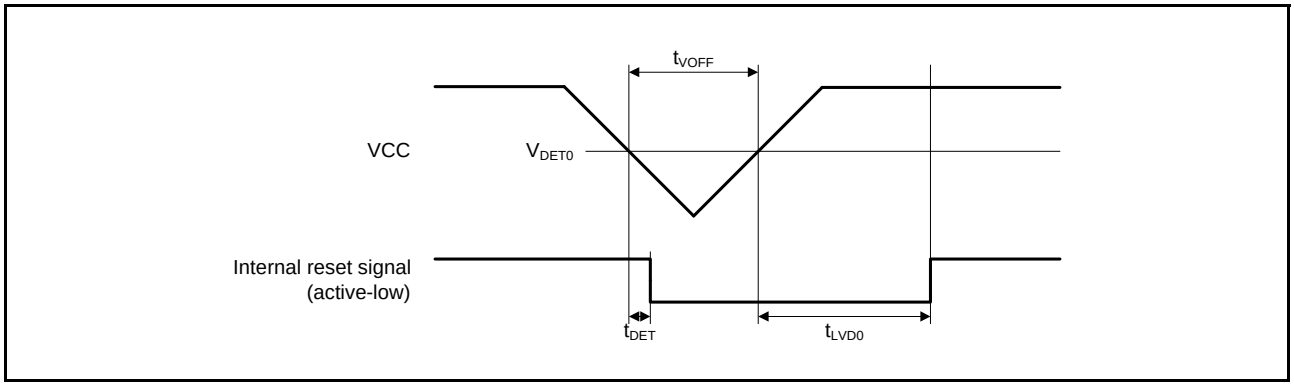


Figure 41.39 Voltage Detection Circuit Timing (V_{DET0})

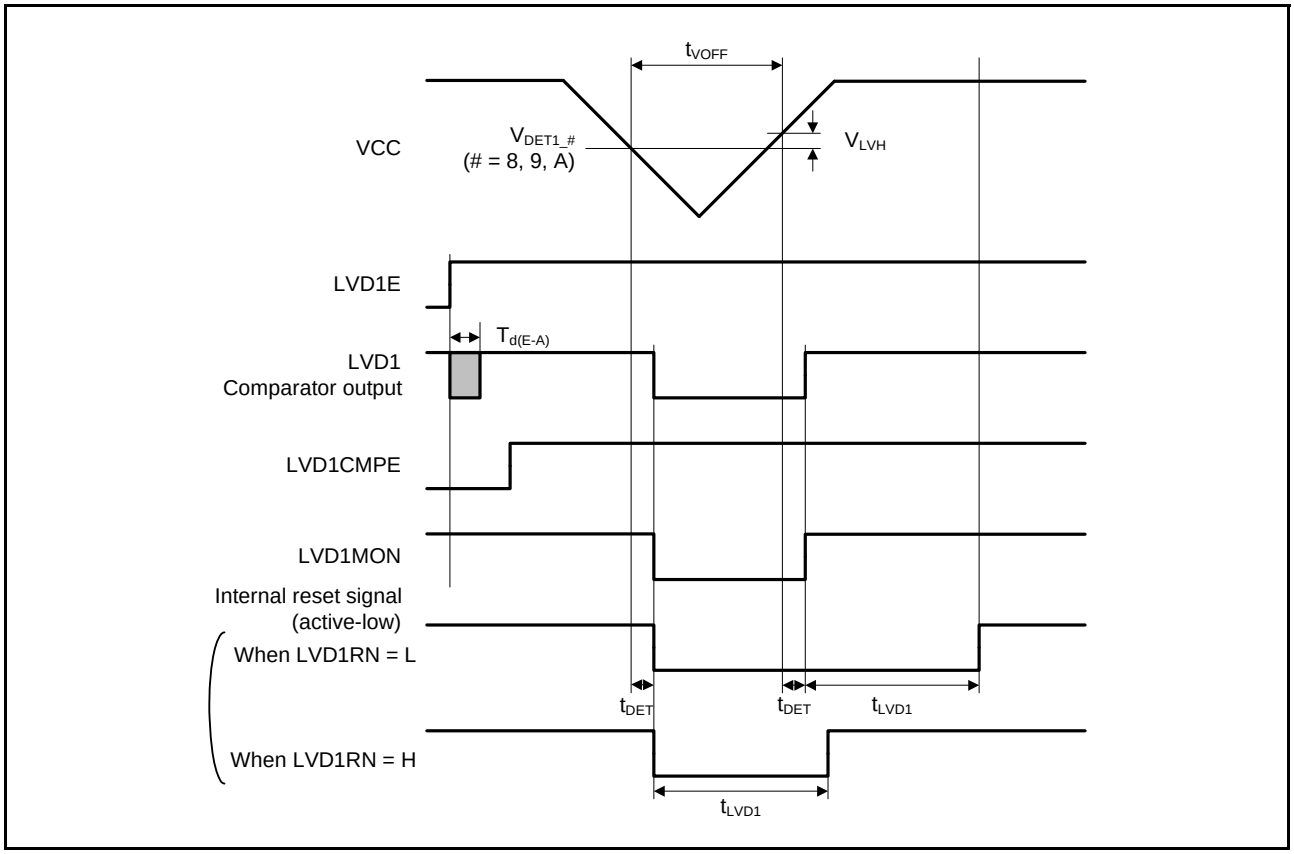


Figure 41.40 Voltage Detection Circuit Timing (V_{DET1})

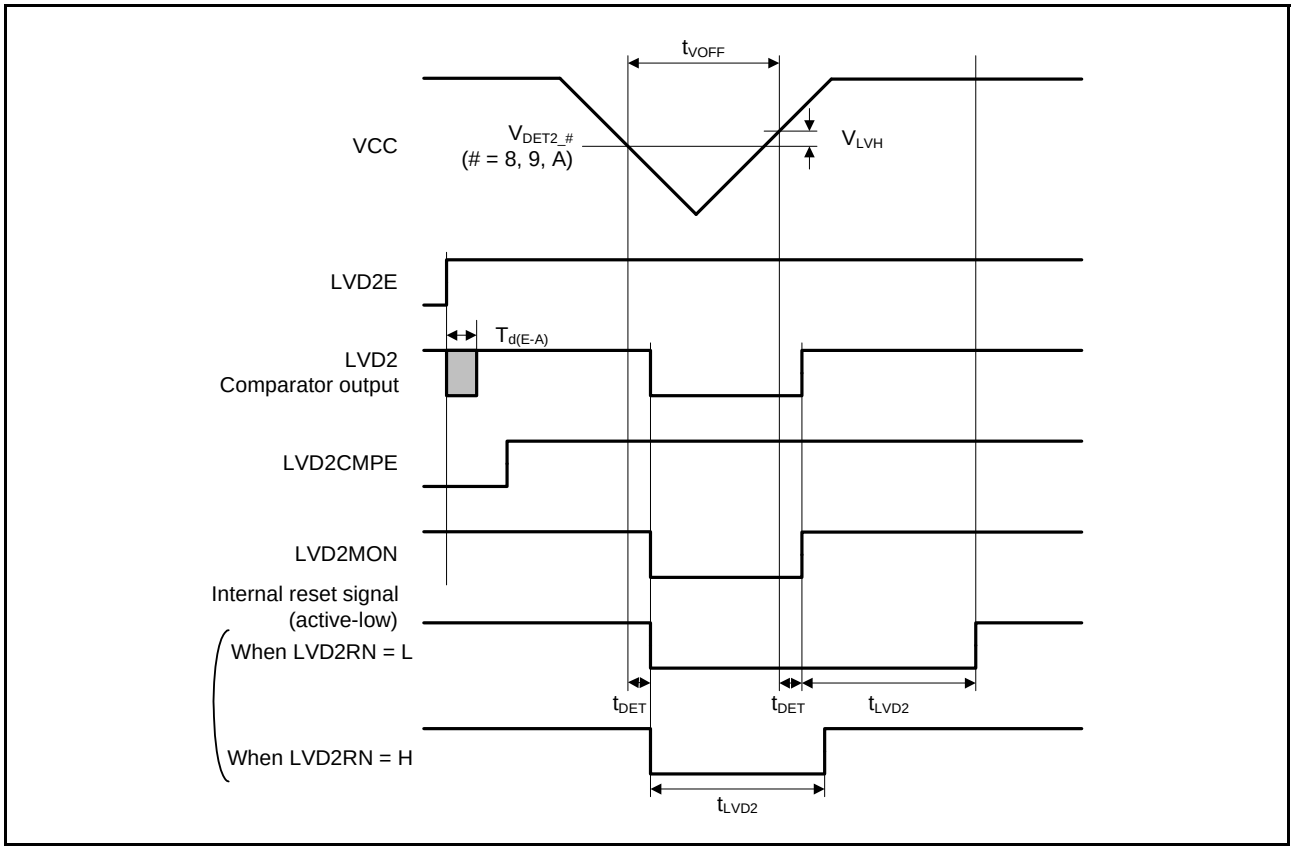


Figure 41.41 Voltage Detection Circuit Timing (V_{DET2})

41.7 Oscillation Stop Detection Timing

Table 41.36 Oscillation Stop Detection Circuit Characteristics

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V, T_a = –40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 41.42

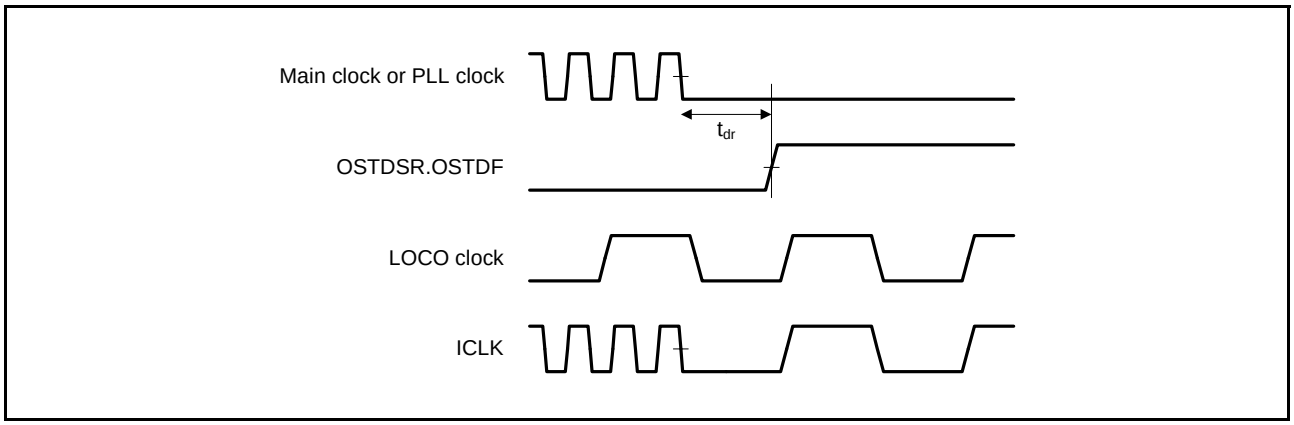


Figure 41.42 Oscillation Stop Detection Timing

41.8 ROM (Flash Memory for Code Storage) Characteristics

Table 41.37 ROM (Flash Memory for Code Storage) Characteristics (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C T_a is common to both conditions 1 and 2.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reprogramming/erasure cycle*1	N_{pec}	1000	—	—	Times	
Data hold time	t_{DRP}	30*2	—	—	Year	T_a = +85°C

Note 1. Definition of reprogram/erase cycle: The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 128-byte programming is performed 16 times for different addresses in 2-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. This result is obtained from reliability testing.

Table 41.38 ROM (Flash Memory for Code Storage) Characteristics (2)

Note: The standard values of the items with no conditions specified in the table are common to conditions 1 and 2.
Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C T_a is common to both conditions 1 and 2.

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{PEC} ≤ 100 times	128 bytes	t_{P128}	—	2.8	28	—	1	10	ms
	4 Kbytes	t_{P4K}	—	63	140	—	23	50	ms
	16 Kbytes	t_{P16K}	—	252	560	—	90	200	ms
Programming time when N_{PEC} > 100 times	128 bytes	t_{P128}	—	3.4	33.6	—	1.2	12	ms
	4 Kbytes	t_{P4K}	—	75.6	168	—	27.6	60	ms
	16 Kbytes	t_{P16K}	—	302.4	672	—	108	240	ms
Erasure time when N_{PEC} ≤ 100 times	4 Kbytes	t_{E4K}	—	50	120	—	25	60	ms
	16 Kbytes	t_{E16K}	—	200	480	—	100	240	ms
Erasure time when N_{PEC} > 100 times	4 Kbytes	t_{E4K}	—	60	144	—	30	72	ms
	16 Kbytes	t_{E16K}	—	240	576	—	120	288	ms
Suspend delay time during programming		t_{SPD}	—	—	400	—	—	120	µs
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	300	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{SESD2}	—	—	1.7	—	—	1.7	ms
Suspend delay time during erasing (in erasure priority mode)		t_{SEED}	—	—	1.7	—	—	1.7	ms
FCU reset time		t_{FCUR}	35	—	—	35	—	—	µs

41.9 E2 DataFlash (Flash Memory for Code Storage) Characteristics

Table 41.39 E2 DataFlash Characteristics (1)

Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VREFH = 2.7 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5 V, VREFH = 4.0 V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C T_a is common to both conditions 1 and 2.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reprogramming/erasure cycle*1	N_{DPEC}	100000	—	—	Times	
Data hold time	t_{DDRP}	30*2	—	—	Year	T_a = +85°C

Note 1. The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100000), erasing can be performed n times for each block. For instance, when 8-byte programming is performed 16 times for different addresses in 128-byte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. This result is obtained from reliability testing.

Table 41.40 E2 DataFlash Characteristics (2)

Note: The standard values of the items with no conditions specified in the table are common to conditions 1 and 2.
Conditions 1: VCC = AVCC0 = VREFH0 = 2.7 to 3.6V, VREFH = 2.7V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0V
Conditions 2: VCC = AVCC0 = VREFH0 = 4.0 to 5.5V, VREFH = 4.0V to AVCC0, VSS = AVSS0 = VREFL = VREFL0 = 0V
Temperature range for the programming/erasure operation: T_a = –40 to +85°C T_a is common to both conditions 1 and 2.

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{PEC} ≤ 100 times	2 bytes	t_{DP2}	—	0.7	6	—	0.25	2	ms
Programming time when N_{PEC} > 100 times	2 bytes	t_{DP2}	—	0.7	6	—	0.25	2	ms
Erasure time when N_{PEC} ≤ 100 times	32 bytes	t_{DE32}	—	4	40	—	2	20	ms
Erasure time when N_{PEC} > 100 times	32 bytes	t_{DE32}	—	7	40	—	4	20	ms
Blank check time	2 bytes	t_{DBC2}	—	—	100	—	—	30	µs
Suspend delay time during programming		t_{DSPD}	—	—	250	—	—	120	µs
First suspend delay time during erasing (in suspend priority mode)		t_{DSESD1}	—	—	250	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{DSESD2}	—	—	500	—	—	300	µs
Suspend delay time during erasing (in erasure priority mode)		t_{DSEED}	—	—	500	—	—	300	µs

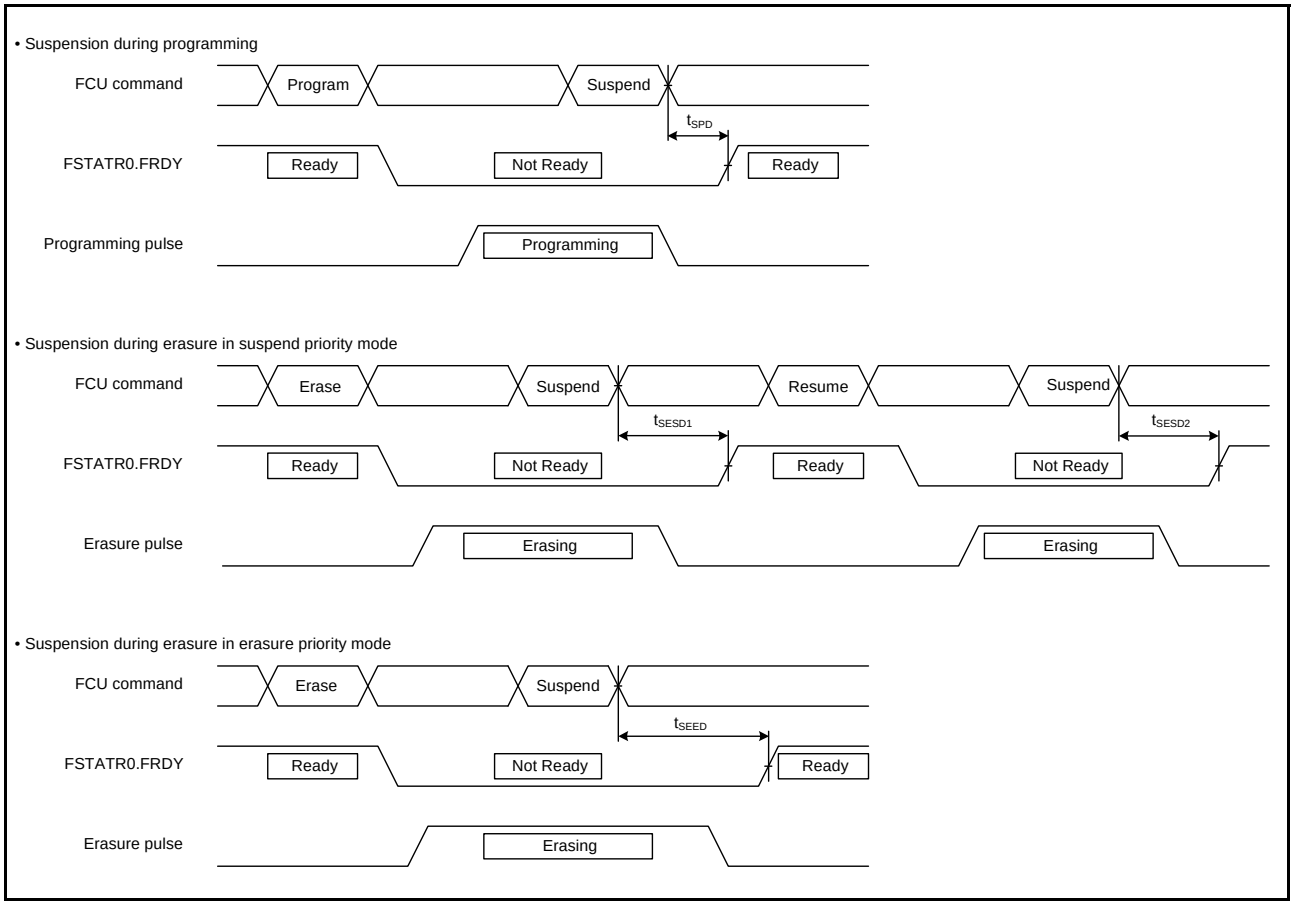


Figure 41.43 Flash Memory Program/Erase Suspend Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1/3)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Startup Mode)	
				OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
P00~P02 (PMC0/IRQ8, PMC1/ IRQ9, IRQ10)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P03 (DA0/IRQ11)	All	DA0 output (DAOE0 = 1)	Hi-Z	DA output retained		Hi-Z	Hi-Z	Hi-Z
		Other than the above (DAOE0 = 0)		Keep-O*2		Keep	Keep	
P05 (DA1)	All	DA1 output (DAOE1 = 1)	Hi-Z	DA output retained		Hi-Z	Hi-Z	Hi-Z
		Other than the above (DAOE1 = 0)		Keep-O		Keep	Keep	
P07	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P12 to P15 (IRQ2 to IRQ5)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P16, P17 (IRQ6/SCL0_DS, IRQ7/SDA0_DS)	All		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P20, P21 (IRQ8, IRQ9)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P22, P23	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P24 to P27 (CS0#, CS1#, CS2#, CS3#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	CS output		H	Hi-Z			
		Other than the above		Keep-O				
P30 to P33 (IRQ0_DS, IRQ1_DS, IRQ2_DS, IRQ3_DS)	All		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P34 (IRQ4)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P35 (NMI)	All		Hi-Z	Keep-O*2		Keep*3	Keep	Hi-Z
P40 to P47	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P50 (WR0#/WR#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended mode (EXBE = 1)			H	Hi-Z			
P51 (WR1#/BC1#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	WR0#/WR# output		H	Hi-Z			
		Other than the above		Keep-O				
P52 (RD#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended mode (EXBE = 1)			H	Hi-Z			
P53 (BCLK)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended mode (EXBE = 1)			H				
P54 (ALE/TRDATA2)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	ALE output		L	Hi-Z			
		Other than the above		Keep-O				
P55 (TRDATA3/IRQ10)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P56	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P60 to P67	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (2/3)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Startup Mode)	
				OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
P70 to P72	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P73 (IRQ12)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P74 to P77	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P80, P81 (TRDATA0, TRDATA1)	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P82, P83, P86, P87	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P90 to P93	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PA0 (A0/BC0#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		BC0# output		H	Hi-Z			
		Other than the above		Keep-O				
PA1 (A1/IRQ11)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O*2				
PA2 (A2)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O				
PA3, PA4 (A3/IRQ6_DS, A4/ IRQ5_DS)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O*2				
PA5 to PA7 (A5 to A7)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O				
PB0 (A8/IRQ12)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O*2				
PB1 (A9/IRQ4_DS)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O*2				
PB2 to PB7 (A10 to A15)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O				
PC0 (A16)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O				

Table 1.1 Port States in Each Processing State (3/3)

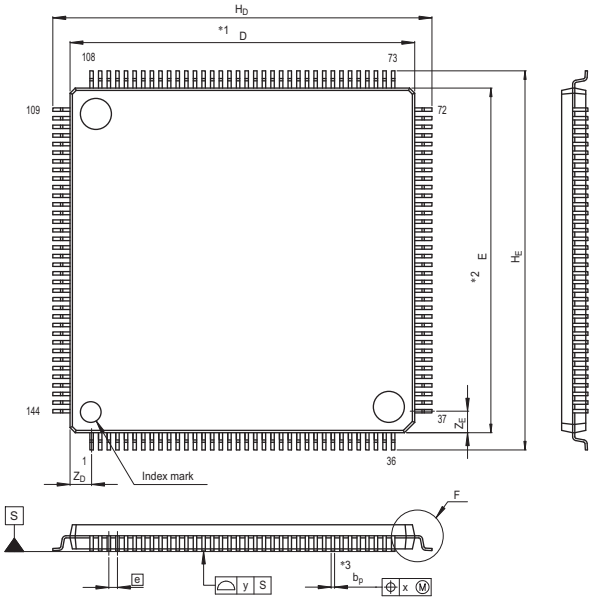
Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Startup Mode)	
				OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
PC1 (A17/IRQ12)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O*2				
PC2, PC3 (A18, A19)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		Other than the above		Keep-O				
PC4, PC5, PC6, PC7 (A20/CS3#, A21/ CS2#, A22/CS1#, A23/CS0#)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	Address output		Address output retained	Hi-Z			
		CS output		H	Hi-Z			
		Other than the above		Keep-O				
PD0 to PD7 (D0/IRQ0 to D7/ IRQ7)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended mode (EXBE = 1)			Hi-Z				
PE0, PE1 (D8, D9)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	8 bits in width of bus		Keep-O				
		16 bits in width of bus		Hi-Z				
PE2 (D10/IRQ7_DS)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	8 bits in width of bus		Keep-O*2				
		16 bits in width of bus		Hi-Z				
PE3, PE4 (D11, D12)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	8 bits in width of bus		Keep-O				
		16 bits in width of bus		Hi-Z				
PE5 to PE7 (D13/IRQ5 to D15/ IRQ7)	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended mode (EXBE = 1)	8 bits in width of bus		Keep-O*2				
		16 bits in width of bus		Hi-Z				
PF5 (IRQ4)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PH0	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PH1, PH2 (IRQ0, IRQ1)	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PH3	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PJ1 to PJ5	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PK2 to PK5	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PL0, PL1	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
PL5 (CECIO*4)	All		Hi-Z	Keep-O	Keep	Keep	Keep	Hi-Z

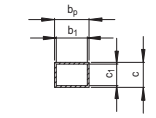
H: High-level
L: Low-level
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Keep: Pin states are retained during periods in software standby mode (pulling up and open-drain settings are also retained.)
Hi-Z: High-impedance
Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.
Note 3. Input is enabled if the pin is specified as the software standby mode canceling source.
Note 4. Input is enabled even in software standby mode when this function is used.

Appendix 2. Package Dimensions

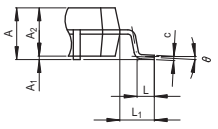
Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure A 144-Pin LQFP (PLQP0144KA-A)

REVISION HISTORY	RX634 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	Feb 25, 2015	—	First edition, issued	

RX634 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Feb 25, 2015

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

RX634 Group

Renesas Electronics Corporation

R01UH0495EJ0100