User's Manual

32

RX220 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX200 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Dec 2013

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 RX220 Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	—	R01DS0130EJ
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX220 Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family Series 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
	–	...	...	...	–	–	–	...
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	...0	... Bit	0: ..... 1: Setting prohibited *3	R/W *1
b3 to b1	—	Reserved *2	The read value is 0. The write value should always be 0.	R/W
b4	...4	... Bit	0: 1:	R
b6, b5	...[1:0]	... Bit	0 0: ..... 0 1: ..... Settings other than above are prohibited. *3	R/(W) *
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

32-MHz 32-bit RX MCUs, 49 DMIPS, up to 256-KB flash memory, 12-bit A/D, ELC, MPC, IrDA, RTC, up to 7 comms channels; incorporating functions for IEC60730 compliance

Features

■ 32-bit RX CPU core
- Max. operating frequency: 32 MHz
- Capable of 49 DMIPS in operation at 32 MHz
- Accumulator handles 64-bit results (for a single instruction) from 32- × 32-bit operations
- Multiplication and division unit handles 32- × 32-bit operations (multiplication instructions take one CPU clock cycle)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- On-chip debugging circuit

■ Low-power design and architecture
- Operation from a single 1.62-V to 5.5-V supply
- 1.62-V operation available (at up to 8 MHz)
- Three low-power modes

■ On-chip flash memory for code, no wait states
- 32-MHz operation, 31.25-ns read cycle
- No wait states for reading at full CPU speed
- Up to 256-Kbyte capacity
- User code programmable via the SCI
- Programmable at 1.62 V
- For instructions and operands

■ On-chip data flash memory
- 8 Kbytes (Number of times of reprogramming: 100,000)
- Erasing and programming impose no load on the CPU.

■ On-chip SRAM, no wait states
- Up to 16-Kbyte size capacity

■ DMA
- DMAC: Incorporates four channels
- DTC: Four transfer modes

■ 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
- Seven types of reset, including the power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions
- Frequency of external clock: Up to 20 MHz
- Frequency of the oscillator for sub-clock generation: 32.768 kHz
- On-chip low- and high-speed oscillators, dedicated on-chip low-speed oscillator for the IWDT
- Generation of a dedicated 32.768-kHz clock for the RTC
- Clock frequency accuracy measurement circuit (CAC)

■ Real-time clock
- Adjustment functions (30 seconds, leap year, and error)
- Year and month display or 32-bit second display (binary counter) is selectable

PLQP0100KB-A 14 × 14 mm, 0.5-mm pitch
PLQP0064KB-A 10 × 10 mm, 0.5-mm pitch
PLQP0048KB-A 7 × 7 mm, 0.5-mm pitch
PLQP0064GA-A 14 × 14 mm, 0.8-mm pitch

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

■ Useful functions for IEC60730 compliance
- Self-diagnostic and disconnection-detection assistance functions for the A/D converter, clock-frequency accuracy-measurement circuit, independent watchdog timer, functions to assist in RAM testing, etc.

■ Up to seven communications channels
- SCI with many useful functions (up to five channels) Asynchronous mode, clock synchronous mode, smart card interface mode
- IrDA Interface (one channel, in cooperation with the SCI5)
- I^2C bus interface: Transfer at up to 400 kbps, capable of SMBus operation (one channel)
- RSPI (one channel)

■ Up to 14 extended-function timers
- 16-bit MTU: input capture, output capture, complementary PWM output, 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 1.56 µs
- Self-diagnostic function and analog input disconnection detection assistance function

■ Analog comparator

■ General I/O ports
- 5-V tolerant, open drain, input pull-up, switching of driving ability

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

■ Operating temp. range
- –40°C to +85°C
- –40°C to +105°C

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 gives a comparison of the functions of products in different packages.
Table 1.1 is for products with the greatest number of functions, so numbers of peripheral modules and channels will differ in accord with the package. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1 / 3)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 32 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: Eight 32-bit registers Accumulator: One 64-bit register • Basic instructions: 73 • 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 • On-chip divider: 32 / 32 → 32 bits • Barrel shifter: 32 bits
Memory	ROM	• Capacity: 32 K/64 K/128 K/256 Kbytes • 32 MHz, no-wait memory access • On-board programming: 3 types
	RAM	• Capacity: 4 K/8 K/16 Kbytes • 32 MHz, no-wait memory access
	E2 DataFlash	E2 DataFlash capacity: 8 Kbytes
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	• Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, 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 (PCLK), and flashIF clock (FCLK) The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 32 MHz (at max.) Peripheral modules run in synchronization with the peripheral module clock (PCLK): 32 MHz (at max.) The flash peripheral circuit runs in synchronization with the flash peripheral clock (FCLK): 32 MHz (at max.)
Reset		RES# pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, and software reset
Voltage detection	Voltage detection circuit (LVDAa)	• When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated. Voltage detection circuit 0 is capable of selecting the detection voltage from 4 levels Voltage detection circuit 1 is capable of selecting the detection voltage from 16 levels Voltage detection circuit 2 is capable of selecting the detection voltage from 16 levels
Low power consumption	Low power consumption facilities	• Module stop function • Three low power consumption modes Sleep mode, all-module clock stop mode, and software standby mode
	Function for lower operating power consumption	• Four operating power control modes Middle-speed operating mode 1A, middle-speed operating mode 1B, low-speed operating mode 1, low-speed operating mode 2
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 106 • External interrupts: 9 (NMI, IRQ0 to IRQ7 pins) • Non-maskable interrupts: 5 (the NMI pin, oscillation stop detection interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and IWDT interrupt) • 16 levels specifiable for the order of priority

Table 1.1 Outline of Specifications (2 / 3)

Classification	Module/Function	Description
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	100-pin/64-pin/48-pin • I/O pin: 84/48/34 • Input: 1/1/1 • Pull-up resistors: 84/48/34 • Open-drain outputs: 35/26/20 • 5-V tolerance: 4/2/2 • 8-bit port switching function: Not supported/supported/supported
Event link controller (ELC)		• Event signals of 46 types can be directly connected to the module • Operations of timer modules are selectable at event input • Capable of event link operation for port B
Multi-function pin controller (MPC)		• Capable of selecting input/output function from multiple pins
Timers	Multi-function timer pulse unit 2 (MTU2a)	• (16 bits × 6 channels) × 1 unit • Time bases for the six 16-bit timer channels can be provided via up to 16 pulse-input/output lines and three pulse-input lines • 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
	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
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	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
	Realtime clock (RTCc)	• Clock source: Sub-clock • Time count or 32-bit binary count in second units basis selectable • Time/calendar • Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt

Table 1.1 Outline of Specifications (3 / 3)

Classification	Module/Function	Description
Communication function	Serial communications interfaces (SCIe, SCIf)	• 5 channels (channel 1, 5, 6, and 9: SCIe, channel 12: SCIf) (including one channel for IrDA) • 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) • Detection of a start bit in asynchronous mode: Low level or falling edge is selectable (SCIe/SCIf)
	IrDA interface (IRDA)	• 1 channel (SCI5 is used) • Supports encoding/decoding the waveforms conforming to the IrDA specification version 1.0
	I^2C bus interface (RIIC)	• 1 channel • Communications formats: I^2C bus format/SMBus format • Master/slave selectable • Supports the fast mode
	Serial peripheral interface (RSPI)	• 1 channel • Transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) • Capable of handling serial transfer as a master or slave • Data formats • 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
12-bit A/D converter (S12ADb)		• 12 bits (16 channels × 1 unit) • 12-bit resolution • Minimum conversion time: 1.56 μs per channel (in operation with ADCLK at 32 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
CRC calculator (CRC)		• CRC code generation for any desired 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.
Comparator A (CMPA)		• 2 channels • Comparison of reference voltage and analog input voltage
Data Operation Circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Power supply voltage/Operating frequency		VCC = 1.62 to 2.7 V: 8 MHz, VCC = 2.7 to 5.5 V: 32 MHz
Operating temperature		D version: –40 to +85°C, G version: –40 to +105°C*1
Package		100-pin LQFP (PLQP0100KB-A) 64-pin LQFP (PLQP0064KB-A) 64-pin LQFP (PLQP0064GA-A) 48-pin LQFP (PLQP0048KB-A)

Note 1. Please contact Renesas Electronics sales office for derating of operation under Ta = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

Table 1.2 Comparison of Functions for Different Packages

Module/Functions		RX220 Group		
		100 Pins	**64 Pins**	**48 Pins**
Interrupt	External interrupts	NMI, IRQ0 to IRQ7	NMI, IRQ0 to IRQ2, IRQ4 to IRQ7	NMI, IRQ0, IRQ1, IRQ4 to IRQ7
DMA	DMA controller	4 channels (DMAC0 to DMAC3)		
	Data transfer controller	Supported		
Timers	Multi-function timer pulse unit 2	6 channels (MTU0 to MTU5)		
	Port output enable 2	POE0# to POE3#, POE8#		
	8-bit timer	2 channels × 2 units		
	Compare match timer	2 channels × 2 units		
	Realtime clock	Supported		Not supported
	Independent watchdog timer	Supported		
Communication function	Serial communications interface (SCIe)	4 channels (SCI1, 5, 6, 9) (including one channel for IrDA)		3 channels (SCI1, 5, 6) (including one channel for IrDA)
	Serial communications interface (SCIf)	1 channel (SCI12)		
	I^2C bus interface	1 channel		
	Serial peripheral interface	1 channel		
12-bit A/D converter		16 channels (AN000 to AN015)	12 channels (AN000 to AN004, AN006, AN008 to AN013)	8 channels (AN000, AN003, AN004, AN006, AN009 to AN012)
CRC calculator		Supported		
Event link controller		Supported		
Comparator A		2 channels		
8-bit port switching function		Not supported in 100-pin packages	Supported in 64-pin packages Switches PB6 to PC0 and PB7 to PC1	Supported in 48-pin packages Switches PB0 to PC0, PB1 to PC1, PB3 to PC2, and PB5 to PC3
Package		100-pin LQFP	64-pin LQFP	48-pin LQFP

1.2 List of Products

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

Table 1.3 List of Products

Group	Part No.	Package	ROM Capacity	RAM Capacity	Operating Frequency (Max.)	Operating temperature
RX220	R5F52206BDFP	PLQP0100KB-A	256 Kbytes	16 Kbytes	32 MHz	–40 to +85°C
	R5F52206BDFM	PLQP0064KB-A				
	R5F52206BDFK	PLQP0064GA-A				
	R5F52206BDFL	PLQP0048KB-A				
	R5F52205BDFP	PLQP0100KB-A	128 Kbytes	8 Kbytes		
	R5F52205BDFM	PLQP0064KB-A				
	R5F52205BDFK	PLQP0064GA-A				
	R5F52205BDFL	PLQP0048KB-A				
	R5F52203BDFP	PLQP0100KB-A	64 Kbytes			
	R5F52203BDFM	PLQP0064KB-A				
	R5F52203BDFK	PLQP0064GA-A				
	R5F52203BDFL	PLQP0048KB-A				
	R5F52201BDFM	PLQP0064KB-A	32 Kbytes	4Kbytes		
	R5F52201BDFK	PLQP0064GA-A				
	R5F52201BDFL	PLQP0048KB-A				
	R5F52206BGFP	PLQP0100KB-A	256 Kbytes	16 Kbytes	32 MHz	–40 to +105°C
	R5F52206BGFM	PLQP0064KB-A				
	R5F52206BGFK	PLQP0064GA-A				
	R5F52206BGFL	PLQP0048KB-A				
	R5F52205BGFP	PLQP0100KB-A	128 Kbytes	8 Kbytes		
	R5F52205BGFM	PLQP0064KB-A				
	R5F52205BGFK	PLQP0064GA-A				
	R5F52205BGFL	PLQP0048KB-A				
	R5F52203BGFP	PLQP0100KB-A	64 Kbytes			
	R5F52203BGFM	PLQP0064KB-A				
	R5F52203BGFK	PLQP0064GA-A				
	R5F52203BGFL	PLQP0048KB-A				
	R5F52201BGFM	PLQP0064KB-A	32 Kbytes	4Kbytes		
	R5F52201BGFK	PLQP0064GA-A				
	R5F52201BGFL	PLQP0048KB-A				

Note: • Please contact Renesas Electronics sales office for derating of operation under Ta = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

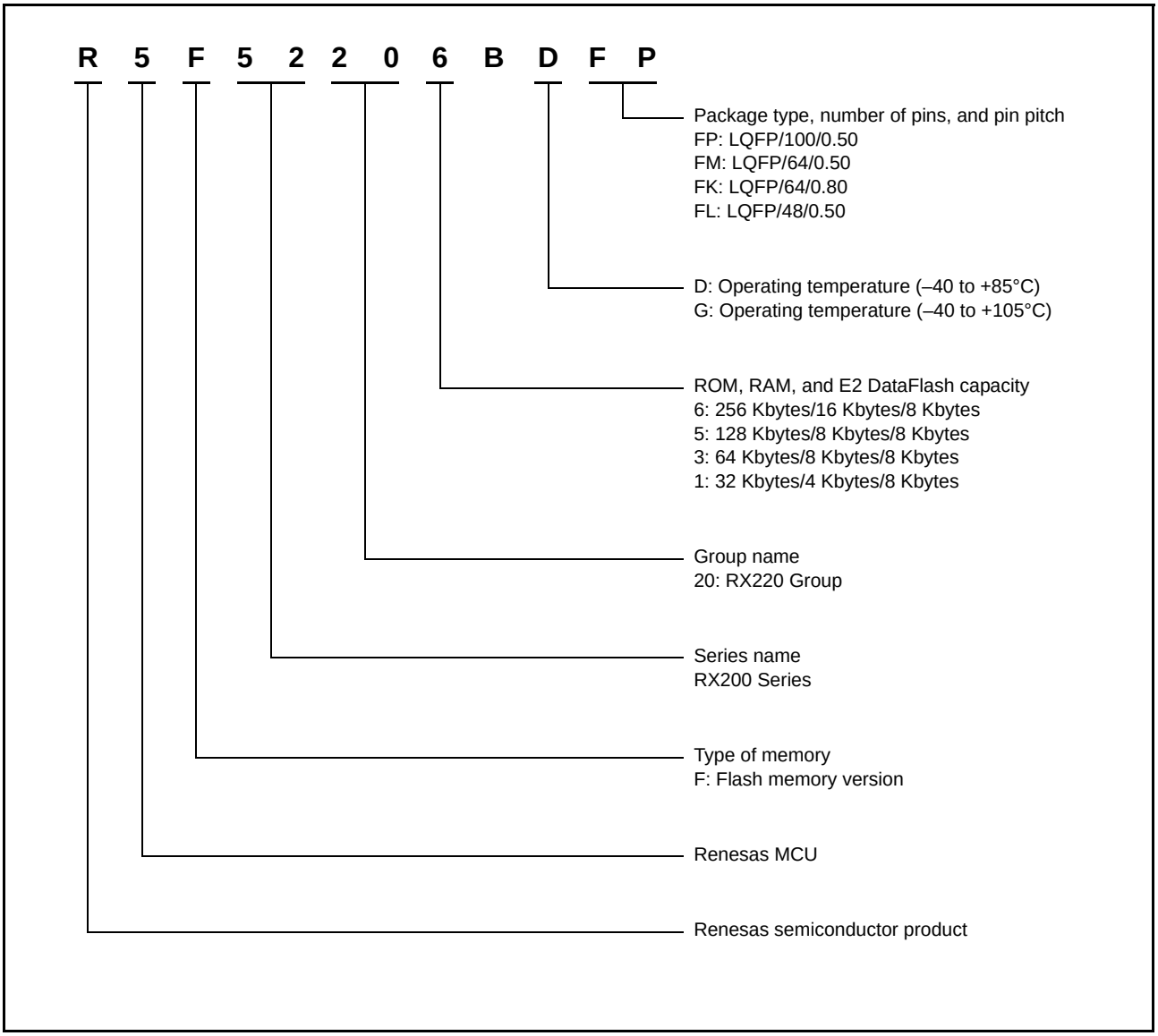
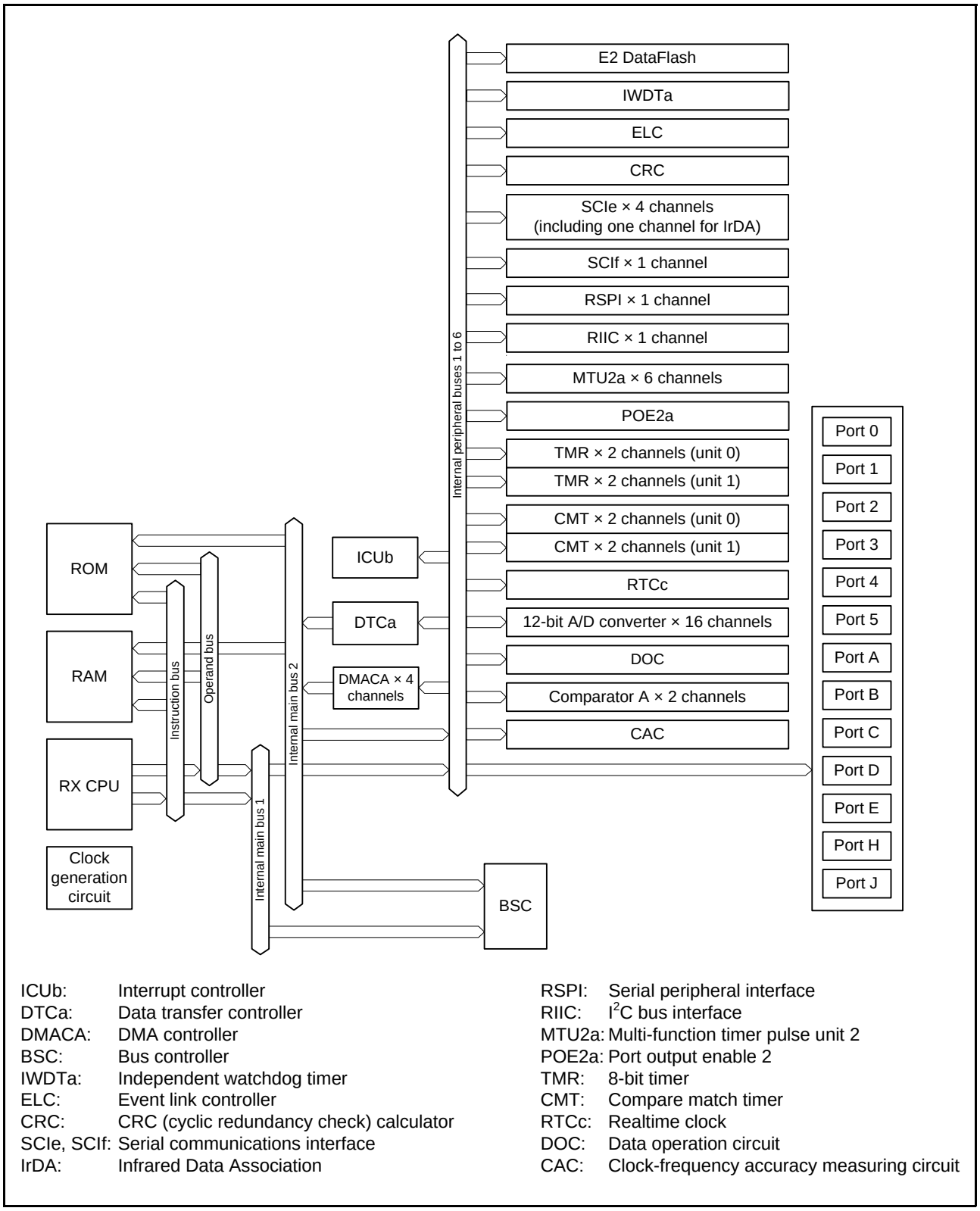


Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICUb: Interrupt controller
DTCa: Data transfer controller
DMACA: DMA controller
BSC: Bus controller
IWDTa: Independent watchdog timer
ELC: Event link controller
CRC: CRC (cyclic redundancy check) calculator
SCIe, SCIf: Serial communications interface
IrDA: Infrared Data Association
RSPI: Serial peripheral interface
RIIC: I^2C bus interface
MTU2a: Multi-function timer pulse unit 2
POE2a: Port output enable 2
TMR: 8-bit timer
CMT: Compare match timer
RTCc: Realtime clock
DOC: Data operation circuit
CAC: Clock-frequency accuracy measuring circuit

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1 / 3)

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	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	Input	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	
	XCIN	Input	Input/output pins for the sub-clock generation circuit. Connect a crystal resonator between XCIN and XCOUT.
	XCOUT	Output	
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 signal input pin. This LSI enters the reset state when this signal goes low.
CAC	CACREF	Input	Input pin for the measuring circuit for clock frequency precision.
On-chip emulator	FINED	I/O	FINE interface pin.
Interrupt	NMI	Input	Non-maskable interrupt request pin.
	IRQ0 to IRQ7	Input	Interrupt request pins.
Multi-function timer pulse unit	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 external clock.
Port output enable	POE0# to POE3#, POE8#	Input	Input pins for request signals to place the MTU pins in the high impedance state.
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.
Realtime clock	RTCOUT	Output	Output pin for 1-Hz clock.

Table 1.4 Pin Functions (2 / 3)

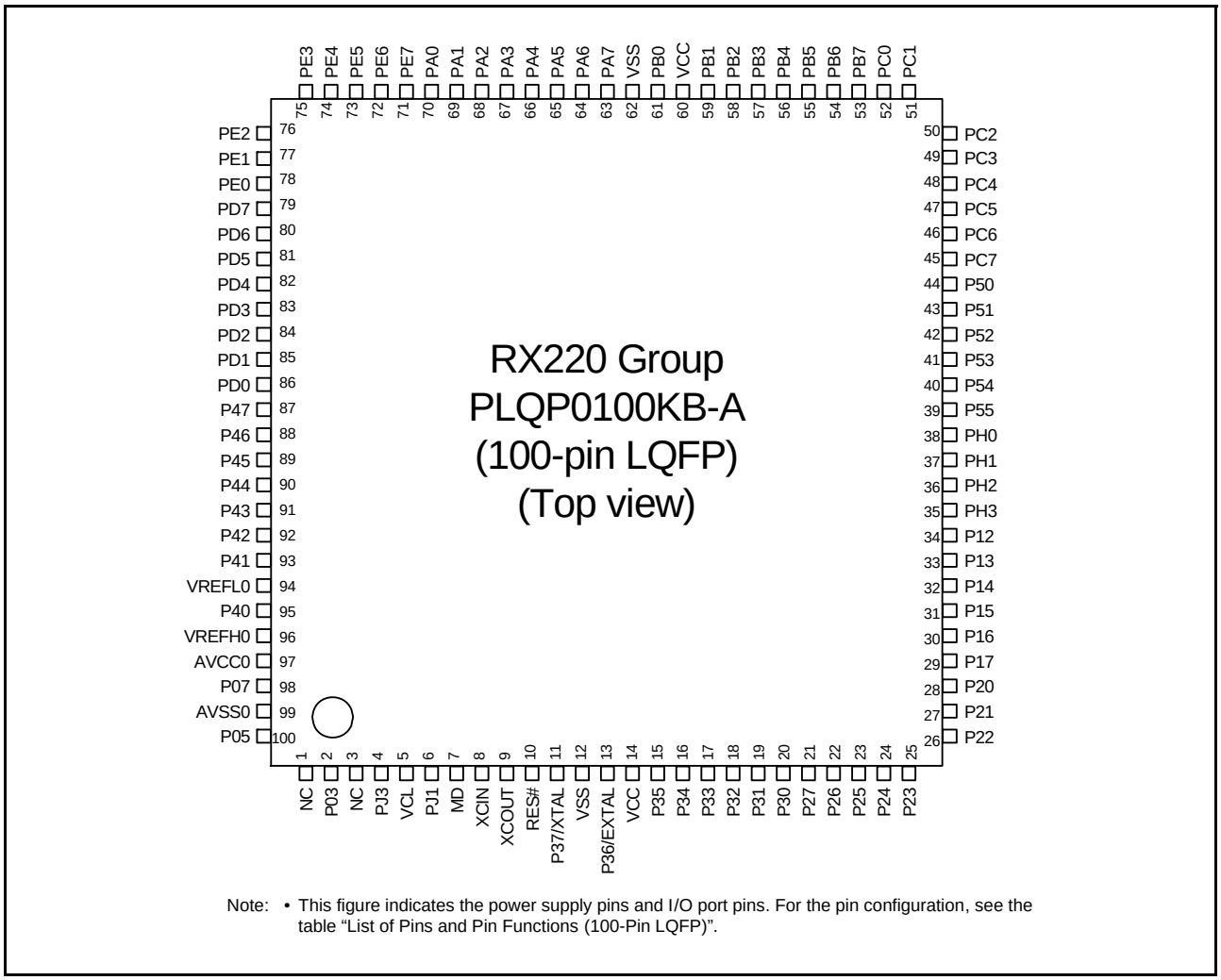
Classifications	Pin Name	I/O	Description
Serial communications interface (SCIe)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5, SCK6, SCK9	I/O	Input/output pins for clock
	RXD1, RXD5, RXD6, RXD9	Input	Input pins for received data
	TXD1, TXD5, TXD6, TXD9	Output	Output pins for transmitted data
	CTS1#, CTS5#, CTS6#, CTS9#	Input	Input pins for controlling the start of transmission and reception
	RTS1#, RTS5#, RTS6#, RTS9#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL1, SSCL5, SSCL6, SSCL9	I/O	Input/output pins for the I^2C clock
	SSDA1, SSDA5, SSDA6, SSDA9	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK1, SCK5, SCK6, SCK9	I/O	Input/output pins for the clock
	SMISO1, SMISO5, SMISO6, SMISO9	I/O	Input/output pins for slave transmission of data
	SMOSI1, SMOSI5, SMOSI6, SMOSI9	I/O	Input/output pins for master transmission of data
	SS1#, SS5#, SS6#, SS9#	Input	Chip-select input pins
	• IrDA Interface		
	IRTXD5	Output	Data output pin in the IrDA format
	IRRXD5	Input	Data input pin in the IrDA format
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 SCIf
	TXDX12	Output	Output pin for data transmission by SCIf
	SIOX12	I/O	Input/output pin for data reception or transmission by SCIf
I^2C bus interface	SCL	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open-drain output.
	SDA	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open-drain output.

Table 1.4 Pin Functions (3 / 3)

Classifications	Pin Name	I/O	Description
Serial peripheral interface	RSPCKA	I/O	Clock input/output pin for the RSPI.
	MOSIA	I/O	Input or output data output from the master for the RSPI.
	MISOA	I/O	Input or output data output from the slave for the RSPI.
	SSLA0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1 to SSLA3	Output	Output pins to select the slave for the RSPI.
12-bit A/D converter	AN000 to AN015	Input	Input pin 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.
Comparator A	CMPA1	Input	Input analog pin for the comparator A1.
	CMPA2	Input	Input analog pin for the comparator A2.
	CVREFA	Input	Input pin for the comparator reference voltage.
Analog power supply	AVCC0	Input	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	Input	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	Input	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	Input	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.
I/O ports	P03, P05, P07	I/O	3-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 P37	I/O	8-bit input/output pins. (P35 input pin)
	P40 to P47	I/O	8-bit input/output pins.
	P50 to P55	I/O	6-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.
	PH0 to PH3	I/O	4-bit input/output pins.
	PJ1, PJ3	I/O	2-bit input/output pins.

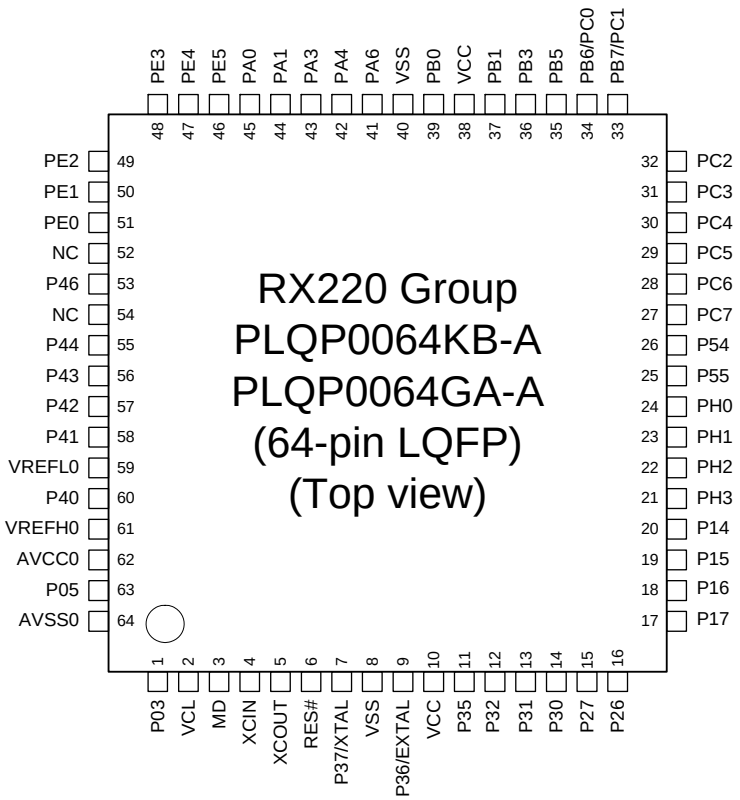
1.5 Pin Assignments

Figure 1.3 to Figure 1.5 show the pin assignments. Table 1.5 to Table 1.7 show the lists of pins and pin functions.



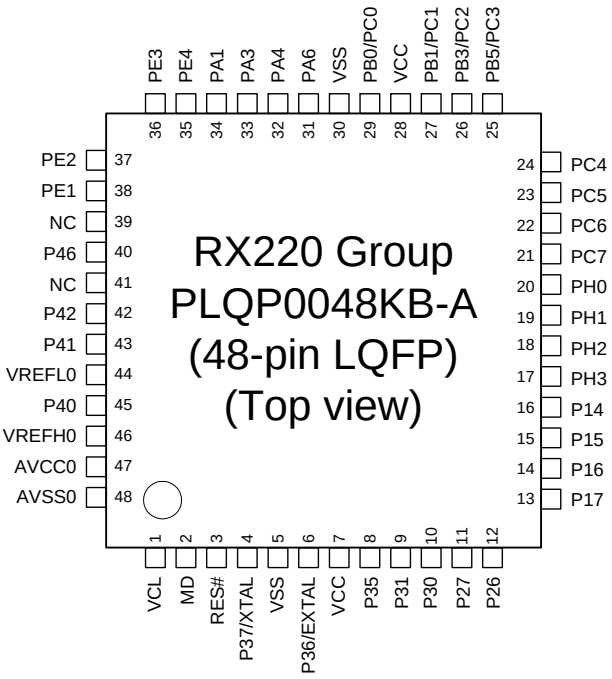
Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see the table “List of Pins and Pin Functions (100-Pin LQFP)”.

Figure 1.3 Pin Assignments of the 100-Pin LQFP



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 (64-Pin LQFP)”.

Figure 1.4 Pin Assignments of the 64-Pin LQFP



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 (48-Pin LQFP)”.

Figure 1.5 Pin Assignments of the 48-Pin LQFP

Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (1 / 3)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communications (SCIe, SCIf, RSPI, RIIC)	Others
1	NC (Non-Connection)				
2		P03			
3	NC (Non-Connection)				
4		PJ3	MTIOC3C	CTS6#/RTS6#/SS6#	
5	VCL				
6		PJ1	MTIOC3A		
7	MD				FINED
8	XCIN				
9	XCOUT				
10	RES#				
11	XTAL	P37			
12	VSS				
13	EXTAL	P36			
14	VCC				
15		P35			NMI
16		P34	MTIOC0A/TMCI3/POE2#	SCK6	IRQ4
17		P33	MTIOC0D/TMRI3/POE3#	RXD6/SMISO6/SSCL6	IRQ3
18		P32	MTIOC0C/TMO3	TXD6/SMOSI6/SSDA6	IRQ2/RTCOUT
19		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	IRQ1
20		P30	MTIOC4B/TMRI3/POE8#	RXD1/SMISO1/SSCL1	IRQ0
21		P27	MTIOC2B/TMCI3	SCK1	
22		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1	
23		P25	MTIOC4C/MTCLKB		ADTRG0#
24		P24	MTIOC4A/MTCLKA/TMRI1		
25		P23	MTIOC3D/MTCLKD		
26		P22	MTIOC3B/MTCLKC/TMO0		
27		P21	MTIOC1B/TMCI0		
28		P20	MTIOC1A/TMRI0		
29		P17	MTIOC3A/MTIOC3B/TMO1/ POE8#	SCK1/MISOA/SDA	IRQ7
30		P16	MTIOC3C/MTIOC3D/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/ SCL	IRQ6/RTCOUT/ADTRG0#
31		P15	MTIOC0B/MTCLKB/TMCI2	RXD1/SMISO1/SSCL1	IRQ5
32		P14	MTIOC3A/MTCLKA/TMRI2	CTS1#/RTS1#/SS1#	IRQ4
33		P13	MTIOC0B/TMO3	SDA	IRQ3
34		P12	TMCI1	SCL	IRQ2
35		PH3	TMCI0		
36		PH2	TMRI0		IRQ1
37		PH1	TMO0		IRQ0
38		PH0			CACREF
39		P55	MTIOC4D/TMO3		
40		P54	MTIOC4B/TMCI1		
41		P53			
42		P52			
43		P51			
44		P50			
45		PC7	MTIOC3A/TMO2/MTCLKB	MISOA	CACREF
46		PC6	MTIOC3C/MTCLKA/TMCI2	MOSIA	
47		PC5	MTIOC3B/MTCLKD/TMRI2	RSPCKA	

Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (2 / 3)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communications (SCIe, SCIf, RSPI, RIIC)	Others
48		PC4	MTIOC3D/MTCLKC/TMCI1/ POE0#	SCK5/SSLA0	
49		PC3	MTIOC4D	TXD5/SMOSI5/SSDA5/IRTXD5	
50		PC2	MTIOC4B	RXD5/SMISO5/SSCL5/IRRXD5/ SSLA3	
51		PC1	MTIOC3A	SCK5/SSLA2	
52		PC0	MTIOC3C	CTS5#/RTS5#/SS5#/SSLA1	
53		PB7	MTIOC3B	TXD9/SMOSI9/SSDA9	
54		PB6	MTIOC3D	RXD9/SMISO9/SSCL9	
55		PB5	MTIOC2A/MTIOC1B/TMRI1/ POE1#	SCK9	
56		PB4		CTS9#/RTS9#/SS9#	
57		PB3	MTIOC0A/MTIOC4A/TMO0/ POE3#	SCK6	
58		PB2		CTS6#/RTS6#/SS6#	
59		PB1	MTIOC0C/MTIOC4C/TMCI0	TXD6/SMOSI6/SSDA6	IRQ4
60	VCC				
61		PB0	MTIC5W	RXD6/SMISO6/SSCL6/RSPCKA	
62	VSS				
63		PA7		MISOA	
64		PA6	MTIC5V/MTCLKB/TMCI3/ POE2#	CTS5#/RTS5#/SS5#/MOSIA	
65		PA5		RSPCKA	
66		PA4	MTIC5U/MTCLKA/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/ SSLA0	IRQ5
67		PA3	MTIOC0D/MTCLKD	RXD5/SMISO5/SSCL5/IRRXD5	IRQ6
68		PA2		RXD5/SMISO5/SSCL5/SSLA3/ IRRXD5	
69		PA1	MTIOC0B/MTCLKC	SCK5/SSLA2	CVREFA
70		PA0	MTIOC4A	SSLA1	CACREF
71		PE7			IRQ7/AN015
72		PE6			IRQ6/AN014
73		PE5	MTIOC4C/MTIOC2B		IRQ5/AN013
74		PE4	MTIOC4D/MTIOC1A		AN012/CMPA2
75		PE3	MTIOC4B/POE8#	CTS12#/RTS12#/SS12#	AN011/CMPA1
76		PE2	MTIOC4A	RXD12/RXDX12/SMISO12/ SSCL12	IRQ7/AN010
77		PE1	MTIOC4C	TXD12/TXDX12/SIOX12/ SMOSI12/SSDA12	AN009
78		PE0		SCK12	AN008
79		PD7	MTIC5U/POE0#		IRQ7
80		PD6	MTIC5V/POE1#		IRQ6
81		PD5	MTIC5W/POE2#		IRQ5
82		PD4	POE3#		IRQ4
83		PD3	POE8#		IRQ3
84		PD2	MTIOC4D		IRQ2
85		PD1	MTIOC4B		IRQ1
86		PD0			IRQ0
87		P47			AN007
88		P46			AN006
89		P45			AN005
90		P44			AN004

Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (3 / 3)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communications (SCIe, SCIf, RSPI, RIIC)	Others
91		P43			AN003
92		P42			AN002
93		P41			AN001
94	VREFL0				
95		P40			AN000
96	VREFH0				
97	AVCC0				
98		P07			ADTRG0#
99	AVSS0				
100		P05			

Table 1.6 List of Pins and Pin Functions (64-Pin LQFP) (1 / 2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communication (SCIe, SCIf, RSPI, RIIC)	Others
1		P03			
2	VCL				
3	MD				FINED
4	XCIN				
5	XCOUT				
6	RES#				
7	XTAL	P37			
8	VSS				
9	EXTAL	P36			
10	VCC				
11		P35			NMI
12		P32	MTIOC0C/TMO3	TXD6/SMOSI6/SSDA6	IRQ2/RTCOUT
13		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	IRQ1
14		P30	MTIOC4B/TMRI3/POE8#	RXD1/SMISO1/SSCL1	IRQ0
15		P27	MTIOC2B/TMCI3	SCK1	
16		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1	
17		P17	MTIOC3A/MTIOC3B/TMO1/ POE8#	SCK1/MISOA/SDA	IRQ7
18		P16	MTIOC3C/MTIOC3D/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/ SCL	IRQ6/RTCOUT/ADTRG0#
19		P15	MTIOC0B/MTCLKB/TMCI2	RXD1/SMISO1/SSCL1	IRQ5
20		P14	MTIOC3A/MTCLKA/TMRI2	CTS1#/RTS1#/SS1#	IRQ4
21		PH3	TMCI0		
22		PH2	TMRI0		IRQ1
23		PH1	TMO0		IRQ0
24		PH0			CACREF
25		P55	MTIOC4D/TMO3		
26		P54	MTIOC4B/TMCI1		
27		PC7	MTIOC3A/TMO2/MTCLKB	MISOA	CACREF
28		PC6	MTIOC3C/MTCLKA/TMCI2	MOSIA	
29		PC5	MTIOC3B/MTCLKD/TMRI2	RSPCKA	
30		PC4	MTIOC3D/MTCLKC/TMCI1/ POE0#	SCK5/SSLA0	
31		PC3	MTIOC4D	TXD5/SMOSI5/SSDA5/IRTXD5	
32		PC2	MTIOC4B	RXD5/SMISO5/SSCL5/IRRXD5/ SSLA3	
33		PB7/PC1	MTIOC3B	TXD9/SMOSI9/SSDA9	
34		PB6/PC0	MTIOC3D	RXD9/SMISO9/SSCL9	
35		PB5	MTIOC2A/MTIOC1B/TMRI1/ POE1#	SCK9	
36		PB3	MTIOC0A/MTIOC4A/TMO0/ POE3#	SCK6	
37		PB1	MTIOC0C/MTIOC4C/TMCI0	TXD6/SMOSI6/SSDA6	IRQ4
38	VCC				
39		PB0	MTIC5W	RXD6/SMISO6/SSCL6/RSPCKA	
40	VSS				
41		PA6	MTIC5V/MTCLKB/TMCI3/ POE2#	CTS5#/RTS5#/SS5#/MOSIA	
42		PA4	MTIC5U/MTCLKA/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/ SSLA0	IRQ5
43		PA3	MTIOC0D/MTCLKD	RXD5/SMISO5/SSCL5/IRRXD5	IRQ6

Table 1.6 List of Pins and Pin Functions (64-Pin LQFP) (2 / 2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communication (SCIe, SCIf, RSPI, RIIC)	Others
44		PA1	MTIOC0B/MTCLKC	SCK5/SSLA2	CVREFA
45		PA0	MTIOC4A	SSLA1	CACREF
46		PE5	MTIOC4C/MTIOC2B		IRQ5/AN013
47		PE4	MTIOC4D/MTIOC1A		AN012/CMPA2
48		PE3	MTIOC4B/POE8#	CTS12#/RTS12#/SS12#	AN011/CMPA1
49		PE2	MTIOC4A	RXD12/RXDX12/SMISO12/ SSCL12	IRQ7/AN010
50		PE1	MTIOC4C	TXD12/TXDX12/SIOX12/ SMOSI12/SSDA12	AN009
51		PE0		SCK12	AN008
52	NC (Non-Connection)				
53		P46			AN006
54	NC (Non-Connection)				
55		P44			AN004
56		P43			AN003
57		P42			AN002
58		P41			AN001
59	VREFL0				
60		P40			AN000
61	VREFH0				
62	AVCC0				
63		P05			
64	AVSS0				

Table 1.7 List of Pins and Pin Functions (48-Pin LQFP) (1 / 2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communication (SCIe, SCIf, RSPI, RIIC)	Others
1	VCL				
2	MD				FINED
3	RES#				
4	XTAL	P37			
5	VSS				
6	EXTAL	P36			
7	VCC				
8		P35			NMI
9		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	IRQ1
10		P30	MTIOC4B/TMRI3/POE8#	RXD1/SMISO1/SSCL1	IRQ0
11		P27	MTIOC2B/TMCI3	SCK1	
12		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1	
13		P17	MTIOC3A/MTIOC3B/TMO1/ POE8#	SCK1/MISOA/SDA	IRQ7
14		P16	MTIOC3C/MTIOC3D/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/ SCL	IRQ6/ADTRG0#
15		P15	MTIOC0B/MTCLKB/TMCI2	RXD1/SMISO1/SSCL1	IRQ5
16		P14	MTIOC3A/MTCLKA/TMRI2	CTS1#/RTS1#/SS1#	IRQ4
17		PH3	TMCI0		
18		PH2	TMRI0		IRQ1
19		PH1	TMO0		IRQ0
20		PH0			CACREF
21		PC7	MTIOC3A/TMO2/MTCLKB	MISOA	CACREF
22		PC6	MTIOC3C/MTCLKA/TMCI2	MOSIA	
23		PC5	MTIOC3B/MTCLKD/TMRI2	RSPCKA	
24		PC4	MTIOC3D/MTCLKC/TMCI1/ POE0#	SCK5/SSLA0	
25		PB5/PC3	MTIOC2A/MTIOC1B/TMRI1/ POE1#		
26		PB3/PC2	MTIOC0A/MTIOC4A/TMO0/ POE3#	SCK6	
27		PB1/PC1	MTIOC0C/MTIOC4C/TMCI0	TXD6/SMOSI6/SSDA6	IRQ4
28	VCC				
29		PB0/PC0	MTIC5W	RXD6/SMISO6/SSCL6/RSPCKA	
30	VSS				
31		PA6	MTIC5V/MTCLKB/TMCI3/ POE2#	CTS5#/RTS5#/SS5#/MOSIA	
32		PA4	MTIC5U/MTCLKA/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/ SSLA0	IRQ5
33		PA3	MTIOC0D/MTCLKD	RXD5/SMISO5/SSCL5/IRRXD5	IRQ6
34		PA1	MTIOC0B/MTCLKC	SCK5/SSLA2	CVREFA
35		PE4	MTIOC4D/MTIOC1A		AN012/CMPA2
36		PE3	MTIOC4B/POE8#	CTS12#/RTS12#	AN011/CMPA1
37		PE2	MTIOC4A	RXD12/RXDX12/SSCL12	IRQ7/AN010
38		PE1	MTIOC4C	TXD12/TXDX12/SIOX12/ SSDA12	AN009
39	NC (Non-Connection)				
40		P46			AN006
41	NC (Non-Connection)				
42		P42			AN002
43		P41			AN001

Table 1.7 List of Pins and Pin Functions (48-Pin LQFP) (2 / 2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE)	Communication (SCIe, SCIf, RSPI, RIIC)	Others
44	VREFL0				
45		P40			AN000
46	VREFH0				
47	AVCC0				
48	AVSS0				

2. CPU

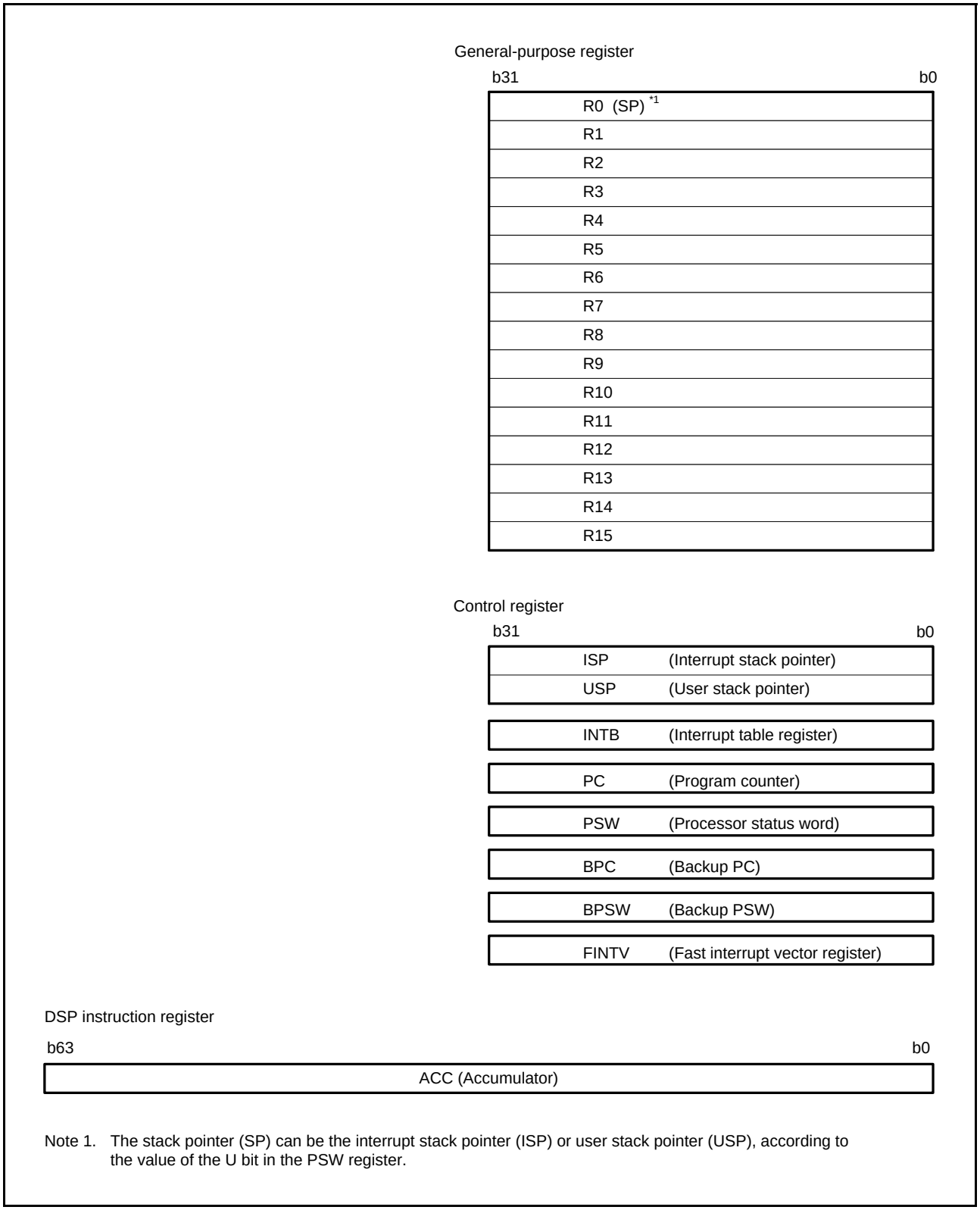
The RX220 Group is an MCU with the high-speed, high-performance RX CPU as its core.
A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.
The CPU has 73 basic instructions and nine DSP instructions, for a total of 82 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: Eight 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
- 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.
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

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



Note 1. The stack pointer (SP) can be the interrupt stack pointer (ISP) or user stack pointer (USP), according to the value of the U bit in the PSW register.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

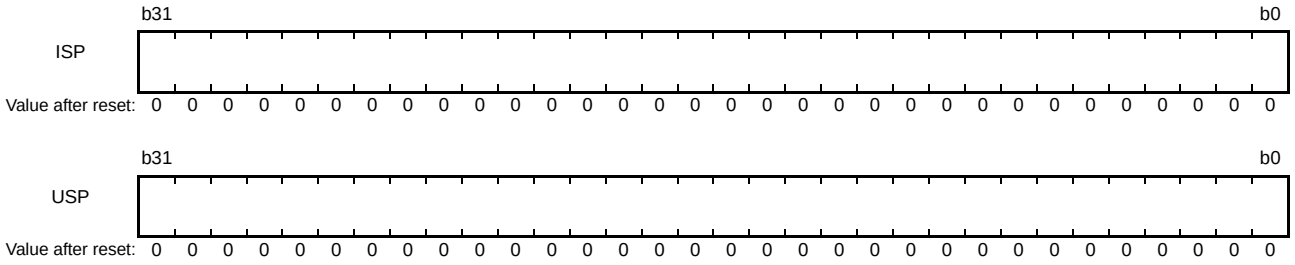
This CPU has sixteen general-purpose registers (R0 to R15). R1 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP). The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP)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

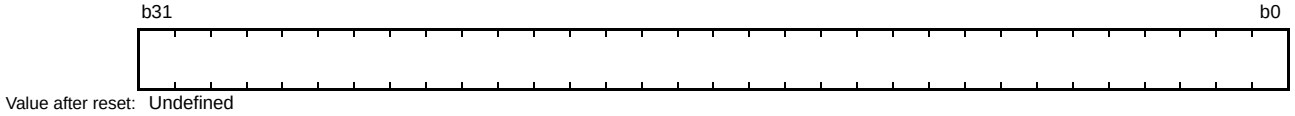
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)

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



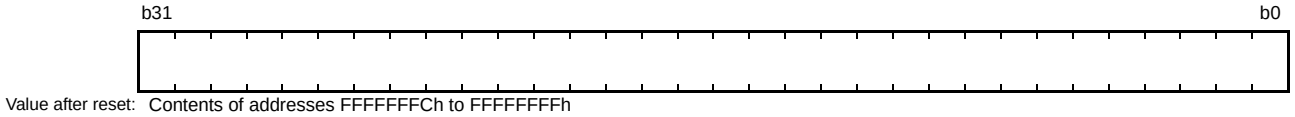
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of four,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

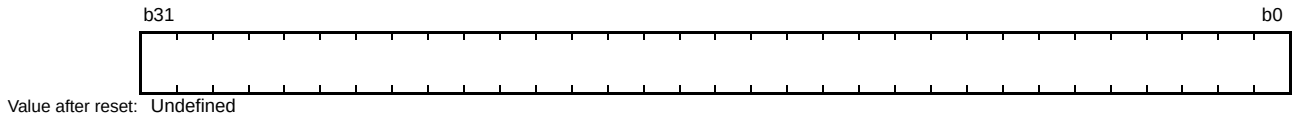
PM Bit (Processor Mode Select)

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)

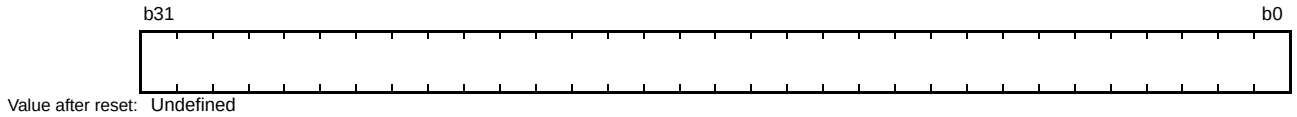
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the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

2.2.2.5 Backup PC (BPC)



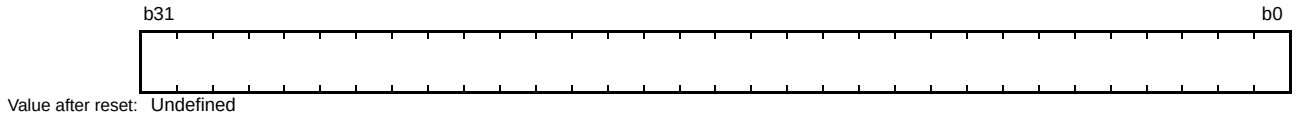
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

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.

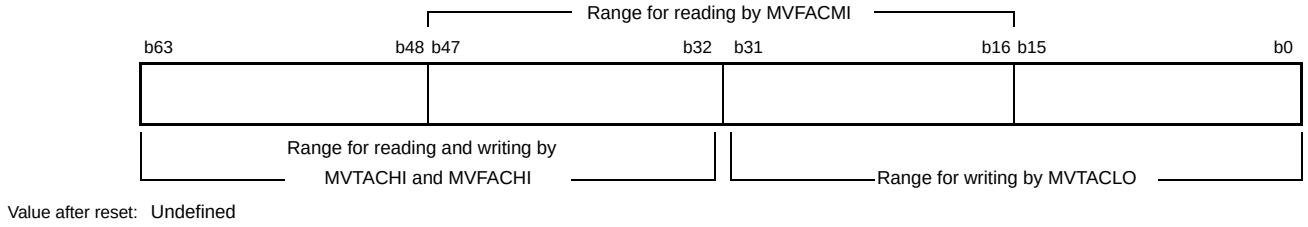
2.2.2.7 Fast Interrupt Vector Register (FINTV)



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.

2.2.3 Register Associated with DSP Instructions

2.2.3.1 Accumulator (ACC)



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, 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 three types of data: integer, 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 data arrangement is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, the RX220 Group supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, see section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.5.2 Access to I/O Registers

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

2.5.3 Notes on Access to I/O Registers

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

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

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

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

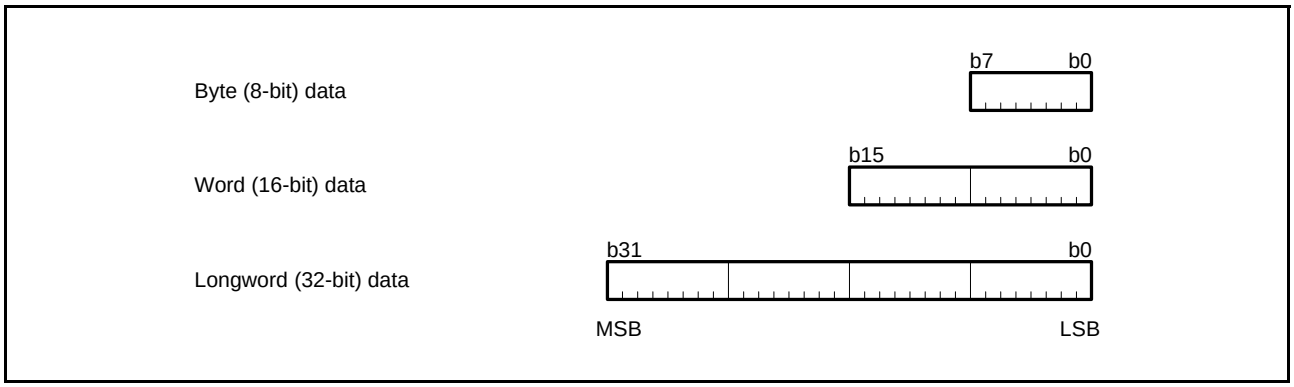


Figure 2.2 Data Arrangement in Registers

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.

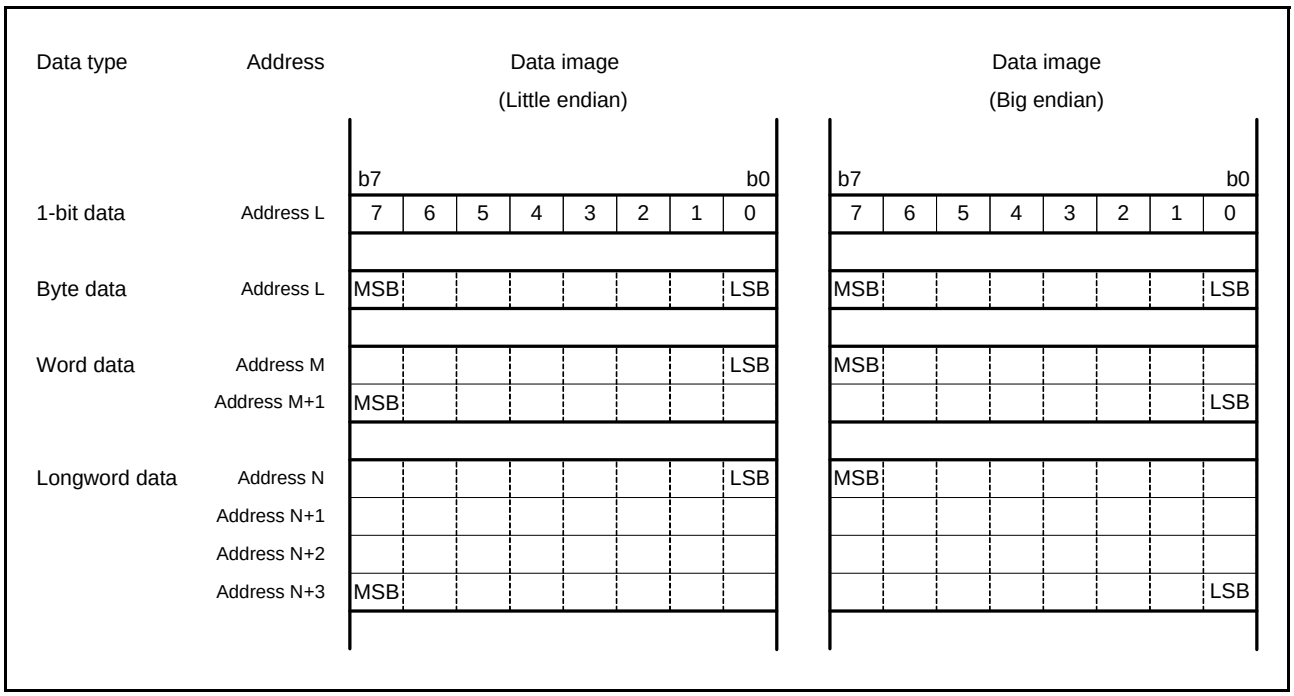


Figure 2.3 Data Arrangement in Memory

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, undefined instruction 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.

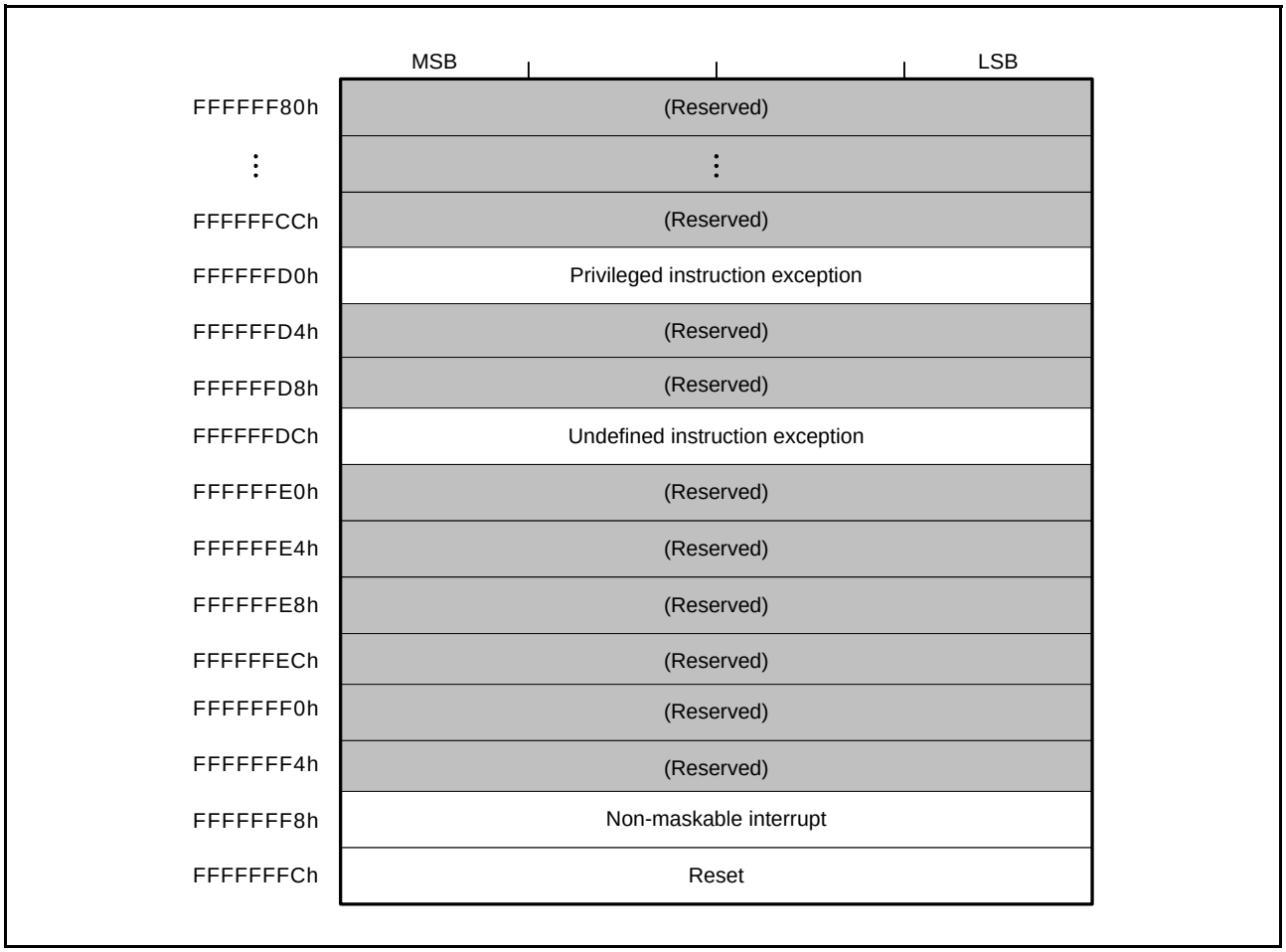


Figure 2.4 Fixed Vector Table

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.

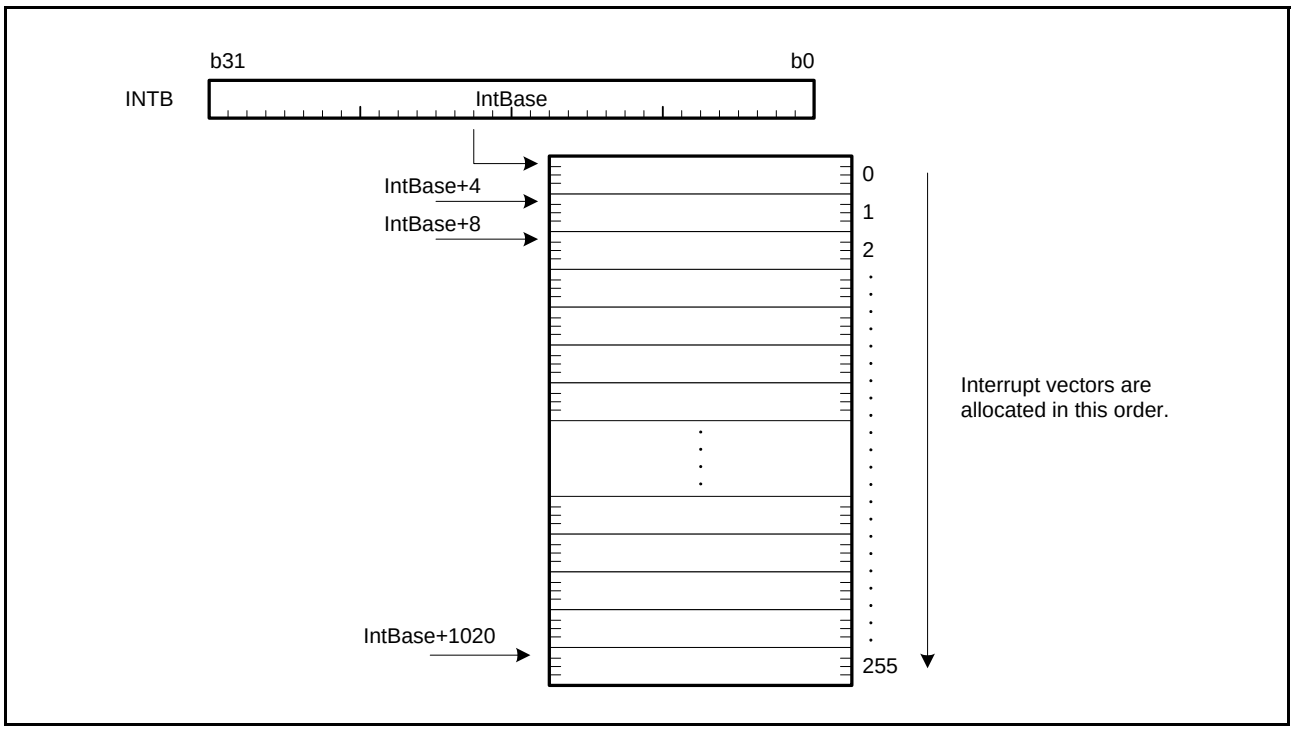


Figure 2.5 Relocatable Vector Table

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 4-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.

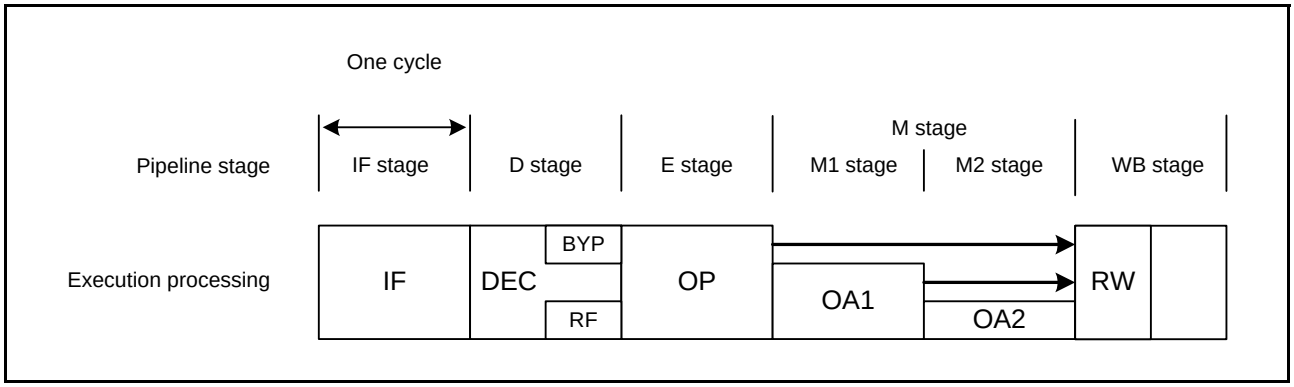


Figure 2.6 Pipeline Configuration and its Operation

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 2.17	Branch taken: 3 Branch not taken: 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.

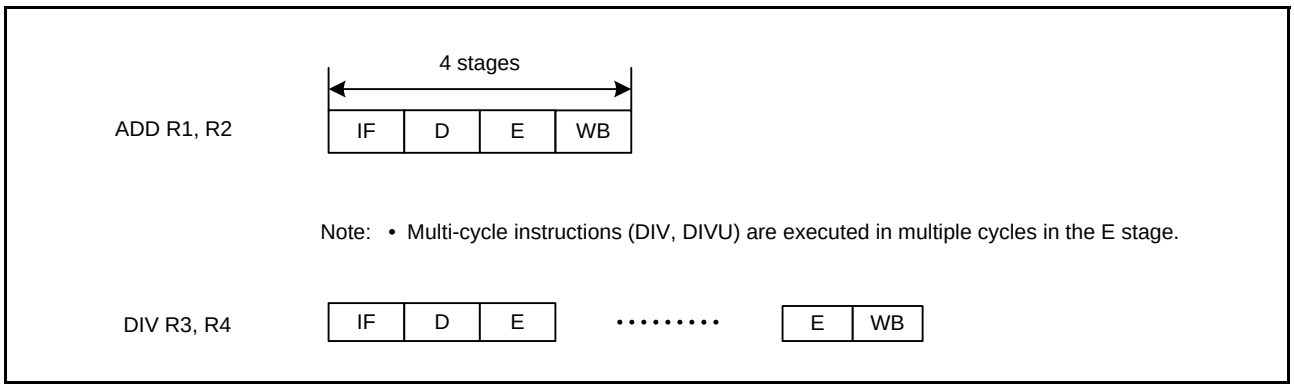


Figure 2.7 Operation for Register-Register, Immediate-Register

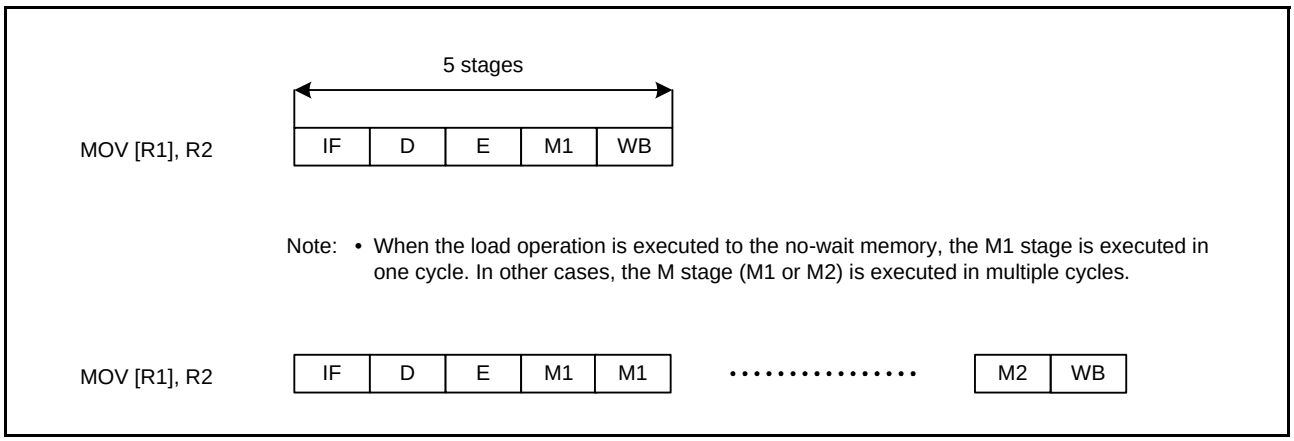


Figure 2.8 Load Operation

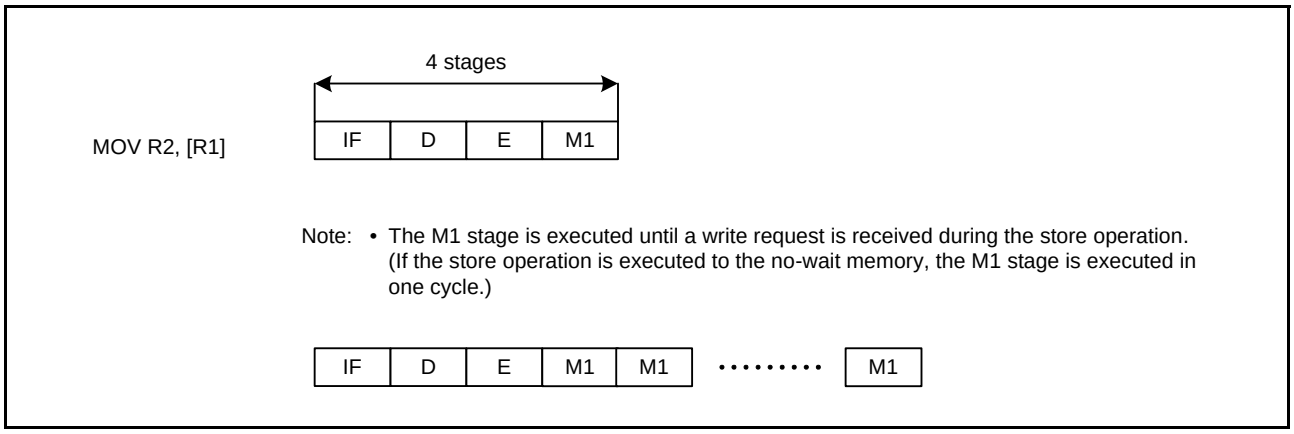


Figure 2.9 Store Operation

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 instruction (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate)	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	Figure 2.12	2
Arithmetic/logic instruction (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]” • BM*Cnd* “#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 instruction (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 instruction (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
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 2.14 show the operation of instructions that are converted into basic multiple micro-operations.
Note: • mop: Micro-operation, stall: Pipeline stall

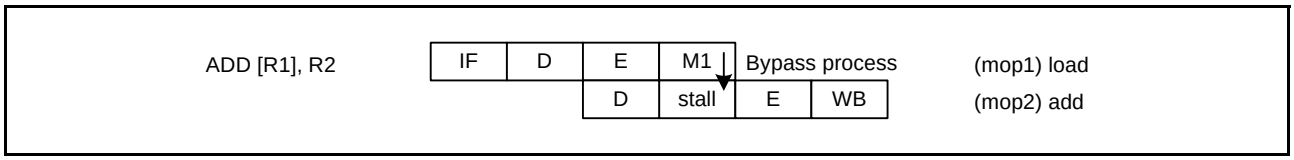


Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)

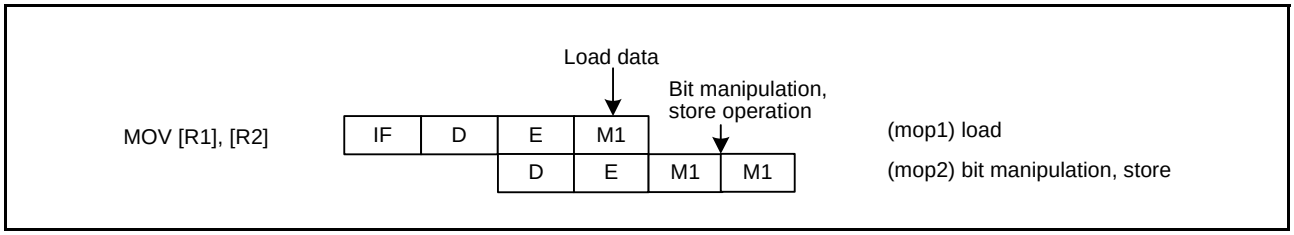


Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

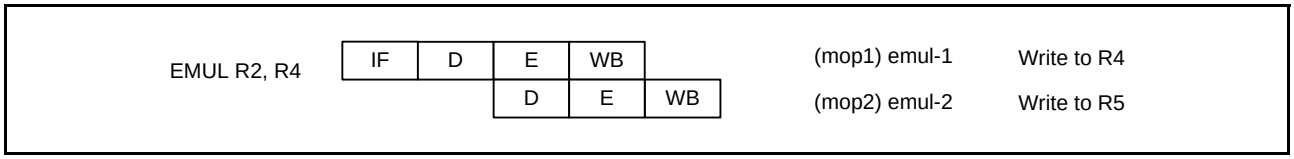


Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

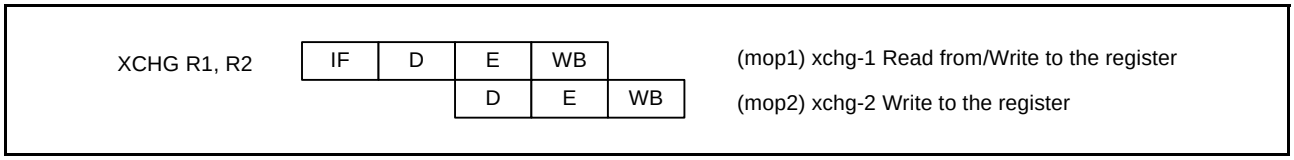


Figure 2.13 XCHG Instruction (Registers)

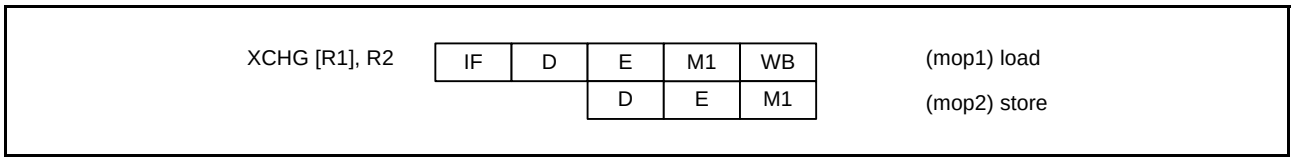


Figure 2.14 XCHG Instruction (Memory Source Operand)

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

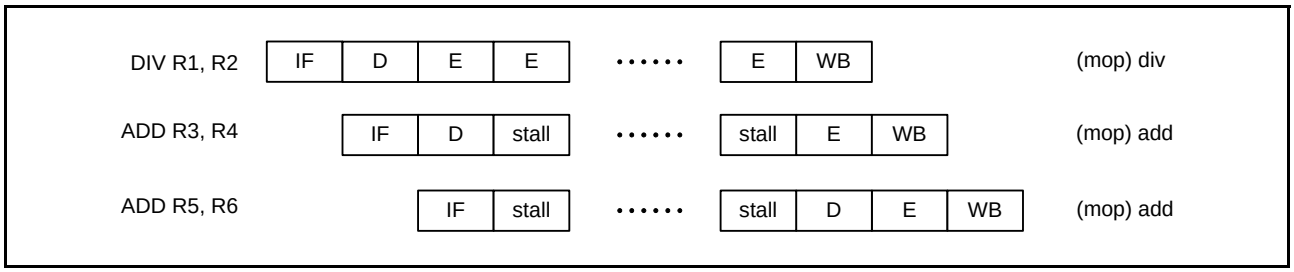


Figure 2.15 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

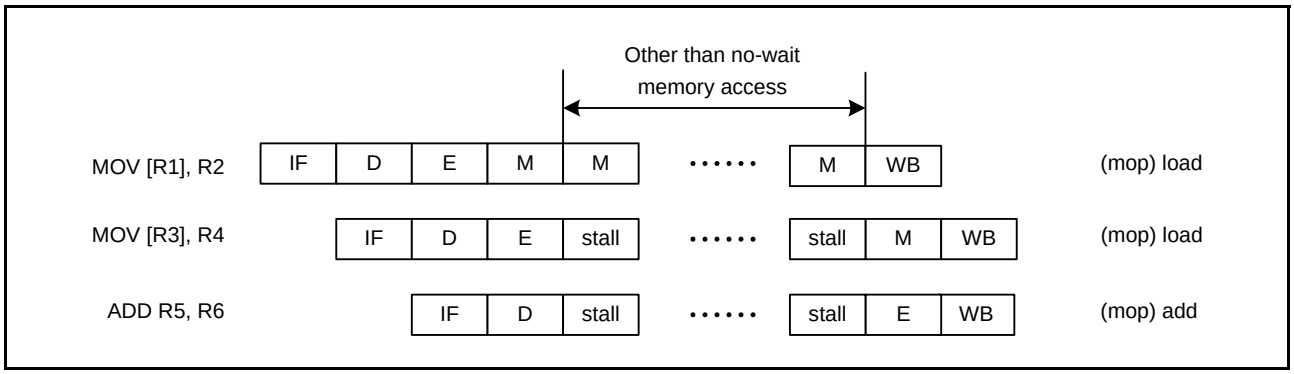


Figure 2.16 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

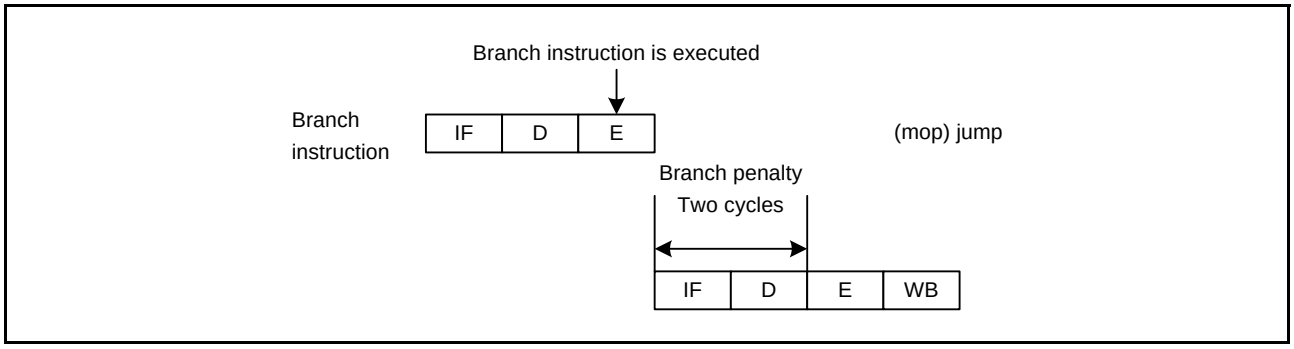


Figure 2.17 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

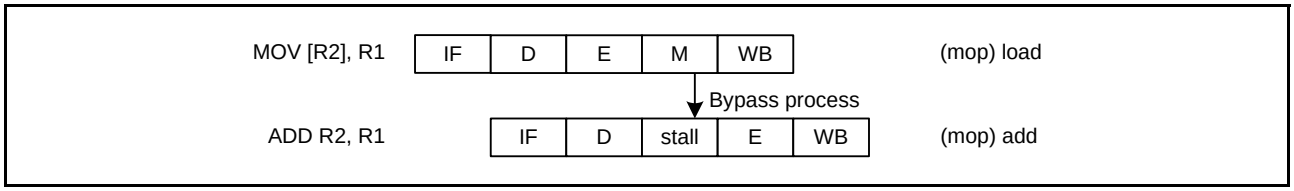


Figure 2.18 When the Subsequent Instruction Uses an Operand Read from the Memory

(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.

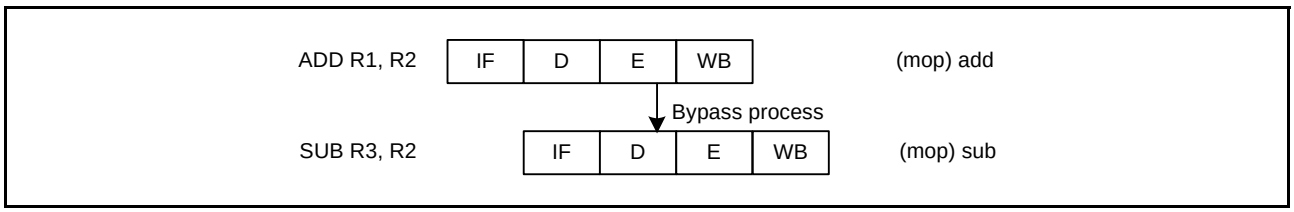


Figure 2.19 Bypass Process

(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.

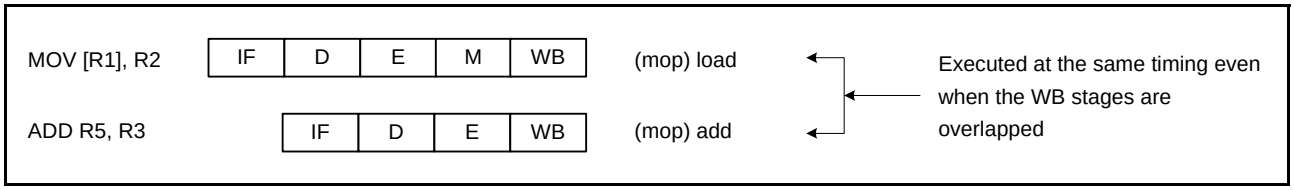


Figure 2.20 When WB Stages for the Memory Load and for the Operation are Overlapped

(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.

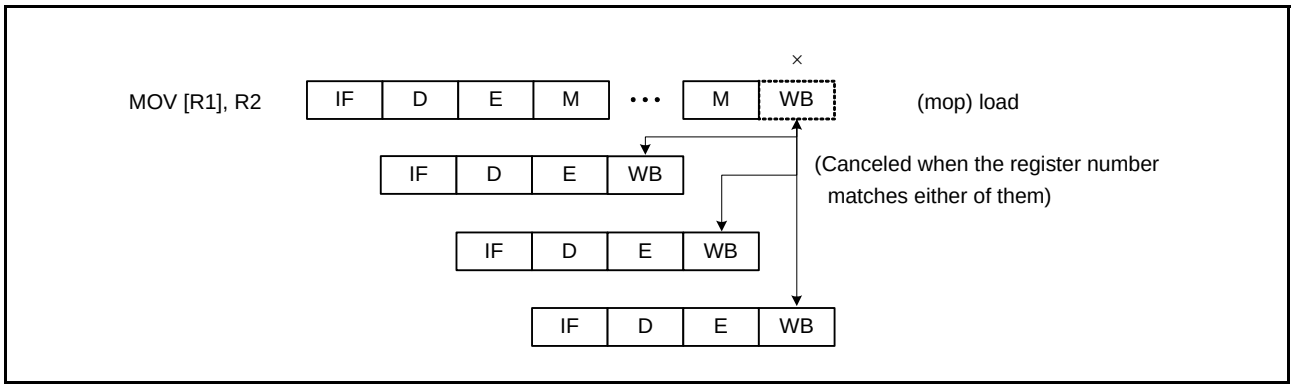


Figure 2.21 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(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).

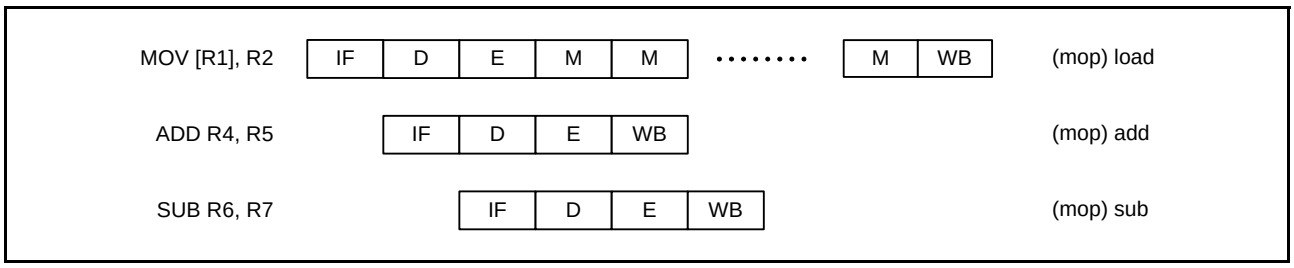


Figure 2.22 When Load Data is not Used by the Subsequent Instruction

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 the RX220 Groups 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 eight-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.

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

Mode-Setting Pin			
MD*1	**PC7*2**	**Operating Mode**	**On-chip ROM*3**
High	—	Single-chip mode	Enabled
Low	Low	Boot mode	Enabled
	High	User boot mode	Enabled

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.
Note 3. The on-chip ROM is classified into two types: ROM and E2 DataFlash.

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 3.2. For the correspondence between the setting and endian, see Table 3.3.

Table 3.2 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 3.3 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 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 38, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, 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.

3.3.2 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 36, ROM (Flash Memory for Code Storage), and section 37, E2 DataFlash Memory (Flash Memory for Data Storage).
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.

3.3.3 User Boot Mode

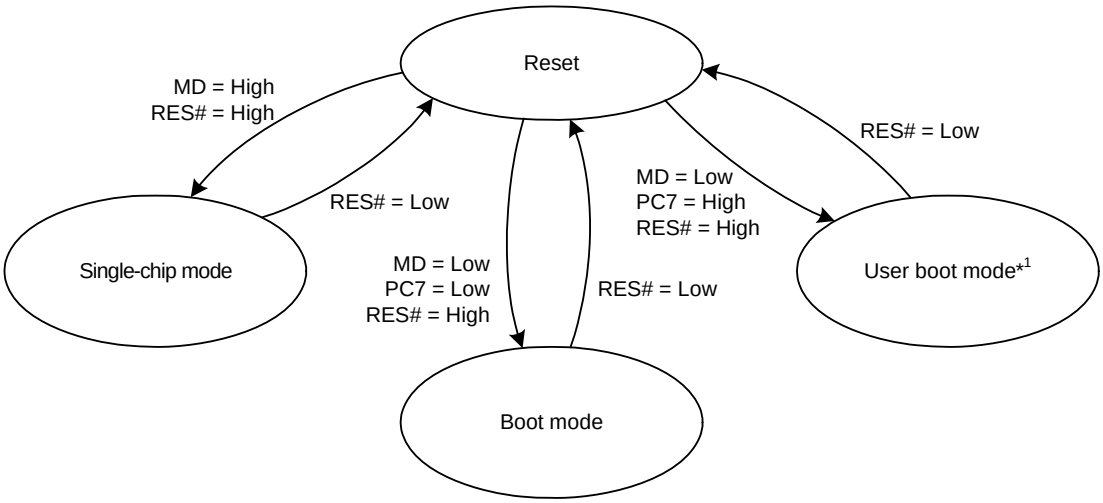
In user boot mode, an on-chip flash memory modifying program (user boot program) created by the user operates. After release from the reset state, the chip starts up in a state equivalent to single-chip mode.
After programming the prescribed values for UB code A and the UB code B, the chip starts up 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.

Note: • In user mode, do not make a transition to 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.

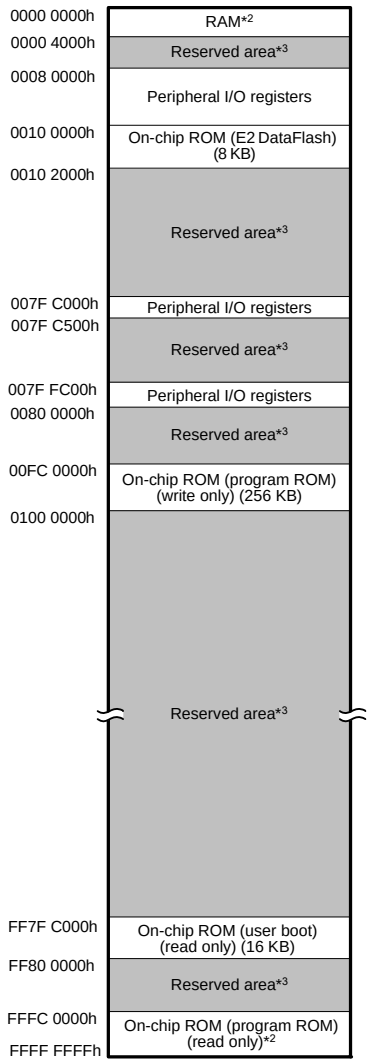
Figure 3.1 Mode-Setting Pin Levels and Operating Modes

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 map.

Single-chip mode*1



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 capacity of ROM/RAM differs depending on the products.

ROM (bytes)		RAM (bytes)	
Capacity	**Address**	**Capacity**	**Address**
256 K	FFFC 0000h to FFFF FFFFh	16 K	0000 0000h to 0000 3FFFh
128 K	FFFE 0000h to FFFF FFFFh	8 K	0000 0000h to 0000 1FFFh
64K	FFFF 0000h to FFFF FFFFh		
32 K	FFFF 8000h to FFFF FFFFh	4 K	0000 0000h to 0000 0FFFh

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

5. I/O Registers

This section provides information on the on-chip I/O register addresses and bit configuration. The information is given as shown below. Notes on writing to registers are also given below.

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

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Numbers of cycles for access indicate numbers of cycles of the given base 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 numbers of clock cycles for access to I/O registers, see 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 bus 1 to 6

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

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the instruction 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 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 0000h	SYSTEM	Mode monitor register	MDMONR	16	16		3 ICLK	section 3.
0008 0002h	SYSTEM	Mode status register	MDSR	16	16		3 ICLK	section 3.
0008 0008h	SYSTEM	System control register 1	SYSCR1	16	16		3 ICLK	section 3.
0008 000Ch	SYSTEM	Standby control register	SBYCR	16	16		3 ICLK	section 11.
0008 0010h	SYSTEM	Module stop control register A	MSTPCRA	32	32		3 ICLK	section 11.
0008 0014h	SYSTEM	Module stop control register B	MSTPCRB	32	32		3 ICLK	section 11.
0008 0018h	SYSTEM	Module stop control register C	MSTPCRC	32	32		3 ICLK	section 11.
0008 0020h	SYSTEM	System clock control register	SCKCR	32	32		3 ICLK	section 9.
0008 0026h	SYSTEM	System clock control register 3	SCKCR3	16	16		3 ICLK	section 9.
0008 0032h	SYSTEM	Main clock oscillator control register	MOSCCR	8	8		3 ICLK	section 9.
0008 0033h	SYSTEM	Sub-clock oscillator control register	SOSCCR	8	8		3 ICLK	section 9.
0008 0035h	SYSTEM	IWDT-dedicated on-chip oscillator control register	ILOCOCR	8	8		3 ICLK	section 9.
0008 0036h	SYSTEM	High-speed on-chip oscillator control register	HOCOCR	8	8		3 ICLK	section 9.
0008 0037h	SYSTEM	High-speed on-chip oscillator control register 2	HOCOCR2	8	8		3 ICLK	section 9.
0008 0040h	SYSTEM	Oscillation stop detection control register	OSTDCR	8	8		3 ICLK	section 9.
0008 0041h	SYSTEM	Oscillation stop detection status register	OSTDSR	8	8		3 ICLK	section 9.
0008 00A0h	SYSTEM	Operating power control register	OPCCR	8	8		3 ICLK	section 11.
0008 00A1h	SYSTEM	Sleep mode return clock source switching register	RSTCKCR	8	8		3 ICLK	section 11.
0008 00A2h	SYSTEM	Main clock oscillator wait control register	MOSCWTCR	8	8		3 ICLK	section 11.
0008 00A3h	SYSTEM	Sub-clock oscillator wait control register	SOSCWTCR	8	8		3 ICLK	section 11.
0008 00A9h	SYSTEM	HOCO wait control register 2	HOCOWTCR2	8	8		3 ICLK	section 11.
0008 00C0h	SYSTEM	Reset status register 2	RSTSR2	8	8		3 ICLK	section 6.
0008 00C2h	SYSTEM	Software reset register	SWRR	16	16		3 ICLK	section 6.
0008 00E0h	SYSTEM	Voltage monitoring 1 circuit/comparator A1 control register 1	LVD1CR1	8	8		3 ICLK	section 8., section 33.
0008 00E1h	SYSTEM	Voltage monitoring 1 circuit/comparator A1 status register	LVD1SR	8	8		3 ICLK	section 8., section 33.
0008 00E2h	SYSTEM	Voltage monitoring 2 circuit/comparator A2 control register 1	LVD2CR1	8	8		3 ICLK	section 8., section 33.
0008 00E3h	SYSTEM	Voltage monitoring 2 circuit/comparator A2 status register	LVD2SR	8	8		3 ICLK	section 8., section 33.
0008 03FEh	SYSTEM	Protect register	PRCR	16	16		3 ICLK	section 12.
0008 1300h	BSC	Bus error status clear register	BERCLR	8	8		2 ICLK	section 15.
0008 1304h	BSC	Bus error monitoring enable register	BEREN	8	8		2 ICLK	section 15.
0008 1308h	BSC	Bus error status register 1	BERSR1	8	8		2 ICLK	section 15.
0008 130Ah	BSC	Bus error status register 2	BERSR2	16	16		2 ICLK	section 15.
0008 1310h	BSC	Bus priority control register	BUSPRI	16	16		2 ICLK	section 15.
0008 2000h	DMAC0	DMA source address register	DMSAR	32	32		2 ICLK	section 16.
0008 2004h	DMAC0	DMA destination address register	DMDAR	32	32		2 ICLK	section 16.
0008 2008h	DMAC0	DMA transfer count register	DMCRA	32	32		2 ICLK	section 16.
0008 200Ch	DMAC0	DMA block transfer count register	DMCRB	16	16		2 ICLK	section 16.
0008 2010h	DMAC0	DMA transfer mode register	DMTMD	16	16		2 ICLK	section 16.
0008 2013h	DMAC0	DMA interrupt setting register	DMINT	8	8		2 ICLK	section 16.
0008 2014h	DMAC0	DMA address mode register	DMAMD	16	16		2 ICLK	section 16.
0008 2018h	DMAC0	DMA offset register	DMOFR	32	32		2 ICLK	section 16.
0008 201Ch	DMAC0	DMA transfer enable register	DMCNT	8	8		2 ICLK	section 16.
0008 201Dh	DMAC0	DMA software start register	DMREQ	8	8		2 ICLK	section 16.
0008 201Eh	DMAC0	DMA status register	DMSTS	8	8		2 ICLK	section 16.
0008 201Fh	DMAC0	DMA activation source flag control register	DMCSL	8	8		2 ICLK	section 16.
0008 2040h	DMAC1	DMA source address register	DMSAR	32	32		2 ICLK	section 16.

Table 5.1 List of I/O Registers (Address Order) (2 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 2044h	DMAC1	DMA destination address register	DMDAR	32	32		2 ICLK	section 16.
0008 2048h	DMAC1	DMA transfer count register	DMCRA	32	32		2 ICLK	section 16.
0008 204Ch	DMAC1	DMA block transfer count register	DMCRB	16	16		2 ICLK	section 16.
0008 2050h	DMAC1	DMA transfer mode register	DMTMD	16	16		2 ICLK	section 16.
0008 2053h	DMAC1	DMA interrupt setting register	DMINT	8	8		2 ICLK	section 16.
0008 2054h	DMAC1	DMA address mode register	DMAMD	16	16		2 ICLK	section 16.
0008 205Ch	DMAC1	DMA transfer enable register	DMCNT	8	8		2 ICLK	section 16.
0008 205Dh	DMAC1	DMA software start register	DMREQ	8	8		2 ICLK	section 16.
0008 205Eh	DMAC1	DMA status register	DMSTS	8	8		2 ICLK	section 16.
0008 205Fh	DMAC1	DMA activation source flag control register	DMCSL	8	8		2 ICLK	section 16.
0008 2080h	DMAC2	DMA source address register	DMSAR	32	32		2 ICLK	section 16.
0008 2084h	DMAC2	DMA destination address register	DMDAR	32	32		2 ICLK	section 16.
0008 2088h	DMAC2	DMA transfer count register	DMCRA	32	32		2 ICLK	section 16.
0008 208Ch	DMAC2	DMA block transfer count register	DMCRB	16	16		2 ICLK	section 16.
0008 2090h	DMAC2	DMA transfer mode register	DMTMD	16	16		2 ICLK	section 16.
0008 2093h	DMAC2	DMA interrupt setting register	DMINT	8	8		2 ICLK	section 16.
0008 2094h	DMAC2	DMA address mode register	DMAMD	16	16		2 ICLK	section 16.
0008 209Ch	DMAC2	DMA transfer enable register	DMCNT	8	8		2 ICLK	section 16.
0008 209Dh	DMAC2	DMA software start register	DMREQ	8	8		2 ICLK	section 16.
0008 209Eh	DMAC2	DMA status register	DMSTS	8	8		2 ICLK	section 16.
0008 209Fh	DMAC2	DMA activation source flag control register	DMCSL	8	8		2 ICLK	section 16.
0008 20C0h	DMAC3	DMA source address register	DMSAR	32	32		2 ICLK	section 16.
0008 20C4h	DMAC3	DMA destination address register	DMDAR	32	32		2 ICLK	section 16.
0008 20C8h	DMAC3	DMA transfer count register	DMCRA	32	32		2 ICLK	section 16.
0008 20CCh	DMAC3	DMA block transfer count register	DMCRB	16	16		2 ICLK	section 16.
0008 20D0h	DMAC3	DMA transfer mode register	DMTMD	16	16		2 ICLK	section 16.
0008 20D3h	DMAC3	DMA interrupt setting register	DMINT	8	8		2 ICLK	section 16.
0008 20D4h	DMAC3	DMA address mode register	DMAMD	16	16		2 ICLK	section 16.
0008 20DCh	DMAC3	DMA transfer enable register	DMCNT	8	8		2 ICLK	section 16.
0008 20DDh	DMAC3	DMA software start register	DMREQ	8	8		2 ICLK	section 16.
0008 20DEh	DMAC3	DMA status register	DMSTS	8	8		2 ICLK	section 16.
0008 20DFh	DMAC3	DMA activation source flag control register	DMCSL	8	8		2 ICLK	section 16.
0008 2200h	DMAC	DMA module activation register	DMAST	8	8		2 ICLK	section 16.
0008 2400h	DTC	DTC control register	DTCCR	8	8		2 ICLK	section 17.
0008 2404h	DTC	DTC vector base register	DTCVBR	32	32		2 ICLK	section 17.
0008 2408h	DTC	DTC address mode register	DTCADMOD	8	8		2 ICLK	section 17.
0008 240Ch	DTC	DTC module start register	DTCST	8	8		2 ICLK	section 17.
0008 240Eh	DTC	DTC status register	DTCSTS	16	16		2 ICLK	section 17.
0008 7010h	ICU	Interrupt request register 016	IR016	8	8		2 ICLK	section 14.
0008 7015h	ICU	Interrupt request register 021	IR021	8	8		2 ICLK	section 14.
0008 7017h	ICU	Interrupt request register 023	IR023	8	8		2 ICLK	section 14.
0008 701Bh	ICU	Interrupt request register 027	IR027	8	8		2 ICLK	section 14.
0008 701Ch	ICU	Interrupt request register 028	IR028	8	8		2 ICLK	section 14.
0008 701Dh	ICU	Interrupt request register 029	IR029	8	8		2 ICLK	section 14.
0008 701Eh	ICU	Interrupt request register 030	IR030	8	8		2 ICLK	section 14.
0008 701Fh	ICU	Interrupt request register 031	IR031	8	8		2 ICLK	section 14.
0008 7020h	ICU	Interrupt request register 032	IR032	8	8		2 ICLK	section 14.
0008 7021h	ICU	Interrupt request register 033	IR033	8	8		2 ICLK	section 14.
0008 7022h	ICU	Interrupt request register 034	IR034	8	8		2 ICLK	section 14.
0008 702Ch	ICU	Interrupt request register 044	IR044	8	8		2 ICLK	section 14.
0008 702Dh	ICU	Interrupt request register 045	IR045	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (3 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 702Eh	ICU	Interrupt request register 046	IR046	8	8		2 ICLK	section 14.
0008 702Fh	ICU	Interrupt request register 047	IR047	8	8		2 ICLK	section 14.
0008 7039h	ICU	Interrupt request register 057	IR057	8	8		2 ICLK	section 14.
0008 703Fh	ICU	Interrupt request register 063	IR063	8	8		2 ICLK	section 14.
0008 7040h	ICU	Interrupt request register 064	IR064	8	8		2 ICLK	section 14.
0008 7041h	ICU	Interrupt request register 065	IR065	8	8		2 ICLK	section 14.
0008 7042h	ICU	Interrupt request register 066	IR066	8	8		2 ICLK	section 14.
0008 7043h	ICU	Interrupt request register 067	IR067	8	8		2 ICLK	section 14.
0008 7044h	ICU	Interrupt request register 068	IR068	8	8		2 ICLK	section 14.
0008 7045h	ICU	Interrupt request register 069	IR069	8	8		2 ICLK	section 14.
0008 7046h	ICU	Interrupt request register 070	IR070	8	8		2 ICLK	section 14.
0008 7047h	ICU	Interrupt request register 071	IR071	8	8		2 ICLK	section 14.
0008 7058h	ICU	Interrupt request register 088	IR088	8	8		2 ICLK	section 14.
0008 7059h	ICU	Interrupt request register 089	IR089	8	8		2 ICLK	section 14.
0008 705Ch	ICU	Interrupt request register 092	IR092	8	8		2 ICLK	section 14.
0008 705Dh	ICU	Interrupt request register 093	IR093	8	8		2 ICLK	section 14.
0008 7066h	ICU	Interrupt request register 102	IR102	8	8		2 ICLK	section 14.
0008 7067h	ICU	Interrupt request register 103	IR103	8	8		2 ICLK	section 14.
0008 706Ah	ICU	Interrupt request register 106	IR106	8	8		2 ICLK	section 14.
0008 7072h	ICU	Interrupt request register 114	IR114	8	8		2 ICLK	section 14.
0008 7073h	ICU	Interrupt request register 115	IR115	8	8		2 ICLK	section 14.
0008 7074h	ICU	Interrupt request register 116	IR116	8	8		2 ICLK	section 14.
0008 7075h	ICU	Interrupt request register 117	IR117	8	8		2 ICLK	section 14.
0008 7076h	ICU	Interrupt request register 118	IR118	8	8		2 ICLK	section 14.
0008 7077h	ICU	Interrupt request register 119	IR119	8	8		2 ICLK	section 14.
0008 7078h	ICU	Interrupt request register 120	IR120	8	8		2 ICLK	section 14.
0008 7079h	ICU	Interrupt request register 121	IR121	8	8		2 ICLK	section 14.
0008 707Ah	ICU	Interrupt request register 122	IR122	8	8		2 ICLK	section 14.
0008 707Bh	ICU	Interrupt request register 123	IR123	8	8		2 ICLK	section 14.
0008 707Ch	ICU	Interrupt request register 124	IR124	8	8		2 ICLK	section 14.
0008 707Dh	ICU	Interrupt request register 125	IR125	8	8		2 ICLK	section 14.
0008 707Eh	ICU	Interrupt request register 126	IR126	8	8		2 ICLK	section 14.
0008 707Fh	ICU	Interrupt request register 127	IR127	8	8		2 ICLK	section 14.
0008 7080h	ICU	Interrupt request register 128	IR128	8	8		2 ICLK	section 14.
0008 7081h	ICU	Interrupt request register 129	IR129	8	8		2 ICLK	section 14.
0008 7082h	ICU	Interrupt request register 130	IR130	8	8		2 ICLK	section 14.
0008 7083h	ICU	Interrupt request register 131	IR131	8	8		2 ICLK	section 14.
0008 7084h	ICU	Interrupt request register 132	IR132	8	8		2 ICLK	section 14.
0008 7085h	ICU	Interrupt request register 133	IR133	8	8		2 ICLK	section 14.
0008 7086h	ICU	Interrupt request register 134	IR134	8	8		2 ICLK	section 14.
0008 7087h	ICU	Interrupt request register 135	IR135	8	8		2 ICLK	section 14.
0008 7088h	ICU	Interrupt request register 136	IR136	8	8		2 ICLK	section 14.
0008 7089h	ICU	Interrupt request register 137	IR137	8	8		2 ICLK	section 14.
0008 708Ah	ICU	Interrupt request register 138	IR138	8	8		2 ICLK	section 14.
0008 708Bh	ICU	Interrupt request register 139	IR139	8	8		2 ICLK	section 14.
0008 708Ch	ICU	Interrupt request register 140	IR140	8	8		2 ICLK	section 14.
0008 708Dh	ICU	Interrupt request register 141	IR141	8	8		2 ICLK	section 14.
0008 70AAh	ICU	Interrupt request register 170	IR170	8	8		2 ICLK	section 14.
0008 70ABh	ICU	Interrupt request register 171	IR171	8	8		2 ICLK	section 14.
0008 70AEh	ICU	Interrupt request register 174	IR174	8	8		2 ICLK	section 14.
0008 70AFh	ICU	Interrupt request register 175	IR175	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (4 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 70B0h	ICU	Interrupt request register 176	IR176	8	8		2 ICLK	section 14.
0008 70B1h	ICU	Interrupt request register 177	IR177	8	8		2 ICLK	section 14.
0008 70B2h	ICU	Interrupt request register 178	IR178	8	8		2 ICLK	section 14.
0008 70B3h	ICU	Interrupt request register 179	IR179	8	8		2 ICLK	section 14.
0008 70B4h	ICU	Interrupt request register 180	IR180	8	8		2 ICLK	section 14.
0008 70B5h	ICU	Interrupt request register 181	IR181	8	8		2 ICLK	section 14.
0008 70B6h	ICU	Interrupt request register 182	IR182	8	8		2 ICLK	section 14.
0008 70B7h	ICU	Interrupt request register 183	IR183	8	8		2 ICLK	section 14.
0008 70B8h	ICU	Interrupt request register 184	IR184	8	8		2 ICLK	section 14.
0008 70B9h	ICU	Interrupt request register 185	IR185	8	8		2 ICLK	section 14.
0008 70C6h	ICU	Interrupt request register 198	IR198	8	8		2 ICLK	section 14.
0008 70C7h	ICU	Interrupt request register 199	IR199	8	8		2 ICLK	section 14.
0008 70C8h	ICU	Interrupt request register 200	IR200	8	8		2 ICLK	section 14.
0008 70C9h	ICU	Interrupt request register 201	IR201	8	8		2 ICLK	section 14.
0008 70DAh	ICU	Interrupt request register 218	IR218	8	8		2 ICLK	section 14.
0008 70DBh	ICU	Interrupt request register 219	IR219	8	8		2 ICLK	section 14.
0008 70DCh	ICU	Interrupt request register 220	IR220	8	8		2 ICLK	section 14.
0008 70DDh	ICU	Interrupt request register 221	IR221	8	8		2 ICLK	section 14.
0008 70DEh	ICU	Interrupt request register 222	IR222	8	8		2 ICLK	section 14.
0008 70DFh	ICU	Interrupt request register 223	IR223	8	8		2 ICLK	section 14.
0008 70E0h	ICU	Interrupt request register 224	IR224	8	8		2 ICLK	section 14.
0008 70E1h	ICU	Interrupt request register 225	IR225	8	8		2 ICLK	section 14.
0008 70E2h	ICU	Interrupt request register 226	IR226	8	8		2 ICLK	section 14.
0008 70E3h	ICU	Interrupt request register 227	IR227	8	8		2 ICLK	section 14.
0008 70E4h	ICU	Interrupt request register 228	IR228	8	8		2 ICLK	section 14.
0008 70E5h	ICU	Interrupt request register 229	IR229	8	8		2 ICLK	section 14.
0008 70EAh	ICU	Interrupt request register 234	IR234	8	8		2 ICLK	section 14.
0008 70EBh	ICU	Interrupt request register 235	IR235	8	8		2 ICLK	section 14.
0008 70ECh	ICU	Interrupt request register 236	IR236	8	8		2 ICLK	section 14.
0008 70EDh	ICU	Interrupt request register 237	IR237	8	8		2 ICLK	section 14.
0008 70EEh	ICU	Interrupt request register 238	IR238	8	8		2 ICLK	section 14.
0008 70EFh	ICU	Interrupt request register 239	IR239	8	8		2 ICLK	section 14.
0008 70F0h	ICU	Interrupt request register 240	IR240	8	8		2 ICLK	section 14.
0008 70F1h	ICU	Interrupt request register 241	IR241	8	8		2 ICLK	section 14.
0008 70F2h	ICU	Interrupt request register 242	IR242	8	8		2 ICLK	section 14.
0008 70F3h	ICU	Interrupt request register 243	IR243	8	8		2 ICLK	section 14.
0008 70F4h	ICU	Interrupt request register 244	IR244	8	8		2 ICLK	section 14.
0008 70F5h	ICU	Interrupt request register 245	IR245	8	8		2 ICLK	section 14.
0008 70F6h	ICU	Interrupt request register 246	IR246	8	8		2 ICLK	section 14.
0008 70F7h	ICU	Interrupt request register 247	IR247	8	8		2 ICLK	section 14.
0008 70F8h	ICU	Interrupt request register 248	IR248	8	8		2 ICLK	section 14.
0008 70F9h	ICU	Interrupt request register 249	IR249	8	8		2 ICLK	section 14.
0008 711Bh	ICU	DTC activation enable register 027	DTCER027	8	8		2 ICLK	section 14.
0008 711Ch	ICU	DTC activation enable register 028	DTCER028	8	8		2 ICLK	section 14.
0008 711Dh	ICU	DTC activation enable register 029	DTCER029	8	8		2 ICLK	section 14.
0008 711Eh	ICU	DTC activation enable register 030	DTCER030	8	8		2 ICLK	section 14.
0008 711Fh	ICU	DTC activation enable register 031	DTCER031	8	8		2 ICLK	section 14.
0008 712Dh	ICU	DTC activation enable register 045	DTCER045	8	8		2 ICLK	section 14.
0008 712Eh	ICU	DTC activation enable register 046	DTCER046	8	8		2 ICLK	section 14.
0008 7140h	ICU	DTC activation enable register 064	DTCER064	8	8		2 ICLK	section 14.
0008 7141h	ICU	DTC activation enable register 065	DTCER065	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (5 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 7142h	ICU	DTC activation enable register 066	DTCER066	8	8		2 ICLK	section 14.
0008 7143h	ICU	DTC activation enable register 067	DTCER067	8	8		2 ICLK	section 14.
0008 7144h	ICU	DTC activation enable register 068	DTCER068	8	8		2 ICLK	section 14.
0008 7145h	ICU	DTC activation enable register 069	DTCER069	8	8		2 ICLK	section 14.
0008 7146h	ICU	DTC activation enable register 070	DTCER070	8	8		2 ICLK	section 14.
0008 7147h	ICU	DTC activation enable register 071	DTCER071	8	8		2 ICLK	section 14.
0008 7166h	ICU	DTC activation enable register 102	DTCER102	8	8		2 ICLK	section 14.
0008 7167h	ICU	DTC activation enable register 103	DTCER103	8	8		2 ICLK	section 14.
0008 716Ah	ICU	DTC activation enable register 106	DTCER106	8	8		2 ICLK	section 14.
0008 7172h	ICU	DTC activation enable register 114	DTCER114	8	8		2 ICLK	section 14.
0008 7173h	ICU	DTC activation enable register 115	DTCER115	8	8		2 ICLK	section 14.
0008 7174h	ICU	DTC activation enable register 116	DTCER116	8	8		2 ICLK	section 14.
0008 7175h	ICU	DTC activation enable register 117	DTCER117	8	8		2 ICLK	section 14.
0008 7179h	ICU	DTC activation enable register 121	DTCER121	8	8		2 ICLK	section 14.
0008 717Ah	ICU	DTC activation enable register 122	DTCER122	8	8		2 ICLK	section 14.
0008 717Dh	ICU	DTC activation enable register 125	DTCER125	8	8		2 ICLK	section 14.
0008 717Eh	ICU	DTC activation enable register 126	DTCER126	8	8		2 ICLK	section 14.
0008 7181h	ICU	DTC activation enable register 129	DTCER129	8	8		2 ICLK	section 14.
0008 7182h	ICU	DTC activation enable register 130	DTCER130	8	8		2 ICLK	section 14.
0008 7183h	ICU	DTC activation enable register 131	DTCER131	8	8		2 ICLK	section 14.
0008 7184h	ICU	DTC activation enable register 132	DTCER132	8	8		2 ICLK	section 14.
0008 7186h	ICU	DTC activation enable register 134	DTCER134	8	8		2 ICLK	section 14.
0008 7187h	ICU	DTC activation enable register 135	DTCER135	8	8		2 ICLK	section 14.
0008 7188h	ICU	DTC activation enable register 136	DTCER136	8	8		2 ICLK	section 14.
0008 7189h	ICU	DTC activation enable register 137	DTCER137	8	8		2 ICLK	section 14.
0008 718Ah	ICU	DTC activation enable register 138	DTCER138	8	8		2 ICLK	section 14.
0008 718Bh	ICU	DTC activation enable register 139	DTCER139	8	8		2 ICLK	section 14.
0008 718Ch	ICU	DTC activation enable register 140	DTCER140	8	8		2 ICLK	section 14.
0008 718Dh	ICU	DTC activation enable register 141	DTCER141	8	8		2 ICLK	section 14.
0008 71AEh	ICU	DTC activation enable register 174	DTCER174	8	8		2 ICLK	section 14.
0008 71AFh	ICU	DTC activation enable register 175	DTCER175	8	8		2 ICLK	section 14.
0008 71B1h	ICU	DTC activation enable register 177	DTCER177	8	8		2 ICLK	section 14.
0008 71B2h	ICU	DTC activation enable register 178	DTCER178	8	8		2 ICLK	section 14.
0008 71B4h	ICU	DTC activation enable register 180	DTCER180	8	8		2 ICLK	section 14.
0008 71B5h	ICU	DTC activation enable register 181	DTCER181	8	8		2 ICLK	section 14.
0008 71B7h	ICU	DTC activation enable register 183	DTCER183	8	8		2 ICLK	section 14.
0008 71B8h	ICU	DTC activation enable register 184	DTCER184	8	8		2 ICLK	section 14.
0008 71C6h	ICU	DTC activation enable register 198	DTCER198	8	8		2 ICLK	section 14.
0008 71C7h	ICU	DTC activation enable register 199	DTCER199	8	8		2 ICLK	section 14.
0008 71C8h	ICU	DTC activation enable register 200	DTCER200	8	8		2 ICLK	section 14.
0008 71C9h	ICU	DTC activation enable register 201	DTCER201	8	8		2 ICLK	section 14.
0008 71DBh	ICU	DTC activation enable register 219	DTCER219	8	8		2 ICLK	section 14.
0008 71DCh	ICU	DTC activation enable register 220	DTCER220	8	8		2 ICLK	section 14.
0008 71DFh	ICU	DTC activation enable register 223	DTCER223	8	8		2 ICLK	section 14.
0008 71E0h	ICU	DTC activation enable register 224	DTCER224	8	8		2 ICLK	section 14.
0008 71E3h	ICU	DTC activation enable register 227	DTCER227	8	8		2 ICLK	section 14.
0008 71E4h	ICU	DTC activation enable register 228	DTCER228	8	8		2 ICLK	section 14.
0008 71EBh	ICU	DTC activation enable register 235	DTCER235	8	8		2 ICLK	section 14.
0008 71ECh	ICU	DTC activation enable register 236	DTCER236	8	8		2 ICLK	section 14.
0008 71EFh	ICU	DTC activation enable register 239	DTCER239	8	8		2 ICLK	section 14.
0008 71F0h	ICU	DTC activation enable register 240	DTCER240	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (6 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Number of Access Cycles ICLK < PCLK	Reference Section
0008 71F7h	ICU	DTC activation enable register 247	DTCER247	8	8		2 ICLK	section 14.
0008 71F8h	ICU	DTC activation enable register 248	DTCER248	8	8		2 ICLK	section 14.
0008 7202h	ICU	Interrupt request enable register 02	IER02	8	8		2 ICLK	section 14.
0008 7203h	ICU	Interrupt request enable register 03	IER03	8	8		2 ICLK	section 14.
0008 7204h	ICU	Interrupt request enable register 04	IER04	8	8		2 ICLK	section 14.
0008 7205h	ICU	Interrupt request enable register 05	IER05	8	8		2 ICLK	section 14.
0008 7207h	ICU	Interrupt request enable register 07	IER07	8	8		2 ICLK	section 14.
0008 7208h	ICU	Interrupt request enable register 08	IER08	8	8		2 ICLK	section 14.
0008 720Bh	ICU	Interrupt request enable register 0B	IER0B	8	8		2 ICLK	section 14.
0008 720Ch	ICU	Interrupt request enable register 0C	IER0C	8	8		2 ICLK	section 14.
0008 720Dh	ICU	Interrupt request enable register 0D	IER0D	8	8		2 ICLK	section 14.
0008 720Eh	ICU	Interrupt request enable register 0E	IER0E	8	8		2 ICLK	section 14.
0008 720Fh	ICU	Interrupt request enable register 0F	IER0F	8	8		2 ICLK	section 14.
0008 7210h	ICU	Interrupt request enable register 10	IER10	8	8		2 ICLK	section 14.
0008 7211h	ICU	Interrupt request enable register 11	IER11	8	8		2 ICLK	section 14.
0008 7215h	ICU	Interrupt request enable register 15	IER15	8	8		2 ICLK	section 14.
0008 7216h	ICU	Interrupt request enable register 16	IER16	8	8		2 ICLK	section 14.
0008 7217h	ICU	Interrupt request enable register 17	IER17	8	8		2 ICLK	section 14.
0008 7218h	ICU	Interrupt request enable register 18	IER18	8	8		2 ICLK	section 14.
0008 7219h	ICU	Interrupt request enable register 19	IER19	8	8		2 ICLK	section 14.
0008 721Bh	ICU	Interrupt request enable register 1B	IER1B	8	8		2 ICLK	section 14.
0008 721Ch	ICU	Interrupt request enable register 1C	IER1C	8	8		2 ICLK	section 14.
0008 721Dh	ICU	Interrupt request enable register 1D	IER1D	8	8		2 ICLK	section 14.
0008 721Eh	ICU	Interrupt request enable register 1E	IER1E	8	8		2 ICLK	section 14.
0008 721Fh	ICU	Interrupt request enable register 1F	IER1F	8	8		2 ICLK	section 14.
0008 72E0h	ICU	Software interrupt activation register	SWINTR	8	8		2 ICLK	section 14.
0008 72F0h	ICU	Fast interrupt set register	FIR	16	16		2 ICLK	section 14.
0008 7300h	ICU	Interrupt source priority register 000	IPR000	8	8		2 ICLK	section 14.
0008 7301h	ICU	Interrupt source priority register 001	IPR001	8	8		2 ICLK	section 14.
0008 7302h	ICU	Interrupt source priority register 002	IPR002	8	8		2 ICLK	section 14.
0008 7303h	ICU	Interrupt source priority register 003	IPR003	8	8		2 ICLK	section 14.
0008 7304h	ICU	Interrupt source priority register 004	IPR004	8	8		2 ICLK	section 14.
0008 7305h	ICU	Interrupt source priority register 005	IPR005	8	8		2 ICLK	section 14.
0008 7306h	ICU	Interrupt source priority register 006	IPR006	8	8		2 ICLK	section 14.
0008 7307h	ICU	Interrupt source priority register 007	IPR007	8	8		2 ICLK	section 14.
0008 7320h	ICU	Interrupt source priority register 032	IPR032	8	8		2 ICLK	section 14.
0008 7321h	ICU	Interrupt source priority register 033	IPR033	8	8		2 ICLK	section 14.
0008 7322h	ICU	Interrupt source priority register 034	IPR034	8	8		2 ICLK	section 14.
0008 732Ch	ICU	Interrupt source priority register 044	IPR044	8	8		2 ICLK	section 14.
0008 7339h	ICU	Interrupt source priority register 057	IPR057	8	8		2 ICLK	section 14.
0008 733Fh	ICU	Interrupt source priority register 063	IPR063	8	8		2 ICLK	section 14.
0008 7340h	ICU	Interrupt source priority register 064	IPR064	8	8		2 ICLK	section 14.
0008 7341h	ICU	Interrupt source priority register 065	IPR065	8	8		2 ICLK	section 14.
0008 7342h	ICU	Interrupt source priority register 066	IPR066	8	8		2 ICLK	section 14.
0008 7343h	ICU	Interrupt source priority register 067	IPR067	8	8		2 ICLK	section 14.
0008 7344h	ICU	Interrupt source priority register 068	IPR068	8	8		2 ICLK	section 14.
0008 7345h	ICU	Interrupt source priority register 069	IPR069	8	8		2 ICLK	section 14.
0008 7346h	ICU	Interrupt source priority register 070	IPR070	8	8		2 ICLK	section 14.
0008 7347h	ICU	Interrupt source priority register 071	IPR071	8	8		2 ICLK	section 14.
0008 7358h	ICU	Interrupt source priority register 088	IPR088	8	8		2 ICLK	section 14.
0008 7359h	ICU	Interrupt source priority register 089	IPR089	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (7 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 735Ch	ICU	Interrupt source priority register 092	IPR092	8	8		2 ICLK	section 14.
0008 735Dh	ICU	Interrupt source priority register 093	IPR093	8	8		2 ICLK	section 14.
0008 7366h	ICU	Interrupt source priority register 102	IPR102	8	8		2 ICLK	section 14.
0008 7367h	ICU	Interrupt source priority register 103	IPR103	8	8		2 ICLK	section 14.
0008 736Ah	ICU	Interrupt source priority register 106	IPR106	8	8		2 ICLK	section 14.
0008 7372h	ICU	Interrupt source priority register 114	IPR114	8	8		2 ICLK	section 14.
0008 7376h	ICU	Interrupt source priority register 118	IPR118	8	8		2 ICLK	section 14.
0008 7379h	ICU	Interrupt source priority register 121	IPR121	8	8		2 ICLK	section 14.
0008 737Bh	ICU	Interrupt source priority register 123	IPR123	8	8		2 ICLK	section 14.
0008 737Dh	ICU	Interrupt source priority register 125	IPR125	8	8		2 ICLK	section 14.
0008 737Fh	ICU	Interrupt source priority register 127	IPR127	8	8		2 ICLK	section 14.
0008 7381h	ICU	Interrupt source priority register 129	IPR129	8	8		2 ICLK	section 14.
0008 7385h	ICU	Interrupt source priority register 133	IPR133	8	8		2 ICLK	section 14.
0008 7386h	ICU	Interrupt source priority register 134	IPR134	8	8		2 ICLK	section 14.
0008 738Ah	ICU	Interrupt source priority register 138	IPR138	8	8		2 ICLK	section 14.
0008 738Bh	ICU	Interrupt source priority register 139	IPR139	8	8		2 ICLK	section 14.
0008 73AAh	ICU	Interrupt source priority register 170	IPR170	8	8		2 ICLK	section 14.
0008 73ABh	ICU	Interrupt source priority register 171	IPR171	8	8		2 ICLK	section 14.
0008 73AEh	ICU	Interrupt source priority register 174	IPR174	8	8		2 ICLK	section 14.
0008 73B1h	ICU	Interrupt source priority register 177	IPR177	8	8		2 ICLK	section 14.
0008 73B4h	ICU	Interrupt source priority register 180	IPR180	8	8		2 ICLK	section 14.
0008 73B7h	ICU	Interrupt source priority register 183	IPR183	8	8		2 ICLK	section 14.
0008 73C6h	ICU	Interrupt source priority register 198	IPR198	8	8		2 ICLK	section 14.
0008 73C7h	ICU	Interrupt source priority register 199	IPR199	8	8		2 ICLK	section 14.
0008 73C8h	ICU	Interrupt source priority register 200	IPR200	8	8		2 ICLK	section 14.
0008 73C9h	ICU	Interrupt source priority register 201	IPR201	8	8		2 ICLK	section 14.
0008 73DAh	ICU	Interrupt source priority register 218	IPR218	8	8		2 ICLK	section 14.
0008 73DEh	ICU	Interrupt source priority register 222	IPR222	8	8		2 ICLK	section 14.
0008 73E2h	ICU	Interrupt source priority register 226	IPR226	8	8		2 ICLK	section 14.
0008 73EAh	ICU	Interrupt source priority register 234	IPR234	8	8		2 ICLK	section 14.
0008 73EEh	ICU	Interrupt source priority register 238	IPR238	8	8		2 ICLK	section 14.
0008 73F2h	ICU	Interrupt source priority register 242	IPR242	8	8		2 ICLK	section 14.
0008 73F3h	ICU	Interrupt source priority register 243	IPR243	8	8		2 ICLK	section 14.
0008 73F4h	ICU	Interrupt source priority register 244	IPR244	8	8		2 ICLK	section 14.
0008 73F5h	ICU	Interrupt source priority register 245	IPR245	8	8		2 ICLK	section 14.
0008 73F6h	ICU	Interrupt source priority register 246	IPR246	8	8		2 ICLK	section 14.
0008 73F7h	ICU	Interrupt source priority register 247	IPR247	8	8		2 ICLK	section 14.
0008 73F8h	ICU	Interrupt source priority register 248	IPR248	8	8		2 ICLK	section 14.
0008 73F9h	ICU	Interrupt source priority register 249	IPR249	8	8		2 ICLK	section 14.
0008 7400h	ICU	DMAC activation request select register 0	DMRSR0	8	8		2 ICLK	section 14.
0008 7404h	ICU	DMAC activation request select register 1	DMRSR1	8	8		2 ICLK	section 14.
0008 7408h	ICU	DMAC activation request select register 2	DMRSR2	8	8		2 ICLK	section 14.
0008 740Ch	ICU	DMAC activation request select register 3	DMRSR3	8	8		2 ICLK	section 14.
0008 7500h	ICU	IRQ control register 0	IRQCR0	8	8		2 ICLK	section 14.
0008 7501h	ICU	IRQ control register 1	IRQCR1	8	8		2 ICLK	section 14.
0008 7502h	ICU	IRQ control register 2	IRQCR2	8	8		2 ICLK	section 14.
0008 7503h	ICU	IRQ control register 3	IRQCR3	8	8		2 ICLK	section 14.
0008 7504h	ICU	IRQ control register 4	IRQCR4	8	8		2 ICLK	section 14.
0008 7505h	ICU	IRQ control register 5	IRQCR5	8	8		2 ICLK	section 14.
0008 7506h	ICU	IRQ control register 6	IRQCR6	8	8		2 ICLK	section 14.
0008 7507h	ICU	IRQ control register 7	IRQCR7	8	8		2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (8 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 7510h	ICU	IRQ pin digital filter enable register 0	IRQFLTE0	8	8	2 ICLK		section 14.
0008 7514h	ICU	IRQ pin digital filter setting register 0	IRQFLTC0	16	16	2 ICLK		section 14.
0008 7580h	ICU	Non-maskable interrupt status register	NMISR	8	8	2 ICLK		section 14.
0008 7581h	ICU	Non-maskable interrupt enable register	NMIER	8	8	2 ICLK		section 14.
0008 7582h	ICU	Non-maskable interrupt clear register	NMICLR	8	8	2 ICLK		section 14.
0008 7583h	ICU	NMI pin interrupt control register	NMICR	8	8	2 ICLK		section 14.
0008 7590h	ICU	NMI pin digital filter enable register	NMIFLTE	8	8	2 ICLK		section 14.
0008 7594h	ICU	NMI pin digital filter setting register	NMIFLTC	8	8	2 ICLK		section 14.
0008 8000h	CMT	Compare match timer start register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8002h	CMT0	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8004h	CMT0	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8006h	CMT0	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8008h	CMT1	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 800Ah	CMT1	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 800Ch	CMT1	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8010h	CMT	Compare match timer start register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8012h	CMT2	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8014h	CMT2	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8016h	CMT2	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8018h	CMT3	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 801Ah	CMT3	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 801Ch	CMT3	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 24.
0008 8030h	IWDT	IWDT refresh register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	section 26.
0008 8032h	IWDT	IWDT control register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	section 26.
0008 8034h	IWDT	IWDT status register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	section 26.
0008 8036h	IWDT	IWDT reset control register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 26.
0008 8038h	IWDT	IWDT count stop control register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	section 26.
0008 8200h	TMR0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8201h	TMR1	Timer counter control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8202h	TMR0	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8203h	TMR1	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8204h	TMR0	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8205h	TMR1	Time constant register A	TCORA	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 8206h	TMR0	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8207h	TMR1	Time constant register B	TCORB	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 8208h	TMR0	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8209h	TMR1	Timer counter	TCNT	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 820Ah	TMR0	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 820Bh	TMR1	Timer counter control register	TCCR	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 820Ch	TMR0	Time count start register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8210h	TMR2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8211h	TMR3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8212h	TMR2	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8213h	TMR3	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8214h	TMR2	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8215h	TMR3	Time constant register A	TCORA	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 8216h	TMR2	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8217h	TMR3	Time constant register B	TCORB	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 8218h	TMR2	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8219h	TMR3	Timer counter	TCNT	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 821Ah	TMR2	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 23.

Table 5.1 List of I/O Registers (Address Order) (9 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 821Bh	TMR3	Timer counter control register	TCCR	8	8*1	2, 3 PCLKB	2 ICLK	section 23.
0008 821Ch	TMR2	Time count start register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 23.
0008 8280h	CRC	CRC control register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	section 31.
0008 8281h	CRC	CRC data input register	CRCDIR	8	8	2, 3 PCLKB	2 ICLK	section 31.
0008 8282h	CRC	CRC data output register	CRCDOR	16	16	2, 3 PCLKB	2 ICLK	section 31.
0008 8300h	RIIC0	I2C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8301h	RIIC0	I2C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8302h	RIIC0	I2C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8303h	RIIC0	I2C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8304h	RIIC0	I2C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8305h	RIIC0	I2C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8306h	RIIC0	I2C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8307h	RIIC0	I2C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8308h	RIIC0	I2C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8309h	RIIC0	I2C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Ah	RIIC0	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Ah	RIIC0	Timeout internal counter L	TMOCNTL	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Bh	RIIC0	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Bh	RIIC0	Timeout internal counter U	TMOCNTU	8	8*2	2, 3 PCLKB	2 ICLK	section 29.
0008 830Ch	RIIC0	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Dh	RIIC0	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Eh	RIIC0	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 830Fh	RIIC0	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8310h	RIIC0	I2C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8311h	RIIC0	I2C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8312h	RIIC0	I2C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8313h	RIIC0	I2C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 29.
0008 8380h	RSPI0	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8381h	RSPI0	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8382h	RSPI0	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8383h	RSPI0	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8384h	RSPI0	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	section 30.
0008 8388h	RSPI0	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8389h	RSPI0	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Ah	RSPI0	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Bh	RSPI0	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Ch	RSPI0	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Dh	RSPI0	RSPI slave select negation delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Eh	RSPI0	RSPI next-access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 838Fh	RSPI0	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK	section 30.
0008 8390h	RSPI0	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 8392h	RSPI0	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 8394h	RSPI0	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 8396h	RSPI0	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 8398h	RSPI0	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 839Ah	RSPI0	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 839Ch	RSPI0	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 839Eh	RSPI0	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	section 30.
0008 8410h	IRDA	IrDA control register	IRCR	8	8	2, 3 PCLKB	2 ICLK	section 28.
0008 8600h	MTU3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8601h	MTU4	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 21.

Table 5.1 List of I/O Registers (Address Order) (10 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 8602h	MTU3	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8603h	MTU4	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8604h	MTU3	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8605h	MTU3	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8606h	MTU4	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8607h	MTU4	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8608h	MTU3	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8609h	MTU4	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 860Ah	MTU	Timer output master enable register	TOER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 860Dh	MTU	Timer gate control register	TGCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 860Eh	MTU	Timer output control register 1	TOCR1	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 860Fh	MTU	Timer output control register 2	TOCR2	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8610h	MTU3	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8612h	MTU4	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8614h	MTU	Timer cycle data register	TCDR	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8616h	MTU	Timer dead time data register	TDDR	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8618h	MTU3	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 861Ah	MTU3	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 861Ch	MTU4	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 861Eh	MTU4	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8620h	MTU	Timer subcounter	TCNTS	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8622h	MTU	Timer cycle buffer register	TCBR	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8624h	MTU3	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8626h	MTU3	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8628h	MTU4	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 862Ah	MTU4	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 862Ch	MTU3	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 862Dh	MTU4	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8630h	MTU	Timer interrupt skipping set register	TITCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8631h	MTU	Timer interrupt skipping counter	TITCNT	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8632h	MTU	Timer buffer transfer set register	TBTER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8634h	MTU	Timer dead time enable register	TDER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8636h	MTU	Timer output level buffer register	TOLBR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8638h	MTU3	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8639h	MTU4	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8640h	MTU4	Timer A/D converter start request control register	TADCR	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8644h	MTU4	Timer A/D converter start request cycle set register A	TADCORA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8646h	MTU4	Timer A/D converter start request cycle set register B	TADCORB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8648h	MTU4	Timer A/D converter start request cycle set buffer register A	TADCOBRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 864Ah	MTU4	Timer A/D converter start request cycle set buffer register B	TADCOBRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8660h	MTU	Timer waveform control register	TWCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8680h	MTU	Timer start register	TSTR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8681h	MTU	Timer synchronous register	TSYR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8684h	MTU	Timer read/write enable register	TRWER	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8690h	MTU0	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8691h	MTU1	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8692h	MTU2	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8693h	MTU3	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8694h	MTU4	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8695h	MTU5	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK	section 21.
0008 8700h	MTU0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 21.

Table 5.1 List of I/O Registers (Address Order) (11 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 8701h	MTU0	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8702h	MTU0	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8703h	MTU0	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8704h	MTU0	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8705h	MTU0	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8706h	MTU0	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8708h	MTU0	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 870Ah	MTU0	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 870Ch	MTU0	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 870Eh	MTU0	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8720h	MTU0	Timer general register E	TGRE	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8722h	MTU0	Timer general register F	TGRF	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8724h	MTU0	Timer interrupt enable register 2	TIER2	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8726h	MTU0	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8780h	MTU1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8781h	MTU1	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8782h	MTU1	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8784h	MTU1	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8785h	MTU1	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8786h	MTU1	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8788h	MTU1	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 878Ah	MTU1	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8790h	MTU1	Timer input capture control register	TICCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8800h	MTU2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8801h	MTU2	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8802h	MTU2	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8804h	MTU2	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8805h	MTU2	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8806h	MTU2	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8808h	MTU2	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 880Ah	MTU2	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8880h	MTU5	Timer counter U	TCNTU	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8882h	MTU5	Timer general register U	TGRU	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8884h	MTU5	Timer control register U	TCRU	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8886h	MTU5	Timer I/O control register U	TIORU	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8890h	MTU5	Timer counter V	TCNTV	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8892h	MTU5	Timer general register V	TGRV	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 8894h	MTU5	Timer control register V	TCRV	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8896h	MTU5	Timer I/O control register V	TIORV	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 88A0h	MTU5	Timer counter W	TCNTW	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 88A2h	MTU5	Timer general register W	TGRW	16	16	2, 3 PCLKB	2 ICLK	section 21.
0008 88A4h	MTU5	Timer control register W	TCRW	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 88A6h	MTU5	Timer I/O control register W	TIORW	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 88B2h	MTU5	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 88B4h	MTU5	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 88B6h	MTU5	Timer compare match clear register	TCNTCMPCLR	8	8	2, 3 PCLKB	2 ICLK	section 21.
0008 8900h	POE	Input level control/status register 1	ICSR1	16	8, 16	2, 3 PCLKB	2 ICLK	section 22.
0008 8902h	POE	Output level control/status register 1	OCSR1	16	8, 16	2, 3 PCLKB	2 ICLK	section 22.
0008 8908h	POE	Input level control/status register 2	ICSR2	16	8, 16	2, 3 PCLKB	2 ICLK	section 22.
0008 890Ah	POE	Software port output enable register	SPOER	8	8	2, 3 PCLKB	2 ICLK	section 22.
0008 890Bh	POE	Port output enable control register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	section 22.

Table 5.1 List of I/O Registers (Address Order) (12 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 890Ch	POE	Port output enable control register 2	POECR2	8	8	2, 3 PCLKB	2 ICLK	section 22.
0008 890Eh	POE	Input level control/status register 3	ICSR3	16	8, 16	2, 3 PCLKB	2 ICLK	section 22.
0008 9000h	S12AD	A/D control register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9004h	S12AD	A/D channel select register A	ADANSA	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9008h	S12AD	A/D-converted value addition mode select register	ADADS	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 900Ch	S12AD	A/D-converted value addition count select register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 900Eh	S12AD	A/D control extended register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9010h	S12AD	A/D start trigger select register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9012h	S12AD	A/D converted extended input control register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9014h	S12AD	A/D channel select register B	ADANSB	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9018h	S12AD	A/D double register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 901Ch	S12AD	A/D internal reference voltage data register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 901Eh	S12AD	A/D self-diagnosis data register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9020h	S12AD	A/D data register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9022h	S12AD	A/D data register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9024h	S12AD	A/D data register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9026h	S12AD	A/D data register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9028h	S12AD	A/D data register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 902Ah	S12AD	A/D data register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 902Ch	S12AD	A/D data register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 902Eh	S12AD	A/D data register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9030h	S12AD	A/D data register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9032h	S12AD	A/D data register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9034h	S12AD	A/D data register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9036h	S12AD	A/D data register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9038h	S12AD	A/D data register 12	ADDR12	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 903Ah	S12AD	A/D data register 13	ADDR13	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 903Ch	S12AD	A/D data register 14	ADDR14	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 903Eh	S12AD	A/D data register 15	ADDR15	16	16	2, 3 PCLKB	2 ICLK	section 32.
0008 9060h	S12AD	A/D sampling state register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9061h	S12AD	A/D sampling state register L	ADSSTRL	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9071h	S12AD	A/D sampling state register O	ADSSTRO	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9073h	S12AD	A/D sampling state register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9074h	S12AD	A/D sampling state register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9075h	S12AD	A/D sampling state register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9076h	S12AD	A/D sampling state register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9077h	S12AD	A/D sampling state register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9078h	S12AD	A/D sampling state register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 9079h	S12AD	A/D sampling state register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 907Ah	S12AD	A/D disconnecting detection control register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 32.
0008 A020h	SCI1	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A021h	SCI1	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A022h	SCI1	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A023h	SCI1	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A024h	SCI1	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A025h	SCI1	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A026h	SCI1	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A027h	SCI1	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A028h	SCI1	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A029h	SCI1	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A02Ah	SCI1	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 27.

Table 5.1 List of I/O Registers (Address Order) (13 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 A02Bh	SCI1	I²C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A02Ch	SCI1	I²C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A02Dh	SCI1	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A0h	SCI5	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A1h	SCI5	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A2h	SCI5	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A3h	SCI5	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A4h	SCI5	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A5h	SCI5	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A6h	SCI5	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A7h	SCI5	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A8h	SCI5	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0A9h	SCI5	I²C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0AAh	SCI5	I²C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0ABh	SCI5	I²C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0ACh	SCI5	I²C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0ADh	SCI5	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C0h	SCI6	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C1h	SCI6	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C2h	SCI6	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C3h	SCI6	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C4h	SCI6	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C5h	SCI6	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C6h	SCI6	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C7h	SCI6	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C8h	SCI6	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0C9h	SCI6	I²C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0CAh	SCI6	I²C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0CBh	SCI6	I²C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0CCh	SCI6	I²C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A0CDh	SCI6	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A120h	SCI9	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A121h	SCI9	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A122h	SCI9	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A123h	SCI9	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A124h	SCI9	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A125h	SCI9	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A126h	SCI9	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A127h	SCI9	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A128h	SCI9	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A129h	SCI9	I²C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A12Ah	SCI9	I²C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A12Bh	SCI9	I²C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A12Ch	SCI9	I²C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 A12Dh	SCI9	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B000h	CAC	CAC control register 0	CACR0	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 B001h	CAC	CAC control register 1	CACR1	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 B002h	CAC	CAC control register 2	CACR2	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 B003h	CAC	CAC interrupt control register	CAICR	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 B004h	CAC	CAC status register	CASTR	8	8	2, 3 PCLKB	2 ICLK	section 10.
0008 B006h	CAC	CAC upper-limit value setting register	CAULVR	16	16	2, 3 PCLKB	2 ICLK	section 10.

Table 5.1 List of I/O Registers (Address Order) (14 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 B008h	CAC	CAC lower-limit value setting register	CALLVR	16	16	2, 3 PCLKB	2 ICLK	section 10.
0008 B00Ah	CAC	CAC counter buffer register	CACNTBR	16	16	2, 3 PCLKB	2 ICLK	section 10.
0008 B080h	DOC	DOC control register	DOCR	8	8	2, 3 PCLKB	2 ICLK	section 34.
0008 B082h	DOC	DOC data input register	DODIR	16	16	2, 3 PCLKB	2 ICLK	section 34.
0008 B084h	DOC	DOC data setting register	DODSR	16	16	2, 3 PCLKB	2 ICLK	section 34.
0008 B100h	ELC	Event link control register	ELCR	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B102h	ELC	Event link setting register 1	ELSR1	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B103h	ELC	Event link setting register 2	ELSR2	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B104h	ELC	Event link setting register 3	ELSR3	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B105h	ELC	Event link setting register 4	ELSR4	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B10Bh	ELC	Event link setting register 10	ELSR10	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B10Dh	ELC	Event link setting register 12	ELSR12	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B110h	ELC	Event link setting register 15	ELSR15	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B113h	ELC	Event link setting register 18	ELSR18	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B115h	ELC	Event link setting register 20	ELSR20	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B117h	ELC	Event link setting register 22	ELSR22	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B119h	ELC	Event link setting register 24	ELSR24	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B11Ah	ELC	Event link setting register 25	ELSR25	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B11Fh	ELC	Event link option setting register A	ELOPA	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B120h	ELC	Event link option setting register B	ELOPB	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B122h	ELC	Event link option setting register D	ELOPD	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B123h	ELC	Port group setting register 1	PGR1	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B125h	ELC	Port group control register 1	PGC1	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B127h	ELC	Port buffer register 1	PDBF1	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B129h	ELC	Event link port setting register 0	PEL0	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B12Ah	ELC	Event link port setting register 1	PEL1	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B12Dh	ELC	Event link software event generation register	ELSEGR	8	8	2, 3 PCLKB	2 ICLK	section 18.
0008 B300h	SCI12	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B301h	SCI12	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B302h	SCI12	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B303h	SCI12	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B304h	SCI12	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B305h	SCI12	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B306h	SCI12	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B307h	SCI12	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B308h	SCI12	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B309h	SCI12	I^2C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B30Ah	SCI12	I^2C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B30Bh	SCI12	I^2C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B30Ch	SCI12	I^2C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B30Dh	SCI12	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B320h	SCI12	Extended serial mode enable register	ESMER	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B321h	SCI12	Control register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B322h	SCI12	Control register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B323h	SCI12	Control register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B324h	SCI12	Control register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B325h	SCI12	Port control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B326h	SCI12	Interrupt control register	ICR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B327h	SCI12	Status register	STR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B328h	SCI12	Status clear register	STCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B329h	SCI12	Control Field 0 data register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	section 27.

Table 5.1 List of I/O Registers (Address Order) (15 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 B32Ah	SCI12	Control Field 0 compare enable register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B32Bh	SCI12	Control Field 0 receive data register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B32Ch	SCI12	Primary control field 1 data register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B32Dh	SCI12	Secondary control field 1 data register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B32Eh	SCI12	Control field 1 compare enable register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B32Fh	SCI12	Control field 1 receive data register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B330h	SCI12	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B331h	SCI12	Timer mode register	TMR	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B332h	SCI12	Timer prescaler register	TPRE	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 B333h	SCI12	Timer count register	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27.
0008 C000h	PORT0	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C001h	PORT1	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C002h	PORT2	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C003h	PORT3	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C004h	PORT4	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C005h	PORT5	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C00Ah	PORTA	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C00Bh	PORTB	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C00Ch	PORTC	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C00Dh	PORTD	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C00Eh	PORTE	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C011h	PORTH	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C012h	PORTJ	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C020h	PORT0	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C021h	PORT1	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C022h	PORT2	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C023h	PORT3	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C024h	PORT4	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C025h	PORT5	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C02Ah	PORTA	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C02Bh	PORTB	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C02Ch	PORTC	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C02Dh	PORTD	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C02Eh	PORTE	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C031h	PORTH	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C032h	PORTJ	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C040h	PORT0	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C041h	PORT1	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C042h	PORT2	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.

Table 5.1 List of I/O Registers (Address Order) (16 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 C043h	PORT3	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C044h	PORT4	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C045h	PORT5	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C04Ah	PORTA	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C04Bh	PORTB	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C04Ch	PORTC	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C04Dh	PORTD	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C04Eh	PORTE	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C051h	PORTH	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C052h	PORTJ	Port input data register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	3 ICLK cycles when reading, 2 ICLK cycles when writing	section 19.
0008 C060h	PORT0	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C061h	PORT1	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.

Table 5.1 List of I/O Registers (Address Order) (17 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 C062h	PORT2	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C063h	PORT3	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C064h	PORT4	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C065h	PORT5	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C06Ah	PORTA	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C06Bh	PORTB	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C06Ch	PORTC	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C06Dh	PORTD	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C06Eh	PORTE	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C071h	PORTH	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C072h	PORTJ	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C082h	PORT1	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C083h	PORT1	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C085h	PORT2	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C086h	PORT3	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C087h	PORT3	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C094h	PORTA	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C095h	PORTA	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C096h	PORTB	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C097h	PORTB	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C098h	PORTC	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C099h	PORTC	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C09Ch	PORTE	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C0h	PORT0	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C1h	PORT1	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C2h	PORT2	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C3h	PORT3	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C4h	PORT4	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0C5h	PORT5	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0CAh	PORTA	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0CBh	PORTB	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0CCh	PORTC	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0CDh	PORTD	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0CEh	PORTE	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0D1h	PORTH	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0D2h	PORTJ	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0E1h	PORT1	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0EBh	PORTB	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C0ECh	PORTC	Drive capacity control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C11Fh	MPC	Write-protect register	PWPR	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C120h	PORT	Port switching register B	PSRB	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C121h	PORT	Port switching register A	PSRA	8	8	2, 3 PCLKB	2 ICLK	section 19.
0008 C147h	MPC	P07 pin function control register	P07PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Ah	MPC	P12 pin function control register	P12PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Bh	MPC	P13 pin function control register	P13PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Ch	MPC	P14 pin function control register	P14PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Dh	MPC	P15 pin function control register	P15PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Eh	MPC	P16 pin function control register	P16PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C14Fh	MPC	P17 pin function control register	P17PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C150h	MPC	P20 pin function control register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C151h	MPC	P21 pin function control register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.

Table 5.1 List of I/O Registers (Address Order) (18 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 C152h	MPC	P22 pin function control register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C153h	MPC	P23 pin function control register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C154h	MPC	P24 pin function control register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C155h	MPC	P25 pin function control register	P25PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C156h	MPC	P26 pin function control register	P26PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C157h	MPC	P27 pin function control register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C158h	MPC	P30 pin function control register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C159h	MPC	P31 pin function control register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C15Ah	MPC	P32 pin function control register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C15Bh	MPC	P33 pin function control register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C15Ch	MPC	P34 pin function control register	P34PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C160h	MPC	P40 pin function control register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C161h	MPC	P41 pin function control register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C162h	MPC	P42 pin function control register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C163h	MPC	P43 pin function control register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C164h	MPC	P44 pin function control register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C165h	MPC	P45 pin function control register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C166h	MPC	P46 pin function control register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C167h	MPC	P47 pin function control register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C16Ch	MPC	P54 pin function control register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C16Dh	MPC	P55 pin function control register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C190h	MPC	PA0 pin function control register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C191h	MPC	PA1 pin function control register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C192h	MPC	PA2 pin function control register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C193h	MPC	PA3 pin function control register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C194h	MPC	PA4 pin function control register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C195h	MPC	PA5 pin function control register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C196h	MPC	PA6 pin function control register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C197h	MPC	PA7 pin function control register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C198h	MPC	PB0 pin function control register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C199h	MPC	PB1 pin function control register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Ah	MPC	PB2 pin function control register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Bh	MPC	PB3 pin function control register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Ch	MPC	PB4 pin function control register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Dh	MPC	PB5 pin function control register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Eh	MPC	PB6 pin function control register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C19Fh	MPC	PB7 pin function control register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A0h	MPC	PC0 pin function control register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A1h	MPC	PC1 pin function control register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A2h	MPC	PC2 pin function control register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A3h	MPC	PC3 pin function control register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A4h	MPC	PC4 pin function control register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A5h	MPC	PC5 pin function control register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A6h	MPC	PC6 pin function control register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A7h	MPC	PC7 pin function control register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A8h	MPC	PD0 pin function control register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1A9h	MPC	PD1 pin function control register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1AAh	MPC	PD2 pin function control register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1ABh	MPC	PD3 pin function control register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1ACh	MPC	PD4 pin function control register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1ADh	MPC	PD5 pin function control register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.

Table 5.1 List of I/O Registers (Address Order) (19 / 20)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 C1AEh	MPC	PD6 pin function control register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1AFh	MPC	PD7 pin function control register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B0h	MPC	PE0 pin function control register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B1h	MPC	PE1 pin function control register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B2h	MPC	PE2 pin function control register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B3h	MPC	PE3 pin function control register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B4h	MPC	PE4 pin function control register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B5h	MPC	PE5 pin function control register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B6h	MPC	PE6 pin function control register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1B7h	MPC	PE7 pin function control register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1C8h	MPC	PH0 pin function control register	PH0PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1C9h	MPC	PH1 pin function control register	PH1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1CAh	MPC	PH2 pin function control register	PH2PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1CBh	MPC	PH3 pin function control register	PH3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1D1h	MPC	PJ1 pin function control register	PJ1PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C1D3h	MPC	PJ3 pin function control register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK	section 20.
0008 C28Fh	SYSTEM	Flash HOCO software standby control register	FHSSBYCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 11.
0008 C290h	SYSTEM	Reset status register 0	RSTSR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 6.
0008 C291h	SYSTEM	Reset status register 1	RSTSR1	8	8	4, 5 PCLKB	2, 3 ICLK	section 6.
0008 C293h	SYSTEM	Main clock oscillator forced oscillation control register	MOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9.
0008 C294h	SYSTEM	High-speed clock oscillator power supply control register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9.
0008 C296h	FLASH	Flash write erase protection register	FWEPROR	8	8	4, 5 PCLKB	2, 3 ICLK	section 36., section 37.
0008 C297h	SYSTEM	Voltage monitoring circuit/comparator A control register	LVCMPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 8., section 33.
0008 C298h	SYSTEM	Voltage detection level select register	LVDLVLR	8	8	4, 5 PCLKB	2, 3 ICLK	section 8., section 33.
0008 C29Ah	SYSTEM	Voltage monitoring 1 circuit/comparator A1 control register 0	LVD1CR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 8., section 33.
0008 C29Bh	SYSTEM	Voltage monitoring 2 circuit/comparator A2 control register 0	LVD2CR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 8., section 33.
0008 C400h	RTC	64-Hz counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C402h	RTC	Second counter/Binary counter 0	RSECCNT/ BCNT0	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C404h	RTC	Minute counter/Binary counter 1	RMINCNT/ BCNT1	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C406h	RTC	Hour counter/Binary counter 2	RHRCNT/ BCNT2	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C408h	RTC	Day-of-week counter/Binary counter 3	RWKCNT/ BCNT3	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C40Ah	RTC	Date counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C40Ch	RTC	Month counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C40Eh	RTC	Year counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK	section 25.
0008 C410h	RTC	Second alarm register/Binary counter 0 alarm register	RSECAR/ BCNT0AR	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C412h	RTC	Minute alarm register/Binary counter 1 alarm register	RMINAR/ BCNT1AR	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C414h	RTC	Hour alarm register/Binary counter 2 alarm register	RHRAR/ BCNT2AR	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C416h	RTC	Day-of-week alarm register/Binary counter 3 alarm register	RWKAR/ BCNT3AR	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C418h	RTC	Date alarm register/Binary counter 0 alarm enable register	RDAYAR/ BCNT0AER	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C41Ah	RTC	Month alarm register/Binary counter 1 alarm enable register	RMONAR/ BCNT1AER	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C41Ch	RTC	Year alarm register/Binary counter 2 alarm enable register	RYRAR/ BCNT2AER	16	16	2, 3 PCLKB	2 ICLK	section 25.
0008 C41Eh	RTC	Year alarm enable register/Binary counter 3 alarm enable register	RYRAREN/ BCNT3AER	8	8	2, 3 PCLKB	2 ICLK	section 25.

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles: ICLK ≥ PCLK	Number of Access Cycles: ICLK < PCLK	Reference Section
0008 C422h	RTC	RTC control register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C424h	RTC	RTC control register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C426h	RTC	RTC control register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	section 25.
0008 C42Eh	RTC	Time error adjustment register	RADJ	8	8	2, 3 PCLKB	2 ICLK	section 25.
007F C402h	FLASH	Flash mode register	FMODR	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F C410h	FLASH	Flash access status register	FASTAT	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F C411h	FLASH	Flash access error interrupt enable register	FAEINT	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F C412h	FLASH	Flash ready interrupt enable register	FRDYIE	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F C440h	FLASH	E2 DataFlash read enable register 0	DFLRE0	16	16	2, 3 FCLK	2 ICLK	section 37.
007F C450h	FLASH	E2 DataFlash programming/erasure enable register 0	DFLWE0	16	16	2, 3 FCLK	2 ICLK	section 37.
007F FFB0h	FLASH	Flash status register 0	FSTATR0	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F FFB1h	FLASH	Flash status register 1	FSTATR1	8	8	2, 3 FCLK	2 ICLK	section 36., section 37.
007F FFB2h	FLASH	Flash P/E mode entry register	FENTRYR	16	16	2, 3 FCLK	2 ICLK	section 36., section 37.
007F FFB4h	FLASH	Flash protection register	FPROTR	16	16	2, 3 FCLK	2 ICLK	section 36.
007F FFB6h	FLASH	Flash reset register	FRESETR	16	16	2, 3 FCLK	2 ICLK	section 36.
007F FFBAh	FLASH	FCU command register	FCMDR	16	16	2, 3 FCLK	2 ICLK	section 36.
007F FFC8h	FLASH	FCU processing switching register	FCPSR	16	16	2, 3 FCLK	2 ICLK	section 36.
007F FFCAh	FLASH	E2 DataFlash blank check control register	DFLBCCNT	16	16	2, 3 FCLK	2 ICLK	section 37.
007F FFCCh	FLASH	Flash P/E status register	FPESTAT	16	16	2, 3 FCLK	2 ICLK	section 36., section 37.
007F FFCEh	FLASH	E2 DataFlash blank check status register	DFLBCSTAT	16	16	2, 3 FCLK	2 ICLK	section 37.
007F FFE8h	FLASH	Peripheral clock notification register	PCKAR	16	16	2, 3 FCLK	2 ICLK	section 36., section 37.

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 23.4 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. Table 29.3 lists register allocation for 16-bit access.

6. Resets

6.1 Overview

There are seven types of resets: RES# pin reset, power-on reset, voltage monitoring 0 reset, voltage monitoring 1 reset, voltage monitoring 2 reset, independent watchdog timer reset, 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 monitored: VPOR)*1
Voltage monitoring 0 reset	VCC falls (voltage monitored: Vdet0)*1
Voltage monitoring 1 reset	VCC falls (voltage monitored: Vdet1)*1
Voltage monitoring 2 reset	VCC falls (voltage monitored: Vdet2)*1
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDAa) and section 38, 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 Initialized by Each Reset Source

Target to be Initialized	Reset Source						
	RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Independent Watchdog Timer Reset	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset
The power-on reset detect flag (RSTSR0.PORF)	○	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—*1	○	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.LVD0RF)	○	○	—	—	—	—	—
The independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	○	—	—	—	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	○	○	○	—	—	—	—
The voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	○	○	○	○	—	—	—
Registers related to voltage monitor function 1 (LVD1CR0, LVCMPCR.EXVREFINP1,EXVCCINP1,LVD1E, LVDLVLR.LVD1LVL[3:0])	○	○	○	○	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	—	—	—
The voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	—	—
Registers related to voltage monitor function 2 (LVD2CR0, LVCMPCR.EXVREFINP2,EXVCCINP2, LVD2E, LVDLVLR.LVD2LVL[3:0])	○	○	○	○	○	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	—	—
Registers related to main clock oscillator (MOFCR)	○	○	○	○	○	○	○
Register related to high-speed on-chip oscillator (HOCOPCR.HOCOPCNT)	○	○	○	○	○	○	○
Pin state	○	○	○	○	○	○	○
The software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	—
Register related to the realtime clock*2	—	—	—	—	—	—	—
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○

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

Note 1. Initialized at a power-on.

Note 2. Some control bits (RCR1.CIE, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of reset. For details on the target bits, refer to section 25, Realtime Clock (RTCc).

When a reset is canceled, the reset exception handling starts. For the reset exception handling, see section 13, Exception Handling.
Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

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

Note 1. The value after reset depends on the reset source.

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

Note 1. Only 0 can be written to clear the flag.

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 resets shown in Table 6.2 occur.
- 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 resets listed in Table 6.2 occur.
- 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.

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

Note 1. The value after reset depends on the reset source.

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	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 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 at a power-on.
[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	—	IWDTRF
Value after reset:	0	0	0	0	0	0*1	0	0*1

Note 1. The value after reset depends on the reset source.

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) *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R(/W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

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.

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

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

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h

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

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

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

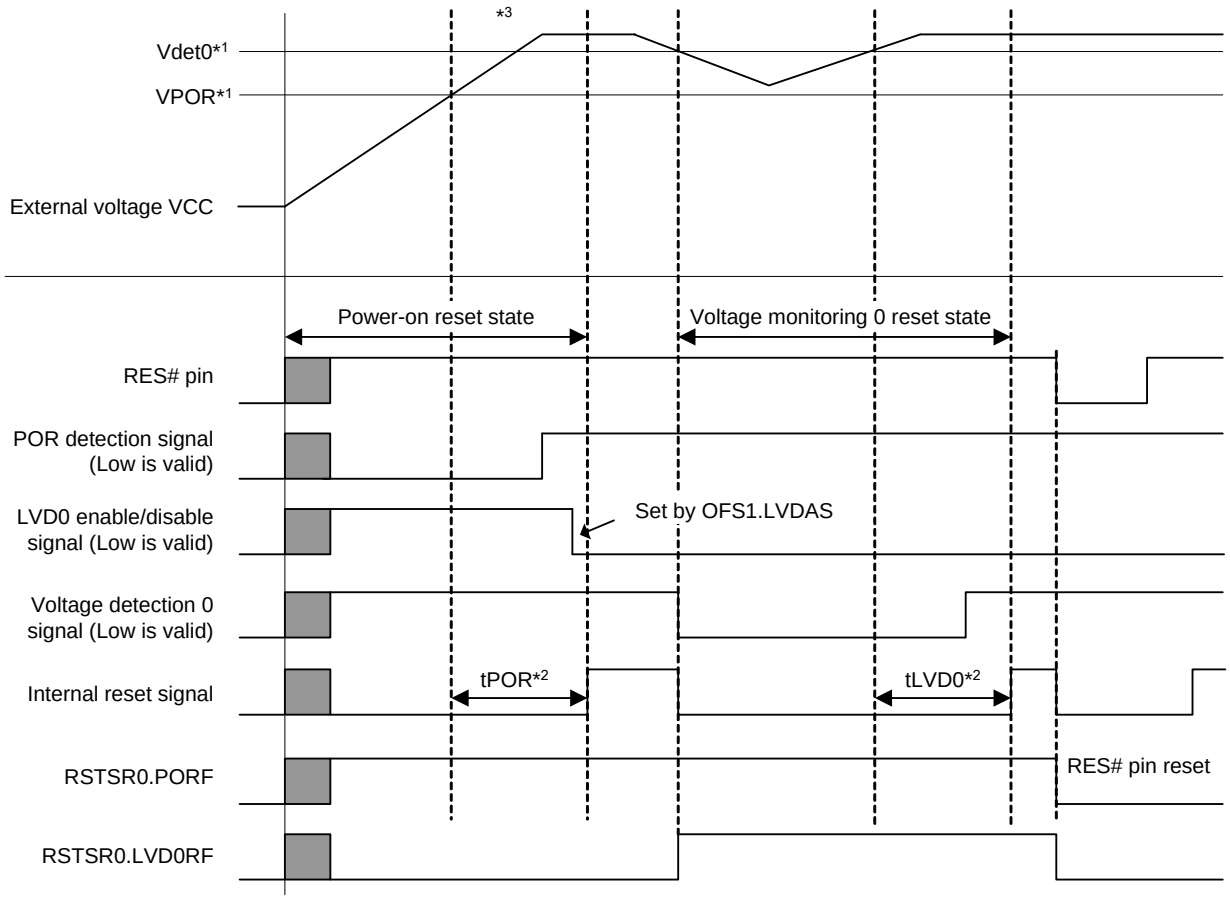
6.3 Operation

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 the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancellation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 38, 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 the LSI circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection circuit 0 start bit (LVDAS) in 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, and then the CPU starts the reset exception handling.
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 (LVDAa).



Note: • For details on the electrical characteristics, see section 38, Electrical Characteristics.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on 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 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.

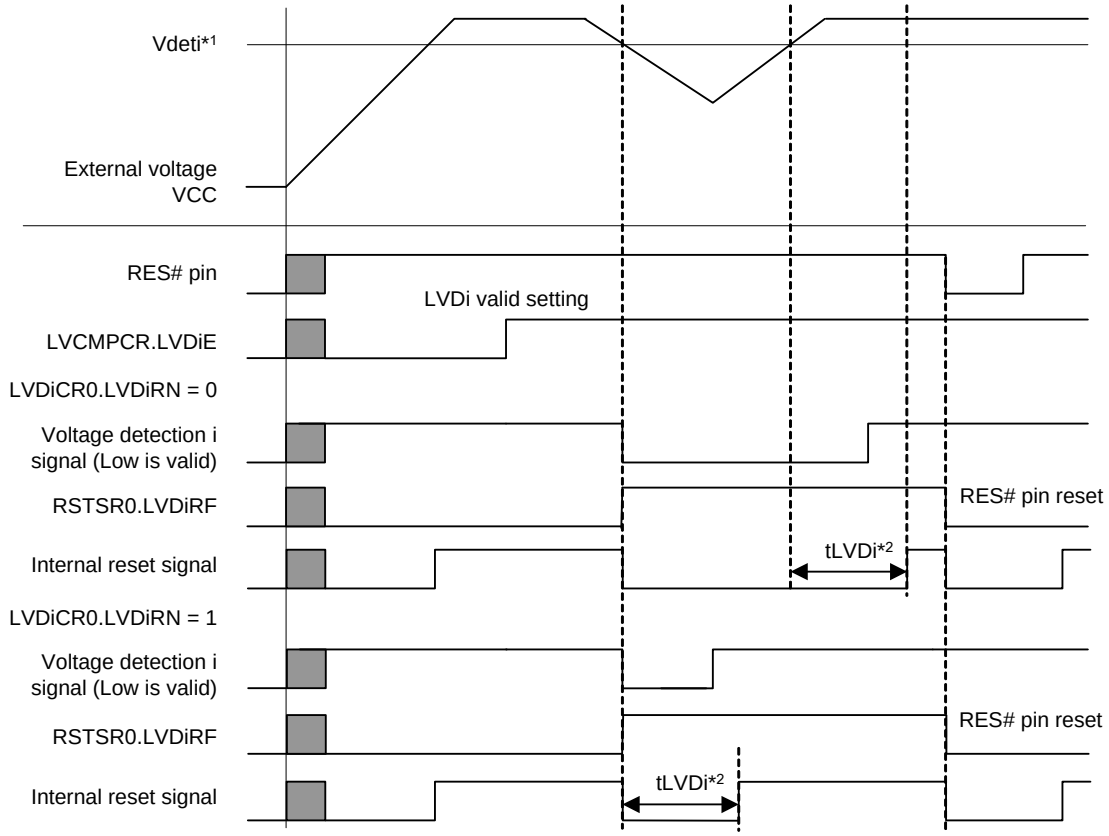
Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negation

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

Likewise, timing for release from the voltage monitoring 2 reset state is selectable by setting the voltage monitoring 2/ comparator A2 reset negation select bit (LVD2RN) in the LVD2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection level select register (LVDLVLR).

Figure 6.2 shows examples of operations during voltage monitoring 1 and 2 resets.

For details on the voltage monitoring 1 reset and voltage monitoring 2 reset, refer to section 8, Voltage Detection Circuit (LVDAa).



Note: • For details on the electrical characteristics, see section 38, Electrical Characteristics.
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 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 setting the IWDT reset control register (IWDTRCR) and 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 outside the refresh-permitted period. 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 26, Independent Watchdog Timer (IWDTa).

6.3.5 Software Reset

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

6.3.6 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.

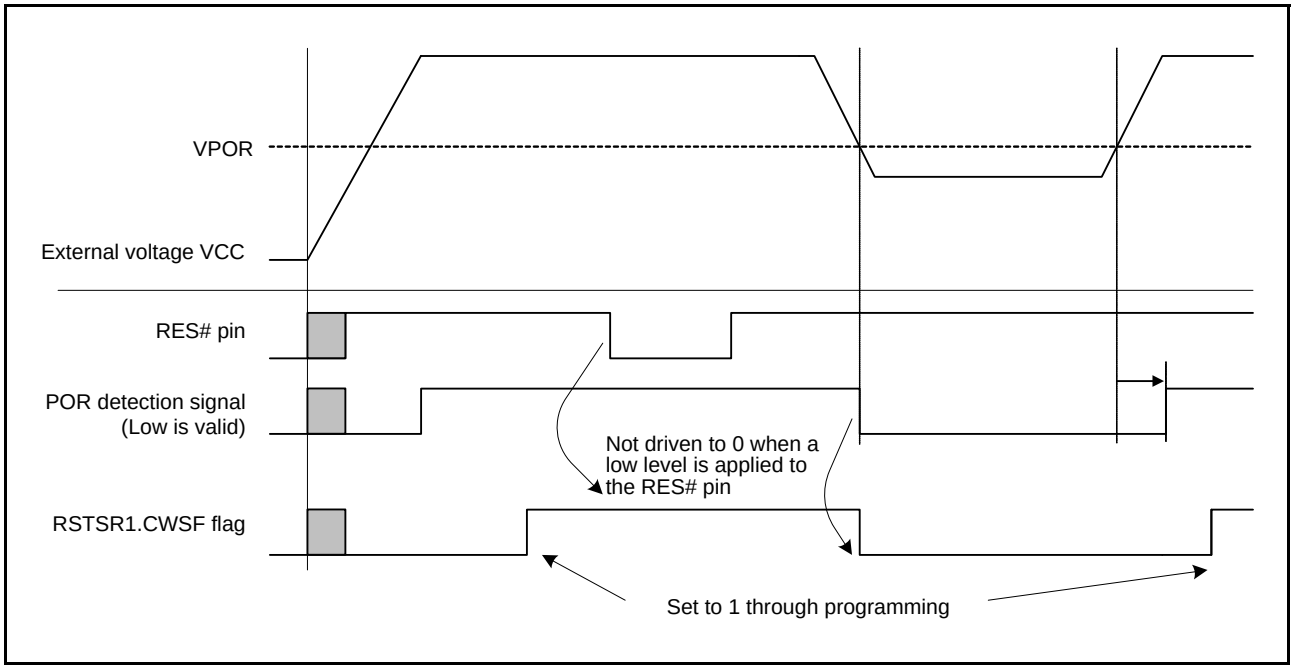


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.7 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.

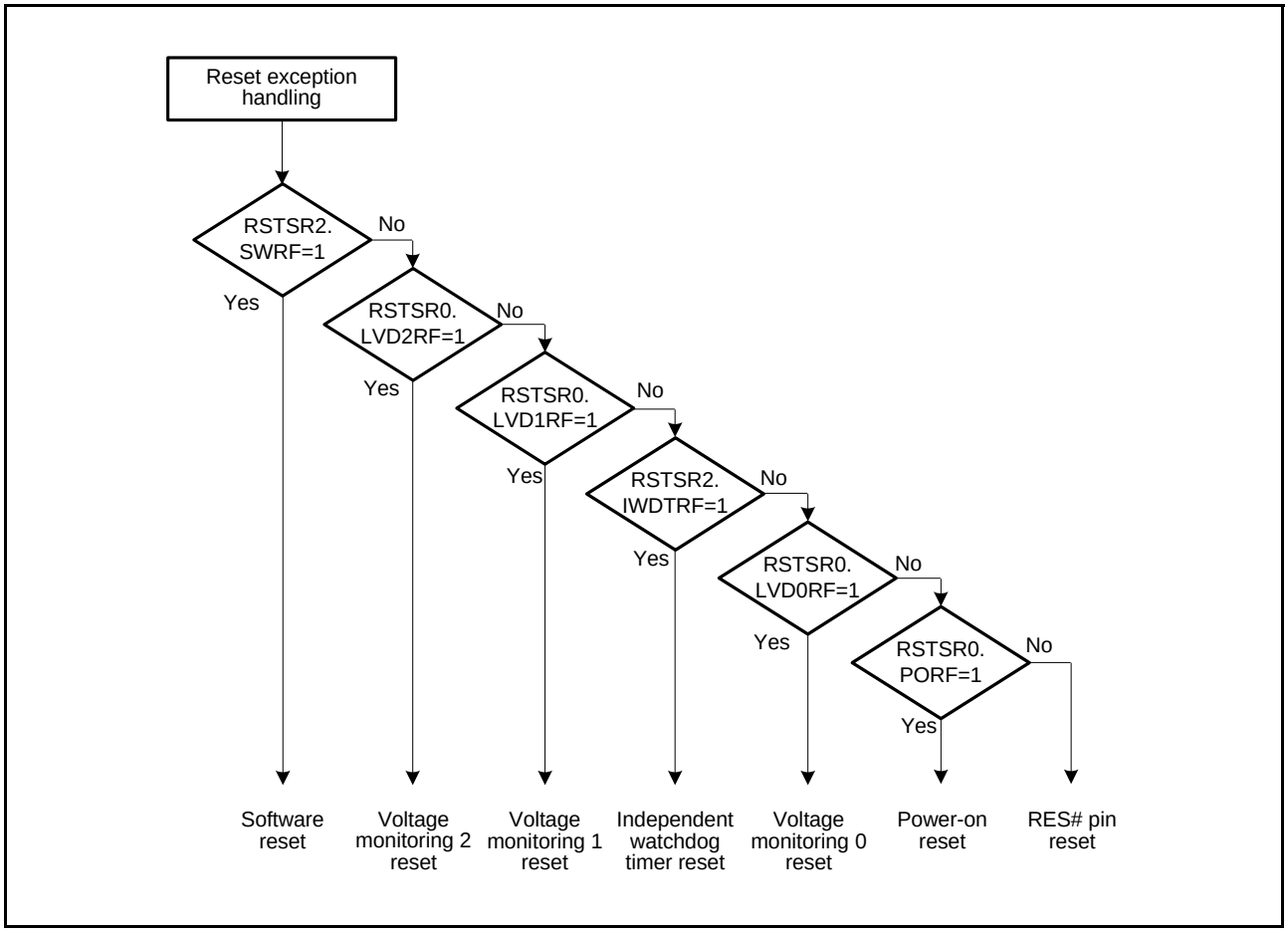


Figure 6.4 Example of Reset Generation Source Determination Flow

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.

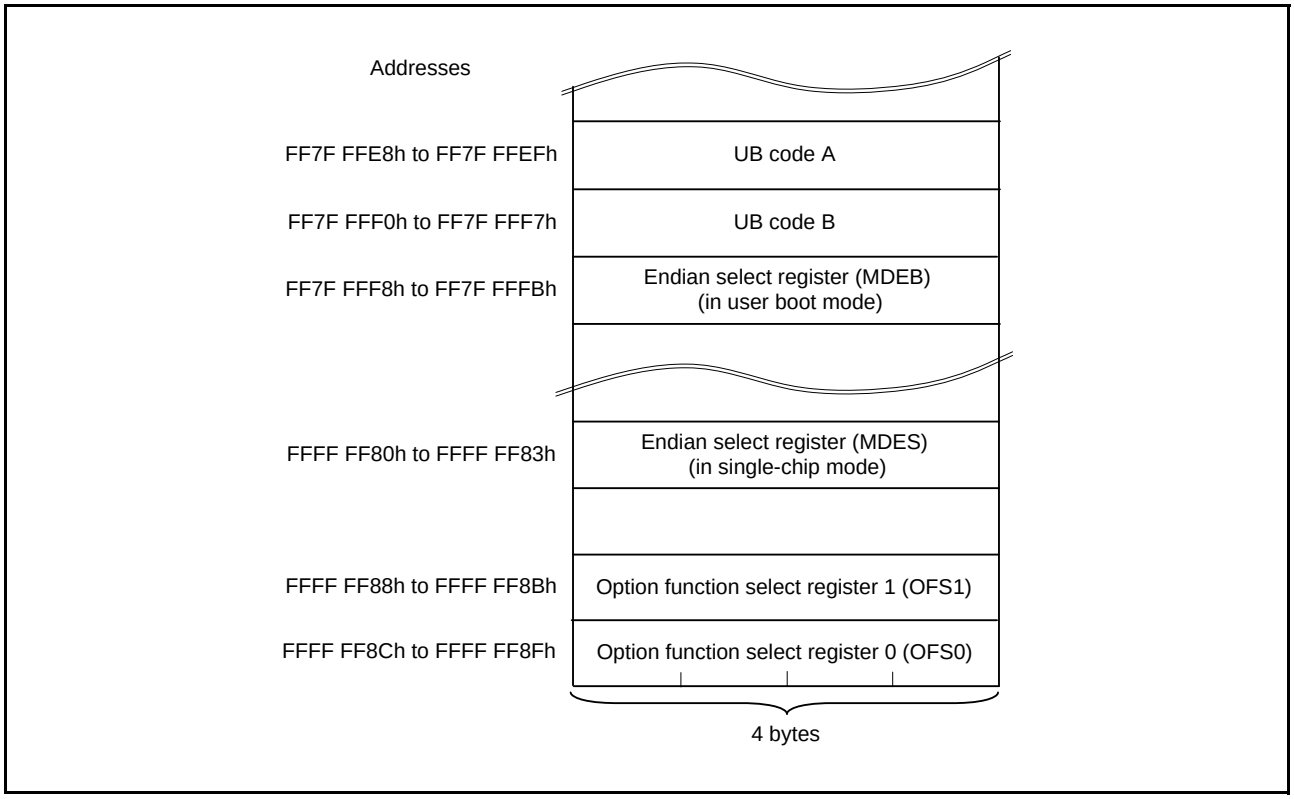


Figure 7.1 Option-Setting Memory Area

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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															

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT 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 the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby, or all-module clock stop mode	R
b31 to b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

The OFS0 register selects the operations of the independent watchdog timer (IWDT) after a reset.
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 IWDT-dedicated clock cycles) 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 26, 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 IWDT-dedicated clock. 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 IWDT-dedicated clock cycles.
For details, see section 26, 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 26, 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 26, 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 26, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects to stop counting when entering sleep, software standby, or all-module clock stop mode.
For details, see section 26, Independent Watchdog Timer (IWDTa).

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
Value after reset:	—	—	—	—	—	—	—	HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	
	The value set by the user*1															

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: 3.80 V is selected 0 1: 2.80 V is selected 1 0: 1.90 V is selected 1 1: 1.72 V is selected	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitoring 0 reset is enabled after a reset 1: Voltage monitoring 0 reset is disabled after a reset	R
b7 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

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.

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

These bits select the voltage detection level to be monitored by the voltage detection 0 circuit.

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitoring 0 reset is enabled or disabled after a reset.
The Vdet0 voltage to be monitored by the voltage detection 0 circuit is selected by the VDSEL[1:0] bits.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation enable bit is effective or not after a reset.
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the wait time for oscillation stabilization.
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.

7.2.3 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															

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

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, these bits return the value written by the user. The write value should be 1.	R

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-ship 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 Code

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 55736572h and 426F6F74h.
- UB code B is FFFFFF07h and 0008C04Ch.
- 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 55736572h and 426F6F74h. Do not set other values for the code.
Figure 7.2 shows the configuration of UB code A. Set UB code A in 32-bit units.

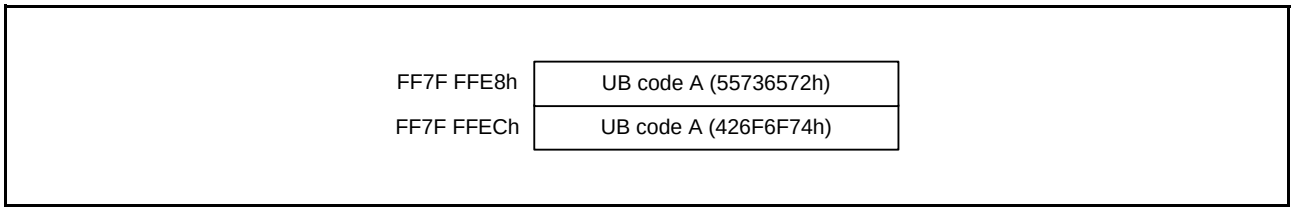


Figure 7.2 UB Code A Configuration

7.3.2 UB Code B

UB code B consists of two 32-bit words. Set UB code B to FFFFFF07h and 0008C04Ch. Do not set other values for the code.
Figure 7.3 shows the configuration of UB code B. Set UB code B in 32-bit units.

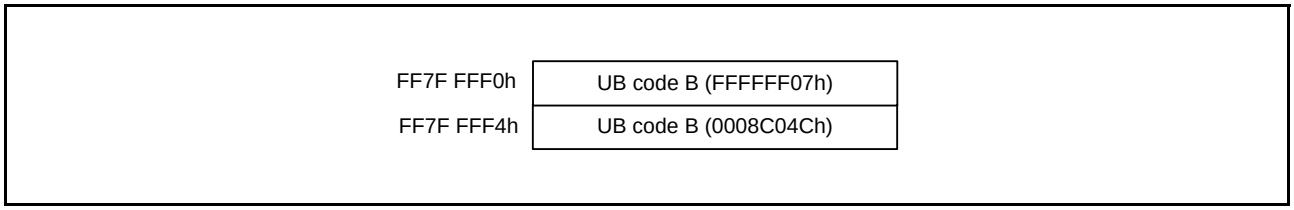


Figure 7.3 UB Code B Configuration

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 (LVDAa)

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

8.1 Overview

In voltage detection 0, the detection voltage can be selected from four levels by using option function select register 1 (OFS1).
In voltage detection 1 and voltage detection 2, the detection voltage can be selected from sixteen levels using the voltage detection level select register (LVDLVLR).
Voltage detection 2 can be switched between input voltages to VCC and the CMPA2 pin.
Reset of voltage monitoring 0, reset/interrupt of voltage monitoring 1, and reset/interrupt of voltage monitoring 2 can be used.
However, voltage monitoring 1 and comparator A1 cannot be used at the same time because they share the voltage detection circuit. Similarly, voltage monitoring 2 and comparator A2 cannot be used at the same time because they share the voltage detection circuit.

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 (1/2)

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
				Input voltages to VCC and the CMPA2 pin can be switched using the LVCMPCR.EXVCCINP2 bit
	Detection voltage	Voltage selectable from four levels using OFS1	Voltage selectable from 16 levels using LVDLVLR.LVD1LVL[3:0] bits	Varies according to whether VCC or the CMPA2 pin is selected. Voltage selectable from 16 levels 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	None	Voltage monitoring 1 interrupt	Voltage monitoring 2 interrupt
			Non-maskable or maskable interrupt is selectable	Non-maskable or maskable interrupt is selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Interrupt request issued when Vdet2 > VCC and VCC > Vdet2 or either
Digital filter	Enable/ Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)

Table 8.1 Voltage Detection Circuit Specifications (2/2)

Item	Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
Event link function	None	Available Vdet1 passage detection event output	None

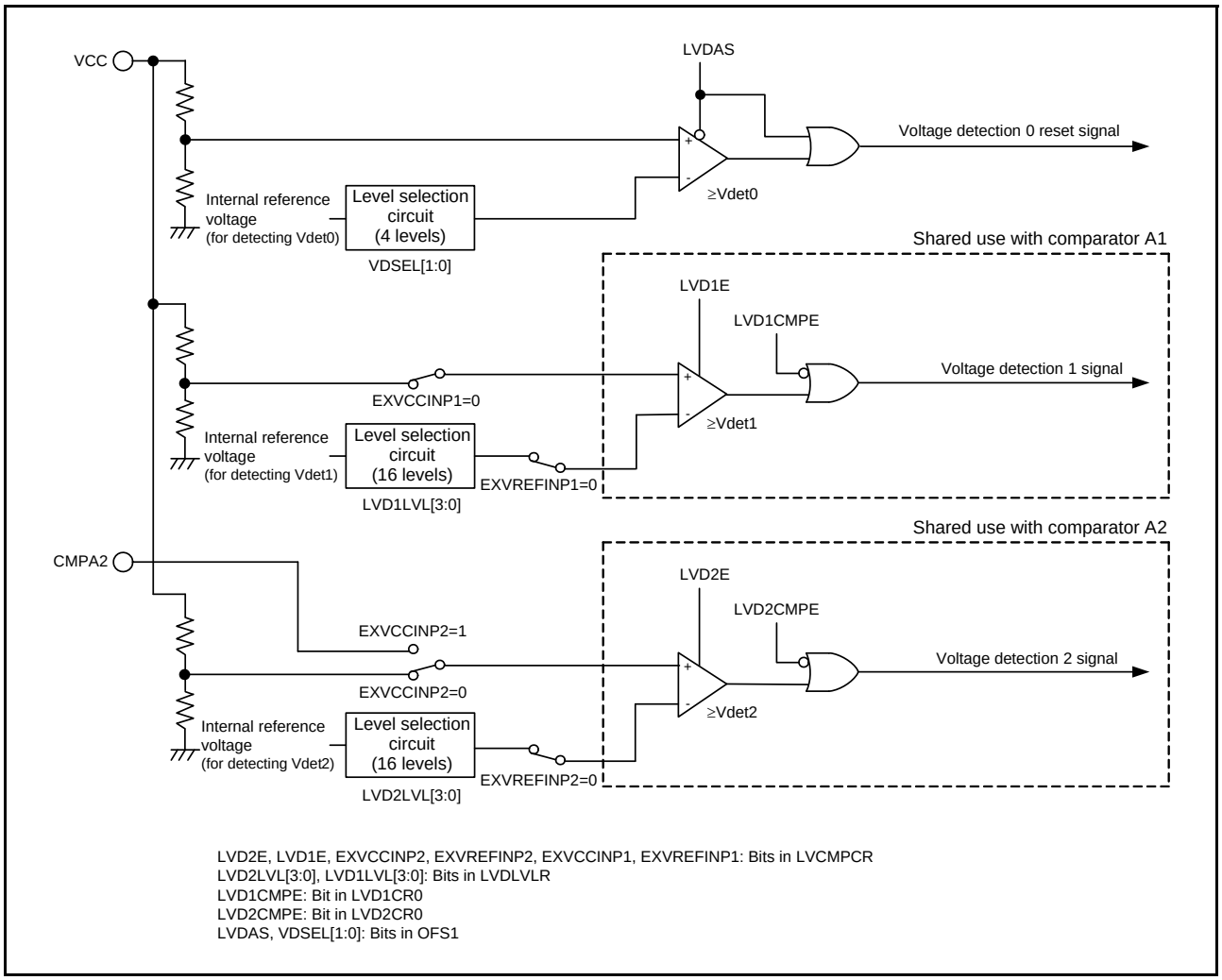


Figure 8.1 Block Diagram of Voltage Detection Circuit

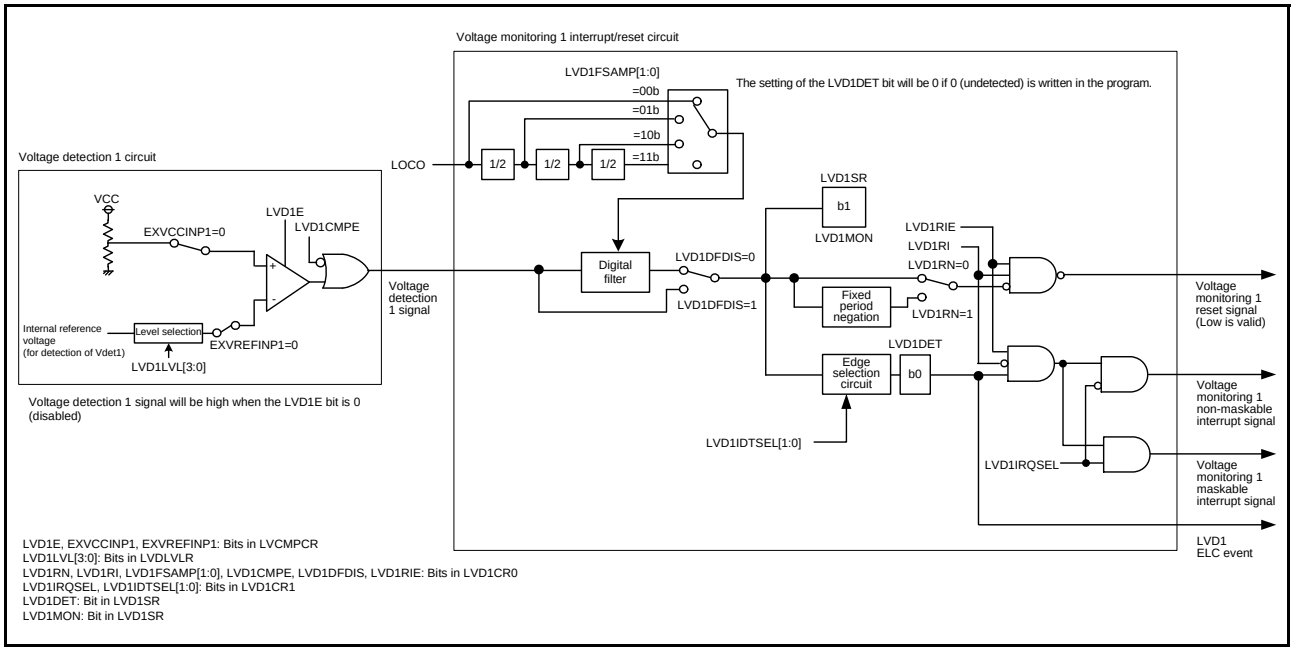


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

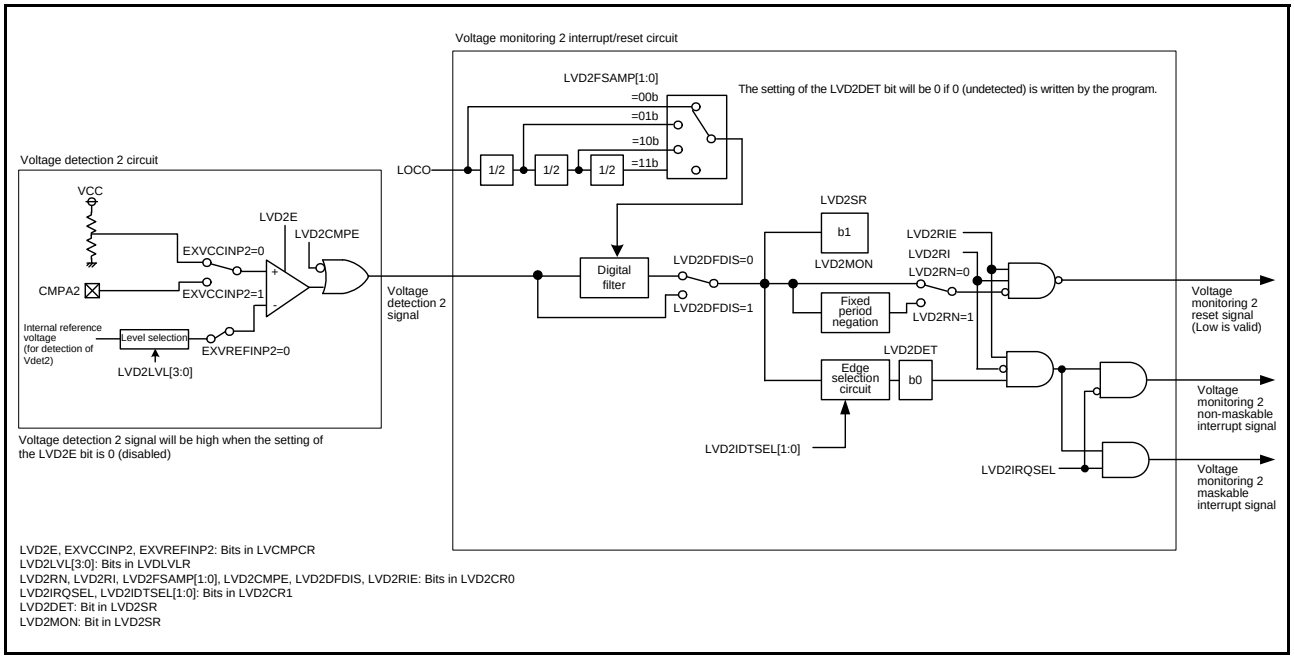


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

Table 8.2 lists the input/output pins relevant to the voltage detection circuit.

Table 8.2 Input/Output Pins of the Voltage Detection Circuit

Pin name	Pin name	Function
CMPA2	Input	Detection target voltage pin for voltage detection 2

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL [1:0]	Voltage Monitoring 1/Comparator A1 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: Do not set	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1/Comparator A1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	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/Comparator A1 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/Comparator A1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R(/W) *1
b1	LVD1MON	Voltage Monitoring 1/Comparator A1 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 system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1/Comparator A1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again after a period of PCLKB2 cycle or more has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, more cycles than PCLKB2 may have to be secured as wait time.

LVD1MON Flag (Voltage Monitoring 1/Comparator A1 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/Comparator A2 Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2/Comparator A2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet2 (rise) is detected 0 1: When VCC < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Do not set	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2/Comparator A2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	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/Comparator A2 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/Comparator A2 Voltage Change Detection Flag	0: Not detected 1: Vdet2 passage detection	R(/W) *1
b1	LVD2MON	Voltage Monitoring 2/Comparator A2 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 system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2/Comparator A2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again after a period of PCLKB2 cycle or more 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 wait time.

LVD2MON Flag (Voltage Monitoring 2/Comparator A2 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/Comparator A Control Register (LVCMPCR)

Address(es): 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	EXVCCINP2	EXVREFINP2	EXVCCINP1	EXVREFINP1
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXVREFINP1	Comparator A1 Reference Voltage External Input Select	0: Internal reference voltage Must be set to 0 when the device is to be used for an LVD.	R/W
b1	EXVCCINP1	Comparator A1 Comparison Voltage External Input Select	0: Power supply voltage (VCC) Must be set to 0 when the device is to be used for an LVD.	R/W
b2	EXVREFINP2	Comparator A2 Reference Voltage External Input Select	0: Internal reference voltage Must be set to 0 when the device is to be used for an LVD.	R/W
b3	EXVCCINP2	Comparator A2 Comparison Voltage External Input Select	0: Source voltage (VCC) 1: CMPA2 pin input voltage*1	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1/Comparator A1 Enable	0: Voltage detection 1/comparator A1 circuit disabled 1: Voltage detection 1/comparator A1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2/Comparator A2 Enable	0: Voltage detection 2/comparator A2 circuit disabled 1: Voltage detection 2/comparator A2 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.
Note 1. The LVDLVLR.LVD2LVL[3:0] bits must be set to 0001b when the EXVCCINP2 bit is to be set to 1 (CMPA2 pin input voltage).

LVD1E Bit (Voltage Detection 1/Comparator A1 Enable)

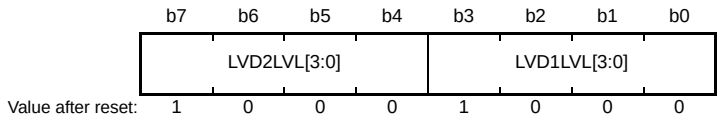
When using voltage detection 1/comparator A1 interrupt/reset or the LVD1SR.LVD1MON flag, set the LVD1E bit to 1. The voltage detection 1/comparator A1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

LVD2E Bit (Voltage Detection 2/Comparator A2 Enable)

When using voltage detection 2/comparator A2 interrupt/reset or the LVD2SR.LVD2MON flag, set the LVD2E bit to 1. The voltage detection 2/comparator A2 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



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 0 0 0 0: 4.15 V 0 0 0 1: 4.00 V 0 0 1 0: 3.85 V 0 0 1 1: 3.70 V 0 1 0 0: 3.55 V 0 1 0 1: 3.40 V 0 1 1 0: 3.25 V 0 1 1 1: 3.10 V 1 0 0 0: 2.95 V 1 0 0 1: 2.80 V 1 0 1 0: 2.65 V 1 0 1 1: 2.50 V 1 1 0 0: 2.35 V 1 1 0 1: 2.20 V 1 1 1 0: 2.05 V 1 1 1 1: 1.90 V	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	(When LVCMPCR.EXVCCINP2 = 0 (VCC select)) b7 b4 0 0 0 0: 4.15 V 0 0 0 1: 4.00 V 0 0 1 0: 3.85 V 0 0 1 1: 3.70 V 0 1 0 0: 3.55 V 0 1 0 1: 3.40 V 0 1 1 0: 3.25 V 0 1 1 1: 3.10 V 1 0 0 0: 2.95 V 1 0 0 1: 2.80 V 1 0 1 0: 2.65 V 1 0 1 1: 2.50 V 1 1 0 0: 2.35 V 1 1 0 1: 2.20 V 1 1 1 0: 2.05 V 1 1 1 1: 1.90 V (When LVCMPCR.EXVCCINP2 = 1 (CMPA2 pin select)) b7 b4 0 0 0 1: 1.33 V Settings other than above are prohibited.	R/W

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

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits to 0 (voltage detection n circuit disabled) (n = 1, 2).

Do not use the voltage detection 1 and 2 circuits at the same detection voltage level. When setting the detection voltage level of the voltage detection 0 circuit to 1.90 V, do not set the detection voltage level of the voltage detection 1 and 2 circuits to 1.90 V. When setting the detection voltage level of the voltage detection 0 circuit to 2.80 V, do not set the detection voltage level of the voltage detection 1 and 2 circuits to 2.80 V.

8.2.7 Voltage Monitoring 1 Circuit/Comparator A1 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/Comparator A1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD1DFDIS	Voltage Monitoring 1/Comparator A1 Digital Filter Disable Mode Select	0: Digital filter enable 1: Digital filter disable	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit/Comparator A1 Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison results output disable 1: Voltage monitoring 1 circuit comparison results output enable	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: LOCO divided by 1 0 1: LOCO divided by 2 1 0: LOCO divided by 4 1 1: LOCO divided by 8	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit/Comparator A1 Mode Select	0: Voltage monitoring 1 interrupt enabled when Vdet1 is crossed 1: Voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1/Comparator A1 Reset Negation 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/Comparator A1 Interrupt/Reset Enable)

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

LVD1DFDIS Bit (Voltage Monitoring 1/Comparator A1 Digital Filter Disable Mode Select)

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

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

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

LVD1RN Bit (Voltage Monitoring 1/Comparator A1 Reset Negation Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby mode 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 voltage monitoring 1 reset signal).

8.2.8 Voltage Monitoring 2 Circuit/Comparator A2 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/Comparator A2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD2DFDIS	Voltage Monitoring 2/Comparator A2 Digital Filter Disable Mode Select	0: Digital filter enable 1: Digital filter disable	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit/Comparator A2 Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison results output disable 1: Voltage monitoring 2 circuit comparison results output enable	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: LOCO divided by 1 0 1: LOCO divided by 2 1 0: LOCO divided by 4 1 1: LOCO divided by 8	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit/Comparator A2 Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2/Comparator A2 Reset Negation 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/Comparator A2 Interrupt/Reset Enable)

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

LVD2DFDIS Bit (Voltage Monitoring 2/Comparator A2 Digital Filter Disable Mode Select)

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

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

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

LVD2RN Bit (Voltage Monitoring 2/Comparator A2 Reset Negation Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby mode 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 voltage monitoring 2 reset signal).

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

Table 8.3 lists the procedures for setting up monitoring against Vdet1. After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

Table 8.3 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 (voltage detection 1 level select).	
2	Set the LVCMPCR.EXVREFINP1 bit to 0 (internal reference voltage). Set the LVCMPCR.EXVCCINP1 bit to 0 (VCC voltage).	
3	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).
4	Set the LVCMPCR.LVD1E bit to 1 (enabling the circuit for voltage detection 1).	
5	Wait for at least td(E-A) or longer.	
6	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by the voltage monitoring 1 circuit).	
7	Wait for at least one cycle of the LOCO.	—
8	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).	—
9	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).	— (No waiting is required.)

8.3.3 Monitoring Vdet2

Table 8.4 lists the procedures for setting up monitoring against Vdet2. After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

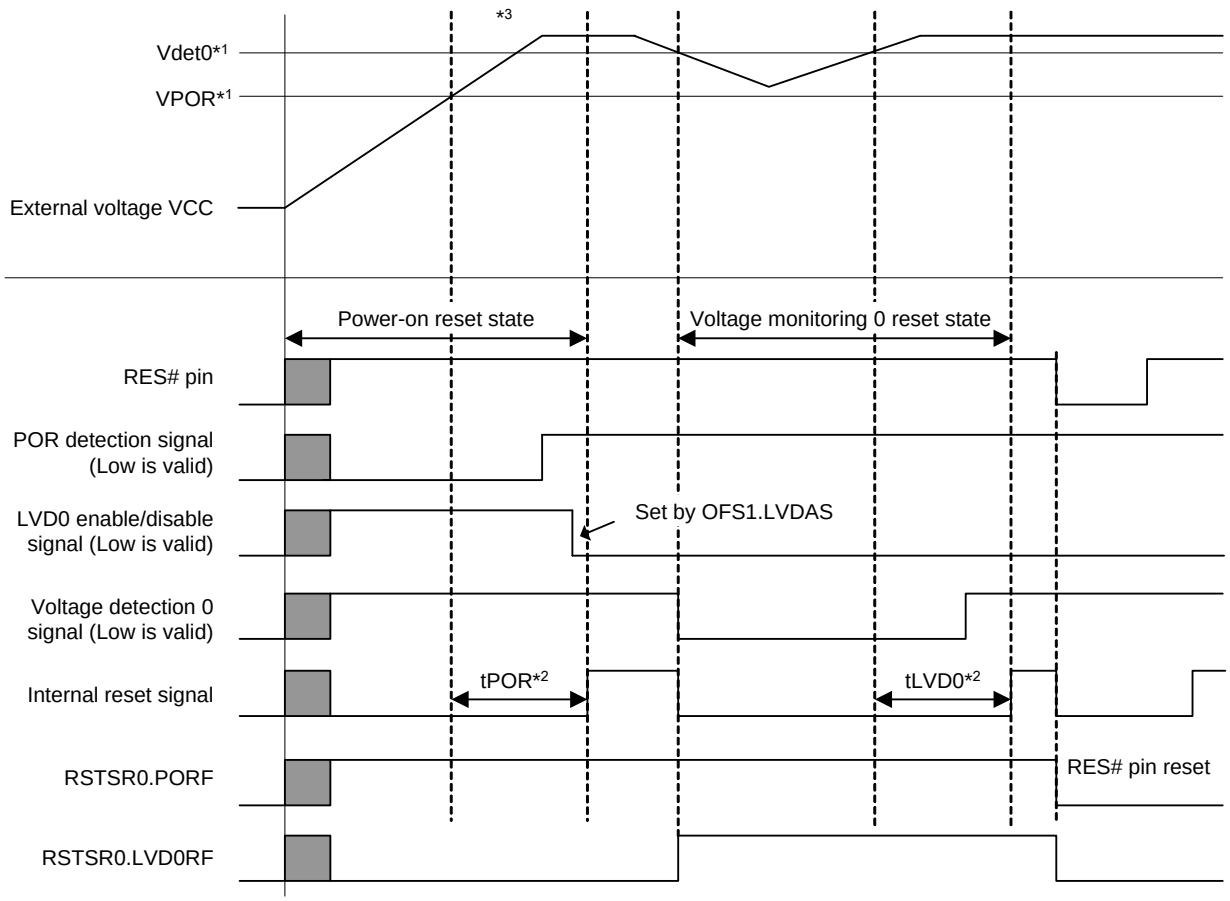
Table 8.4 Procedures for Setting up Monitoring against Vdet1

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.LVD2LVL[3:0] bits (voltage detection 2 level select).	
2	Set the LVCMPCR.EXVREFINP2 bit to 0 (internal reference voltage). Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (selecting the input voltage on the CMPA2 pin).	
3	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).
4	Set the LVCMPCR.LVD2E bit to 1 (enabling the circuit for voltage detection 2).	
5	Wait for at least td(E-A) or longer.	
6	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by the voltage monitoring 2 circuit).	
7	Wait for at least one cycle of the LOCO.	—
8	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).	—
9	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).	— (No waiting is required.)

8.4 Reset from Voltage Monitor 0

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

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



Note: • For details on the electrical characteristics, see section 38, Electrical Characteristics.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on 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 8.5 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.6 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. 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 to initiate release from software standby mode.

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

	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*2	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.			
2*2	Set the LVCMPCR.EXVREFINP1 bit to 0 (internal reference voltage). Set the LVCMPCR.EXVCCINP1 bit to 0 (VCC voltage).			
3*1	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).	
4 *1, *2	Clear the LVD1CR0.LVD1RI bit to 0 (to select the voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (to select the voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.	Clear the LVD1CR0.LVD1RI bit to 0 (to select the voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (to select the voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
5	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
6	—	Set the LVD1CR0.LVD1RIE bit to 1 (enabling voltage monitoring 1 interrupt/reset)	—	Set the LVD1CR0.LVD1RIE bit to 1 (enabling voltage monitoring 1 interrupt/reset)
7*2	Set the LVCMPCR.LVD1E bit to 1 (enabling the circuit for voltage detection 1).			
8*2	Wait for at least td(E-A) or longer.			
9	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by the voltage monitoring 1 circuit).			
10	Wait for at least one cycle of the LOCO.		—	
11	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).		—	
12	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).		— (No waiting is required.)	
13	Clear the LVD1SR.LVD1DET bit to 0.	—	Clear the LVD1SR.LVD1DET bit to 0.	—
14	Set the LVD1CR0.LVD1RIE bit to 1 (enabling voltage monitoring 1 interrupt/reset)	—	Set the LVD1CR0.LVD1RIE bit to 1 (enabling voltage monitoring 1 interrupt/reset)	—

Note 1. Executing steps 3 and 4 at the same time (with a single instruction) creates no problems.
Note 2. Steps 1, 2, 4, 7, and 8 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.LVD1DFDIS and LVD1FSAMP bits or LVD1CR1.LVD1IRQSEL and LVD1IDTSEL bits after monitoring is stopped 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 14.

Table 8.6 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1	Clear the LVD1CR0.LVD1RIE bit to 0 (disabling voltage monitoring 1 interrupt/reset).	—
2	Clear the LVD1CR0.LVD1CMPE bit to 0 (disabling output of the results of comparison by the voltage monitoring 1 circuit).	
3*1	Clear the LVCMPCR.LVD1E bit to 0 (disable the voltage monitoring 1 circuit).	
4	—	Clear the LVD1CR0.LVD1RIE bit to 0 (disabling voltage monitoring 1 interrupt/reset).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE.	

Note 1. Step 3 is 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.LVD1DFDIS and LVD1FSAMP bits or LVD1CR1.LVD1IRQSEL and LVD1IDTSEL bits after monitoring is stopped 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 5.

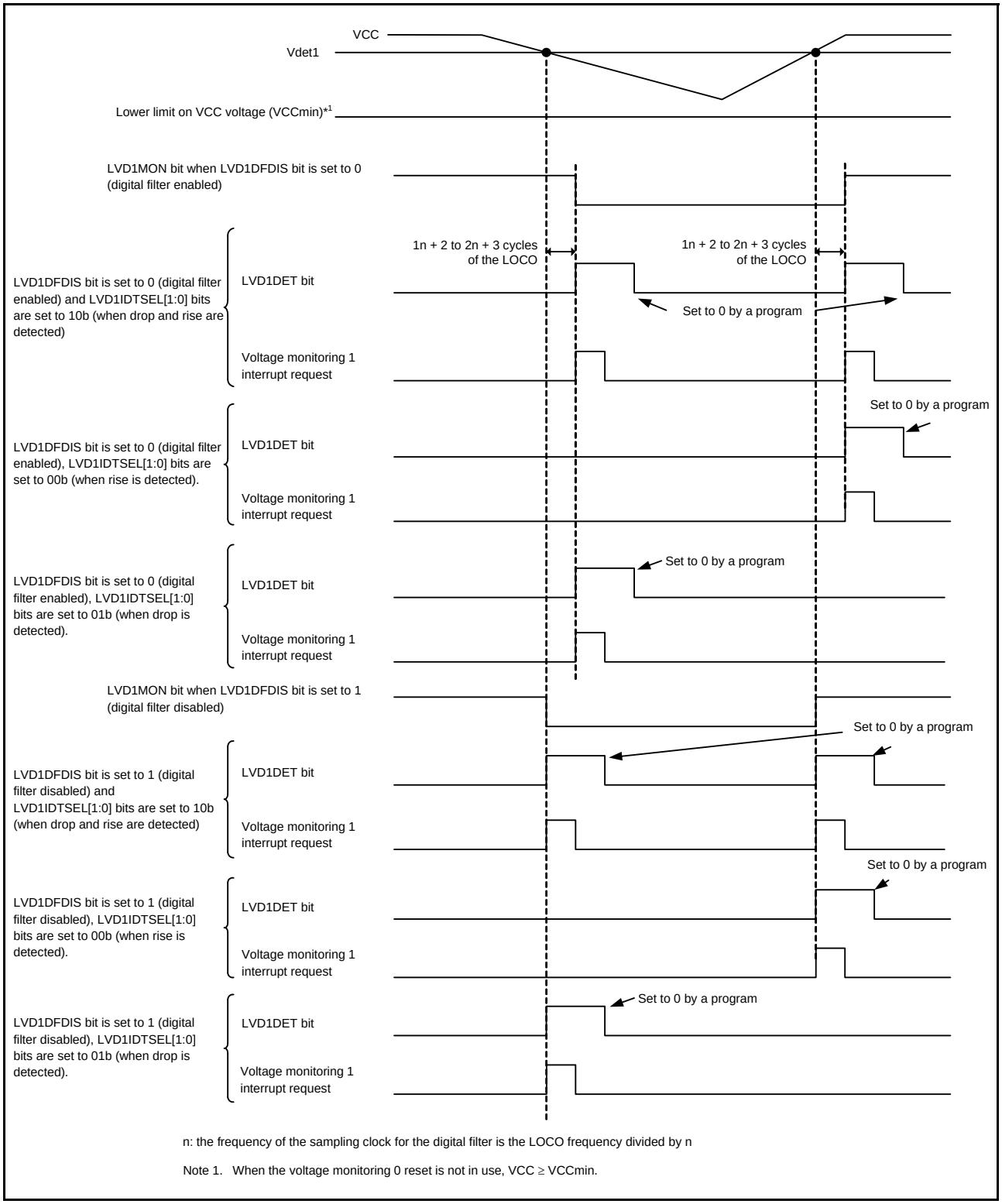


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitoring 2

Table 8.7 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Table 8.8 shows the procedure for stopping bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. 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 to initiate release from software standby mode.

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

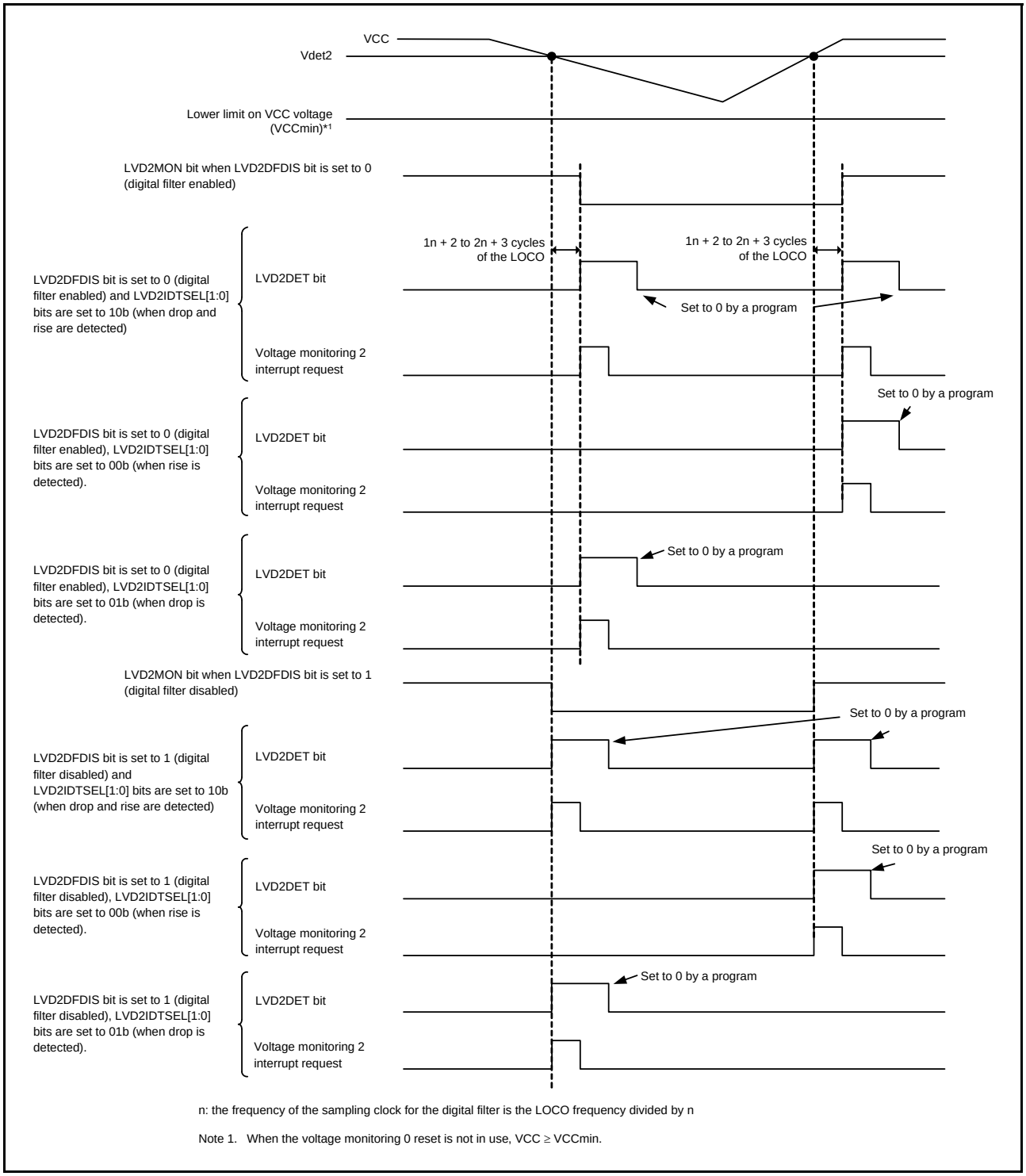
	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
Step	**Voltage Monitoring 2 Interrupt**	**Voltage Monitoring 2 Reset**	**Voltage Monitoring 2 Interrupt**	**Voltage Monitoring 2 Reset**
1*2	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.			
2*2	Set the LVCMPCR.EXVREFINP2 bit to 0 (internal reference voltage). Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (selecting the input voltage on the CMPA2 pin).			
3*1	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).	
4 *1, *2	Clear the LVD2CR0.LVD2RI bit to 0 (to select the voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (to select the voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.	Clear the LVD2CR0.LVD2RI bit to 0 (to select the voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (to select the voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.
5	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
6	—	Set the LVD2CR0.LVD2RIE bit to 1 (enabling voltage monitoring 2 interrupt/reset)	—	Set the LVD2CR0.LVD2RIE bit to 1 (enabling voltage monitoring 2 interrupt/reset)
7*2	Set the LVCMPCR.LVD2E bit to 1 (enabling the circuit for voltage detection 2).			
8*2	Wait for at least td(E-A) or longer.			
9	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by the voltage monitoring 2 circuit).			
10	Wait for at least one cycle of the LOCO.		—	
11	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).		—	
12	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).		— (No waiting is required.)	
13	Clear the LVD2SR.LVD2DET bit to 0.	—	Clear the LVD2SR.LVD2DET bit to 0.	—
14	Set the LVD2CR0.LVD2RIE bit to 1 (enabling voltage monitoring 2 interrupt/reset)	—	Set the LVD2CR0.LVD2RIE bit to 1 (enabling voltage monitoring 2 interrupt/reset)	—

Note 1. Executing steps 3 and 4 at the same time (with a single instruction) creates no problems.
Note 2. Steps 1, 2, 4, 7, and 8 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.LVD2DFDIS and LVD2FSAMP bits or LVD2CR1.LVD2IRQSEL and LVD2IDTSEL bits after monitoring is stopped 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 14.

Table 8.8 Procedures for Stopping Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset

Step	Voltage Monitoring 2 Interrupt	Voltage Monitoring 2 Reset
1	Clear the LVD2CR0.LVD2RIE bit to 0 (disabling voltage monitoring 2 interrupt/reset).	—
2	Clear the LVD2CR0.LVD2CMPE bit to 0 (disabling output of the results of comparison by the voltage monitoring 2 circuit).	
3*1	Clear the LVCMPCR.LVD2E bit to 0 (disable the voltage monitoring 2 circuit).	
4	—	Clear the LVD2CR0.LVD2RIE bit to 0 (disabling voltage monitoring 2 interrupt/reset).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD2E, LVD2CR0.LVD2RIE, and LVD2CR0.LVD2CMPE.	

Note 1. Step 3 is 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.LVD2DFDIS and LVD2FSAMP bits or LVD2CR1.LVD2IRQSEL and LVD2IDTSEL bits after monitoring is stopped 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 5.



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

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

Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

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

(1) Vdet1 passage detection event

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

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

8.7.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring 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 mode. The event signals for the ELC, however, are output as follows:

When the events of passing Vdet1 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 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 passage detection flags.

9. Clock Generation Circuit

9.1 Overview

The RX220 Group 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	Description
Use	• Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, ROM, and RAM. • Generates the peripheral module clocks (PCLKB and PCLKD) to be supplied to peripheral modules. The peripheral module clock used as the operating clock is PCLKD for S12AD and PCLKB for other modules. • Generates the FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the RTC-dedicated sub clock (RTCSCLK) to be supplied to the RTC. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT.
Operating frequency*1	• ICLK: 32 MHz (max) • PCLKB: 32 MHz (max) • PCLKD: 32 MHz (max) • FCLK: 4 MHz to 32 MHz (for programming and erasing the ROM and E2 DataFlash) 32 MHz (max) (for reading from the E2 DataFlash) • CACCLK: Same as frequency of each oscillator • RTCSCLK: 32.768 kHz • IWDTCLK: 125 kHz
Main clock oscillator	• Resonator frequency: 1 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 is detected with the main clock, the system clock source is switched to LOCO and MTU output can be forcedly driven to the high-impedance.
Sub-clock oscillator	• Resonator frequency: 32.768 kHz • Connectable resonator or additional circuit: crystal resonator • Connection pin: XCIN, XCOUT
High-speed on-chip oscillator (HOCO)	• Oscillation frequency: 32 MHz/36.864 MHz/40 MHz/50 MHz • HOCO power supply control
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 125 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 125 kHz

Note 1. The maximum operating frequency in middle-speed operating mode A1. For the maximum operating frequency in the other operating modes, see section 11.2.5, Operating Power Control Register (OPCCR).

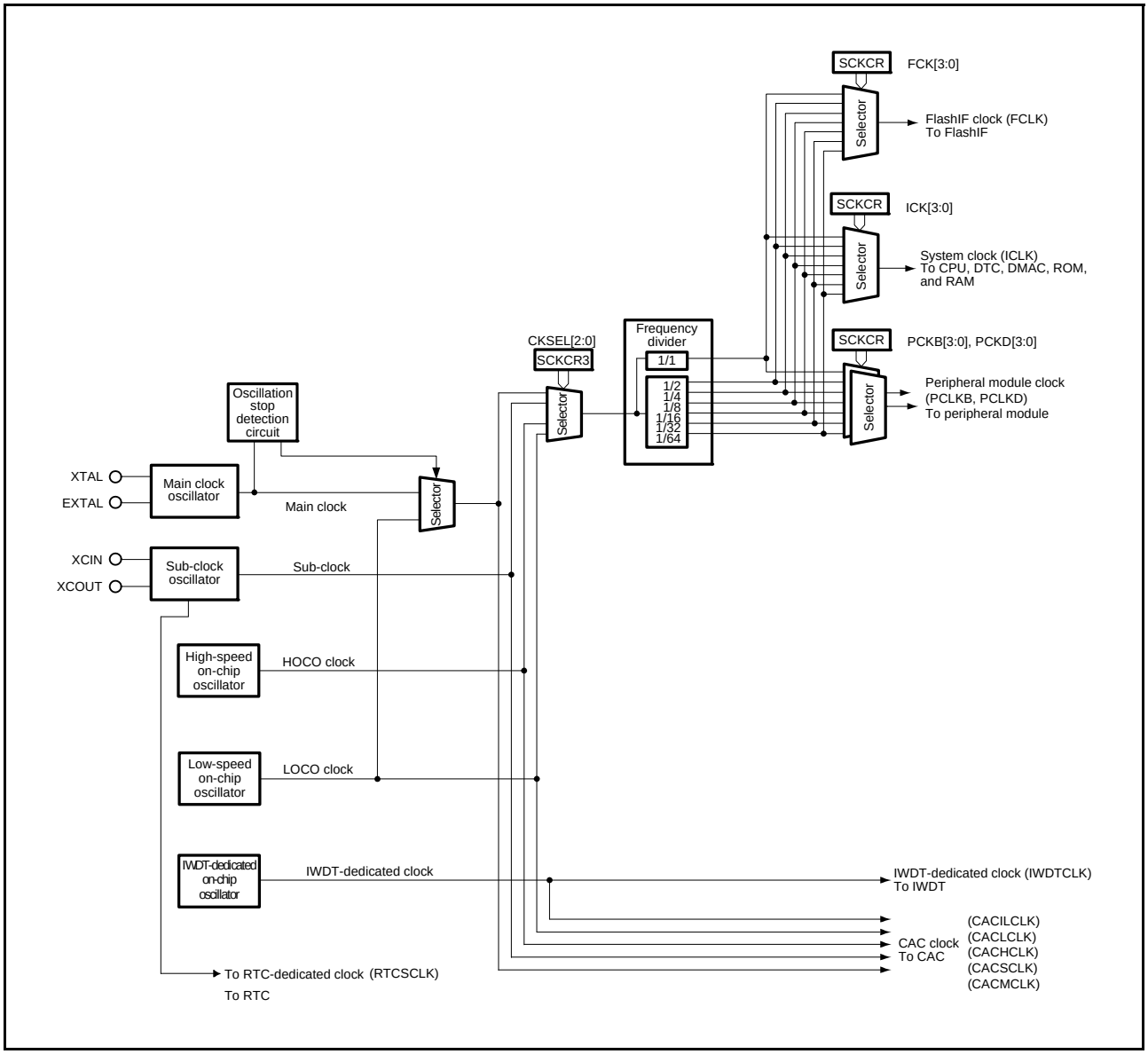


Figure 9.1 Block Diagram of Clock Generation Circuit

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	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator.
XCOUT	Output	

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]				—	—	—	—	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]	Peripheral Module Clock D (PCLKD) Select	b3 b0 0 0 0 0: ×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	These bits should be set to 0001b.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	b11 b8 0 0 0 0: ×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	These bits should be set to 0001b.	R/W
b19 to b16	BCK[3:0]	External Bus Clock (BCLK) Select	b19 b16 0 0 0 0: ×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
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select	b27 b24 0 0 0 0: ×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	b31 b28 0 0 0 0: ×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.
The following frequency relationship need be established between the system clock (ICLK) and external bus clock (BCLK):
ICLK ≥ BCLK
When BCLK is not to be used, specify either one of the frequency division settings for ICLK and PCLKB, whichever is greater, in the SCKCR.BCK[3:0] bits. If this setting causes BCLK to exceed 25 MHz, however, specify the maximum frequency division setting that will never cause BCLK to exceed 25 MHz in the SCKCR.BCK[3:0] bits.

When an instruction for writing to SCKCR, or SCKCR3 is to follow writing to the SCKCR register, do so in accord with the procedure below.

1. Write to the SCKCR register.
2. Confirm that the value has actually been written to the SCKCR register.
3. Proceed to the next step.

SCKCR should not be modified in the following cases:

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

PCKD[3:0] Bits (Peripheral Module Clock D (PCLKD) Select)

These bits select the frequency of the peripheral module clock (PCLKD).
This is the operating clock for the S12AD module.

PCKB[3:0] Bits (Peripheral Module Clock B (PCLKB) Select)

These bits select the frequency of the peripheral module clock (PCLKB).
This is the operating clock for modules other than the S12AD module.

BCK[3:0] Bits (External Bus Clock (BCLK) Select)

These bits select the frequency of the external bus clock (BCLK).

ICK[3:0] Bits (System Clock (ICLK) Select)

These bits select the frequency of the system clock (ICLK).

FCK[3:0] Bits (FlashIF Clock (FCLK) Select)

These bits select the frequency of the FlashIF clock (FCLK).

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 0 1: HOCO 0 1 0: Main clock oscillator 0 1 1: Sub-clock oscillator 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 the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The ROM P/E mode entry i bit in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (i = 0, D)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

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

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKB and PCLKD), and FlashIF clock (FCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, and the sub-clock oscillator.
Transitions to clock sources which are not in operation are prohibited.

9.2.3 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.

Set up this register after setting up the main clock oscillator wait control register described in section 11, Low Power Consumption.

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

When a crystal resonator is connected to supply the main clock, after changing the MOSTP bit so that the main clock oscillator operates, only use the main clock after the main clock oscillation stabilization wait time (crystal; $t_{MAINOSCWT}$) has elapsed.

When an external clock is input to supply the main clock, set the MOSCWTCR.MSTS[4:0] bits to 00000b. After changing the MOSTP bit so that the main clock oscillator operates, only use the main clock after the EXTAL external clock input wait time (t_{EXWT}) 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 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 mode.
- When a transition to software standby mode 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.

Do not set the MOSTP bit to 1 when one of the following conditions is met.

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)

9.2.4 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Sub-clock oscillator is operating. 1: Sub-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.

Set up this register after setting up the sub-clock oscillator wait control register described in section 11, Low Power Consumption.

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) control whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.

When changing the value of the SOSTP bit or RCR3.RTCEN 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).

After the setting of the SOSTP bit or the RCR3.RTCEN bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization wait time ($t_{SUBOSCWT}$) has elapsed.

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

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

9.2.5 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 wait 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.
- When a transition to software standby mode 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 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 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).

9.2.6 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 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 wait for stabilization of the LOCO (tLOCOWT) has elapsed. If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait 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 mode.

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

Address(es): 0008 0036h

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

Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

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

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

Set up this register after setting up the HOCO wait control register 2 described in section 11, Low Power Consumption.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

When changing the value of the HCSTP bit from 1 to 0 (stopped to operating), supply of the clock signal within the LSI circuit starts after the wait time set by the HOCOWTCR2.HSTS2[3:0] bits has elapsed. Only start using the oscillator after this wait time has elapsed. That is, a fixed time for stabilization of oscillation is required after the setting for HOCO 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 HOCO after it has been stopped, allow at least five cycles of the HOCO as an interval over which it is still stopped.
- Ensure that oscillation by the HOCO is stable when making the setting to stop the HOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the HOCO is stable before executing a WAIT instruction to place the chip on software standby mode.
- When a transition to software standby mode is to follow the setting to stop the HOCO, wait for at least two cycles of the HOCO after the setting to stop the HOCO and before executing the WAIT instruction.

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

Writing 0 to the HCSTP bit (making the HOCO operate) is prohibited when the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 2.

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

Address(es): 0008 0037h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 32 MHz 0 1: 36.864 MHz 1 0: 40 MHz 1 1: 50 MHz	R/W
b7 to b2	—	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 HOCOCR2 register is prohibited while the HOCOCR.HCSTP bit is 0 (the setting for the HOCO to operate).

HCFRQ[1:0] Bits (HOCO Frequency Setting)

These bits set the frequency of the HOCO.

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 POE. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE.	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 a given I/O register, wait 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.

After this function is enabled, t_{dr} (see Table 38.39, Oscillation Stop Detection Circuit Characteristics) is required before stable operation starts.

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. To make a transition to 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. This bit can only be set to 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 flag is not cleared to 0 even though the main clock oscillation is restarted. The OSTDF flag 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 the OSTDF flag and reading the OSTDF flag as 0. If the OSTDF flag is cleared to 0 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.
When the main clock oscillator (010b) is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), the OSTDF flag cannot be modified to 0.

[Setting condition]

- The main clock oscillation is stopped with the OSTDCR.OSTDE bit 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.

9.2.11 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV2[1:0]		MODRV[2:0]			—
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	MODRV[2:0]	Main Clock Oscillator Drive Capability Switch	b3 b2 b1 0 0 0: Other than 16-MHz to 20-MHz lead type ceramic resonator 1 1 1: 16-MHz to 20-MHz lead type ceramic resonator Settings other than the above are prohibited.	R/W
b5, b4	MODRV2[1:0]	Main Clock Oscillator Drive Capability Switch 2	b5 b4 0 1: 1 MHz to 8 MHz 1 0: 8.1 MHz to 15.9 MHz 1 1: 16 MHz to 20 MHz Settings other than the above are prohibited.	R/W
b6	MOSEL	Main Clock Oscillator Switch	0: Resonator 1: External oscillator input	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.

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port.

MODRV[2:0] Bits (Main Clock Oscillator Drive Capability Switch)

These bits select the drive capability of the main clock oscillator.

MODRV2[1:0] Bits (Main Clock Oscillator Drive Capability Switch 2)

These bits select the drive capability of the main clock oscillator.

MOSEL Bit (Main Clock Oscillator Switch)

This bit selects the oscillation source of the main clock oscillator.

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

Address(es): 0008 C294h

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

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOPCNT	High-Speed On-Chip Oscillator Power Supply Control	0: Turns the power supply of the HOCO on. 1: Turns the power supply of the HOCO off.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

HOCOPCNT Bit (High-Speed On-Chip Oscillator Power Supply Control)

This bit controls the power supply for the HOCO.

When this bit is set to 0, the power supply of the HOCO is turned on, enabling oscillation.

When this bit is set to 1, the power supply of the HOCO is turned off, reducing power consumption.

When setting the HOCOPCNT bit to 1, set the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) to 1 (HOCO stopped).

After the HOCOPCNT bit is changed from 1 to 0, oscillation stabilization time is required before the HOCOCR.HCSTP bit is set to 0. For details, see section 38, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit in the following cases:

- When the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- When the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or low-speed operating mode 2. The value of the HOCOPCNT bit can be changed only when the setting is for middle-speed operating mode 1A or middle-speed operating mode 1B.

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal from 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.

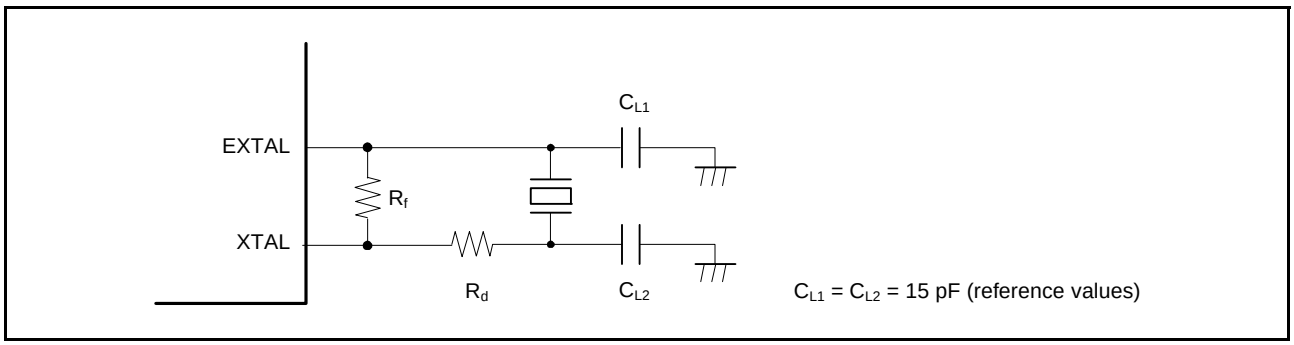


Figure 9.2 Example of Crystal Resonator Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	1	8	16	20
Rd (Ω)	750	0	0	0

Figure 9.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 9.4.

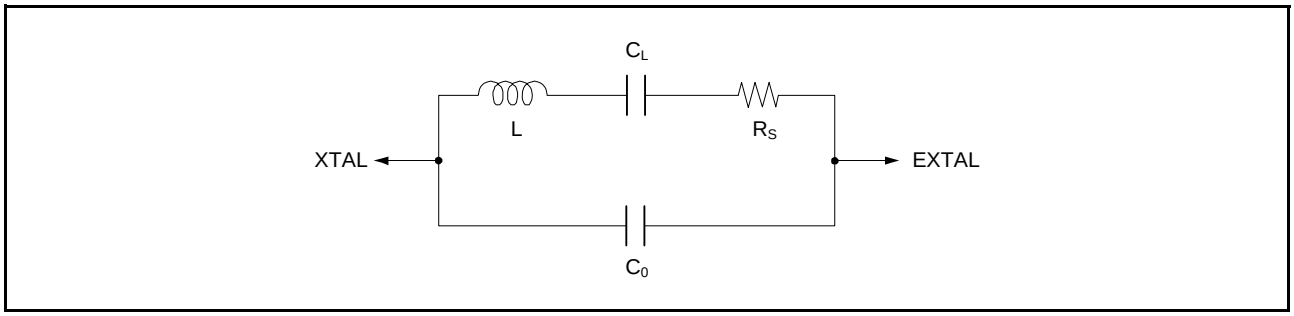


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Resonator Characteristics (Reference Values)

Frequency (MHz)	1	8	16	20
R_S max (Ω)	480	14	16.8	8.1
C_0 max (pF)	3	2.2	3	3.5

9.3.2 External Clock Input

Figure 9.4 illustrates connection of an external clock. Set the MOFCR.MOSEL bit to 1 if operation is to be driven by an external clock. In this case, the XTAL pin will be in the Hi-Z state.

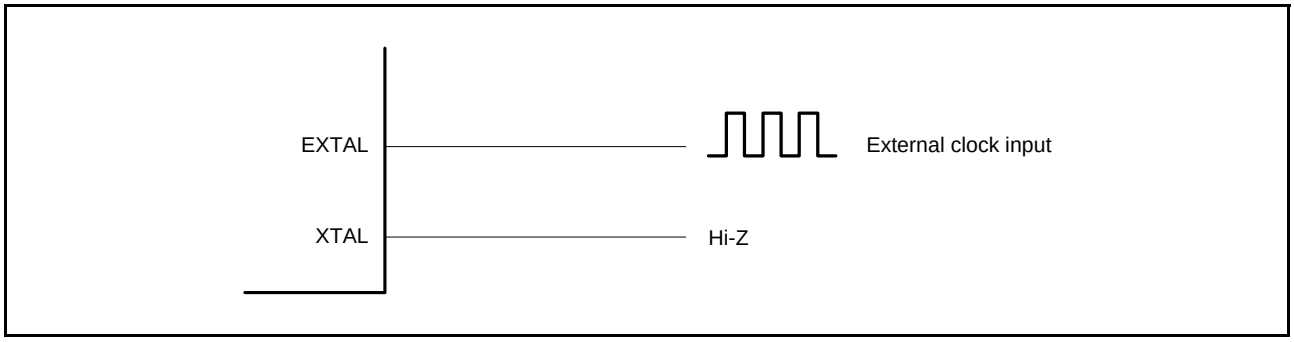


Figure 9.4 Connection Example of External Clock

9.3.3 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (making the main clock oscillator run).

9.4 Sub-Clock Oscillator

The only way of supplying the clock signal from the sub-clock oscillator is connecting a crystal resonator.

9.4.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.5. 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 XCIN and XCOUT 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 sub-clock oscillator described in Table 9.1.

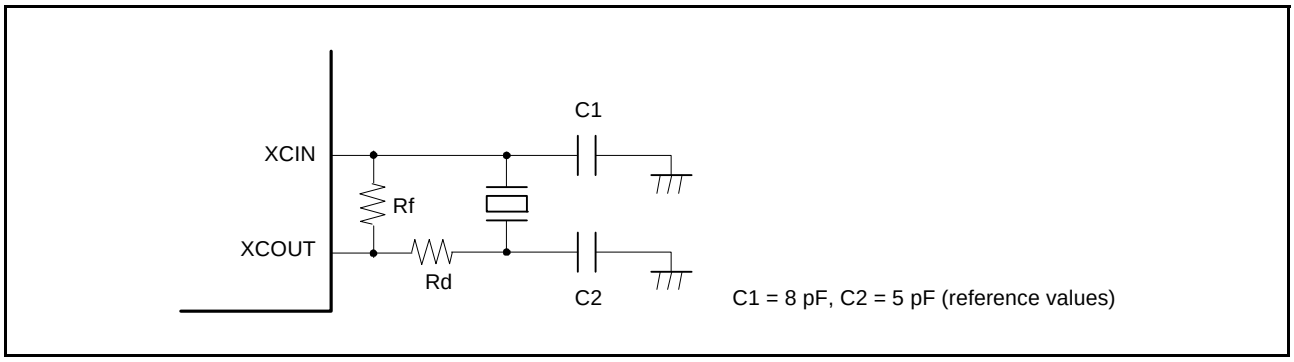


Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator

Figure 9.6 shows an equivalent circuit for the 32.768-kHz crystal resonator. Use a crystal resonator that has the characteristics listed in Table 9.5.

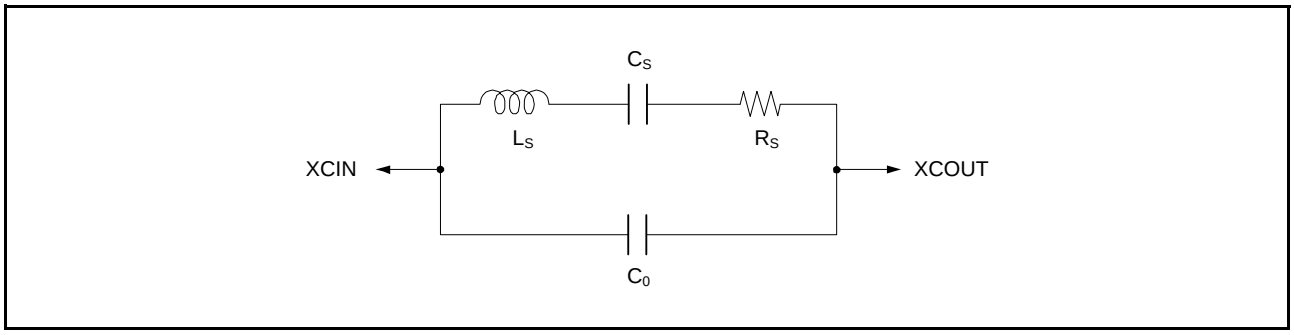


Figure 9.6 Equivalent Circuit for Crystal Resonator

Table 9.5 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768 (Low CL)	32.768 (Standard CL)
R_S max (Ω)	37	35
C_0 max (pF)	0.9	2.0

9.4.2 Handling of Pins when Sub-Clock is not Used

If the sub-clock is not in use, connect the XCIN pin to VSS and leave the XCOUT pin open as shown in Figure 9.7. Set the sub-clock oscillator stop bit (SOSCCR.SOSTP) to 1 (stopping the sub-clock oscillator) and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0 (stopping the sub-clock oscillator). The state of the sub-clock control circuit will be undefined after a cold start. Accordingly, be sure to set these bits after a cold start.

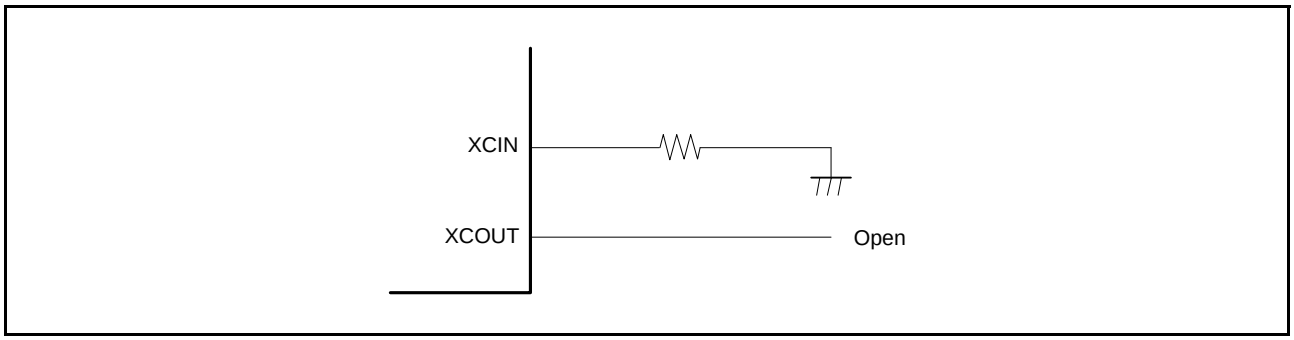


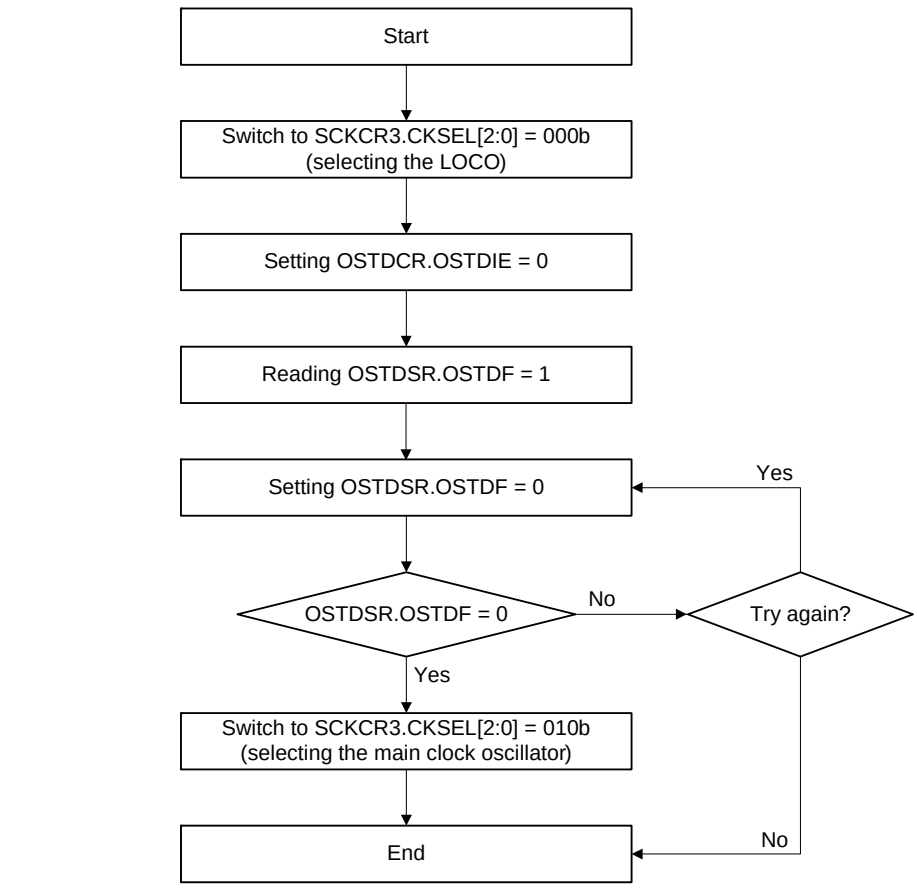
Figure 9.7 Pin Handling when Sub-Clock is not Used

9.5 Oscillation Stop Detection Function

9.5.1 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.
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 21, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 22, Port Output Enable 2 (POE2a).
In the RX220 Group, 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 section 38, Electrical Characteristics).
When an oscillation stop is detected, the main 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 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 is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the low-speed clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is cleared to 0. At this time, if the main 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 again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main 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 after the specified oscillation stabilization 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 stabilization 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.
The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, and CAC main clock (CACMCLK), which are provided as the system clock sources.
The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock select bits (SCKCR.ICK[3:0])



Note: • On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.8 Flow of Recovery from Detection of Oscillator Stop

9.5.2 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 detection). At this time, the main clock oscillator stop is notified to the port output enable 2 (POE). On accepting the notification of the oscillation stop, the POE 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 lease 10 cycles of PCLK 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 a given I/O register, wait 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).

9.6 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock and IWDT-dedicated clock. 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 and PCLKD)
(3) Operating clock of the FlashIF: FlashIF clock (FCLK)
(4) Operating clock for the CAC: CAC clock (CACCLK)
(5) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCLK)
(6) Operating clock for the IWDT: IWDT-dedicated clock (IWDTCLK)

Frequencies of the internal clocks are set by the combination of the division ratios selected by the FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits in SCKCR, and the clock source selected by the CKSEL[2:0] bits in SCKCR3. 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, and the CKSEL[2:0] bits in SCKCR3.

9.6.2 Peripheral Module Clock

The peripheral module clocks (PCLKB and PCLKD) are the operating clocks for use by peripheral modules.
The PCLKB and PCLKD frequencies are specified by the PCKB[3:0] and PCKD[3:0] bits in SCKCR, and the CKSEL[2:0] bits in SCKCR3.
The peripheral module clock used as the operating clock is PCLKD for S12ADand PCLKB for other modules.

9.6.3 FlashIF Clock

The flash-interface clock (FCLK) is used as the operating clock for the flash-memory interfaces. That is, FCLK is used for the ROM, programming and erasure of the E2 DataFlash, and reading from the E2 DataFlash.
The FCLK frequency is specified by the FCK[3:0] bits in SCKCR, and the CKSEL[2:0] bits in SCKCR3.

9.6.4 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.
The CACCLK clocks include CACMCLK which is generated by the main clock oscillator, CACSCLK which is generated by the sub-clock oscillator, CACHCLK which is generated by the high-speed on-chip oscillator, CACLCLK which is generated by the low-speed on-chip oscillator, and CACILCLK which is generated by the IWDT-dedicated on-chip oscillator.

9.6.5 RTC-Dedicated Clock

The RTC-dedicated clock (RTCSCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator.

9.6.6 IWDT-Dedicated Clock

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

9.7 Usage Notes

9.7.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKB and PCLKD), and FlashIF clock (FCLK) supplied to each module change according to the settings of SCKCR. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKB and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) 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.

(3) The RX220 Group does not have an external bus, but it is necessary to set the external bus clock (BCLK) select bits. Specify either one of the frequency division settings for ICLK and PCLKB, whichever is greater, in the SCKCR.BCK bit. If this setting causes BCLK to exceed 25 MHz, however, specify the maximum frequency division setting that will never cause BCLK to exceed 25 MHz in the SCKCR.BCK bit.

9.7.2 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.

9.7.3 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 9.9 to prevent electromagnetic induction from interfering with correct oscillation.

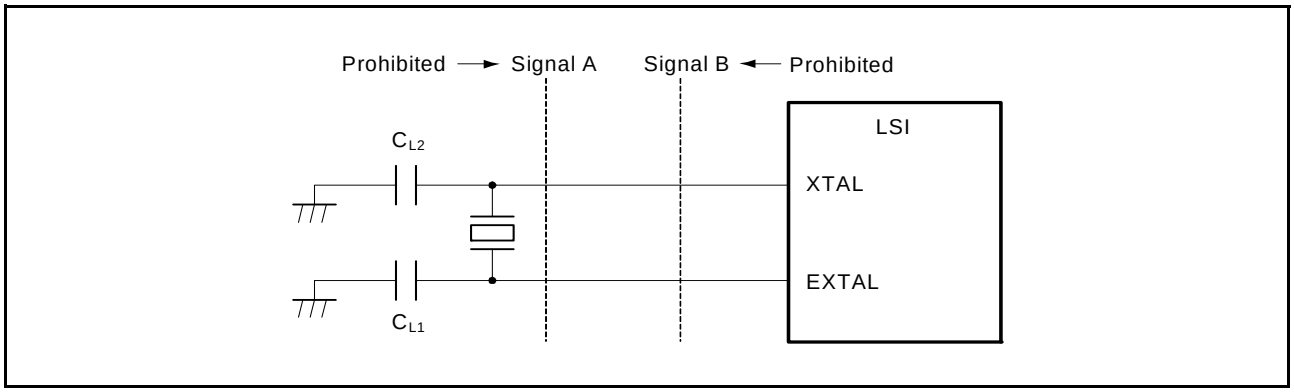


Figure 9.9 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

9.7.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1. However, with the system using the main clock, the EXTAL and XTAL pins should not be used as the general ports.
If the main clock is used, P36 and P37 should not be set to be output.

9.7.5 Notes on Sub-Clock

The sub-clock can be used as the system clock, as the count source for the realtime clock, or as both. Take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

- With regard to making the sub-clock oscillator run or stop, setting either the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) or the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) will make the oscillator run.
- To use the sub-clock as the system clock and as the count source for the realtime clock simultaneously, after the sub-clock starts oscillating and the oscillation stabilization wait time has elapsed, the SOSCWTCR.SSTS[4:0] bits must be set to 00000b. To set these bits, perform the initial setting procedure below. After setting the SSTS[4:0] bits, perform the clock setting procedure shown in section 25.3.2.

Initial setting procedure

(1) Wait for the oscillation stabilization wait time of the sub-clock*1 to elapse.
(2) Set the SOSCCR.SOSTP bit to 1.
(3) Read the SOSCCR.SOSTP bit and confirm that it is 1.
(4) Set the RCR3.RTCEN bit to 0.
(5) Read the RCR3.RTCEN bit and confirm that it is 0.
(6) Wait for at least five cycles of the sub-clock to elapse.
(7) Set the RCR3.RTCDV[2:0] bits.
If the RCR3.RTCDV[2:0] bits are set in this step, they do not need to be reset in section 25.3.2, Clock and count mode Setting Procedure.
(8) Set the SOSCWTCR.SSTS[4:0] bits to specify the wait time necessary for sub-clock oscillation.
(9) Set the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).
(10) Wait for the oscillation stabilization wait time of the sub-clock*1 to elapse.
(11) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 0, then set the RCR3.RTCEN bit to 1 (sub-clock oscillator is operating).
(12) Read the RCR3.RTCEN bit and confirm that it has been rewritten to 1, then set the SOSCCR.SOSTP bit to 1.
(13) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten to 1, then wait for at least five cycles of the sub-clock to elapse.
(14) Set the SOSCWTCR.SSTS[4:0] bits to 00000b.
(15) Set the SOSCCR.SOSTP bit to 0.
(16) Wait for at least two cycles of the sub-clock to elapse.
(17) Read the SOSCCR.SOSTP bit and confirm that it has been rewritten.

Note 1. For details on the oscillation stabilization wait time of the sub-clock, see section 11.2.8, Sub-Clock Oscillator Wait Control Register (SOSCWTCR).

- Regardless of the RCR3.RTCEN bit setting, wait until the oscillator stabilization wait time elapses before rewriting the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).
- Since the sub-clock control circuit is in an unstable state after a cold start, it must be initialized regardless of whether or not the sub-clock is in use. The sub-clock is initialized by setting the SOSCCR.SOSTP bit to 1 and the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped). See section 25.2.19, RTC Control Register 3 (RCR3), for instructions to initialize the RCR3.RTCEN bit.
  Although the sub-clock oscillator pins are not available in 48-pin LQFP packages, initialize the sub-clock control circuit in the same way.
- The RCR3.RTCDV[2:0] bits must also be set when operating the sub-clock oscillator. Set these bits while the sub-clock oscillator is stopped. Do not rewrite these bits while the sub-clock oscillator is operating.
- When successively rewriting the SOSCCR.SOSTP bit followed by the RCR3.RTCEN bit or vice versa, confirm that the first bit rewrite was completed successfully before rewriting the second bit.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

The clock frequency accuracy measurement circuit (CAC) monitors the clock frequency based on a reference signal input to the LSI externally or another clock source, and generates interrupts when the setting range is exceeded.

10.1 Overview

Table 10.1 shows the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 Specifications of CAC

Item	Description
Clock frequency measurement	The frequency of the following clocks can be measured. • Clock output from main clock oscillator (main clock) • Clock output from sub-clock oscillator (sub-clock) • Clock output from high-speed on-chip oscillator (HOCO clock) • Clock output from low-speed on-chip oscillator (LOCO clock) • Clock output from IWDT-dedicated on-chip oscillator (IWDTCLK clock)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Power consumption reduction function	Module stop state can be set.

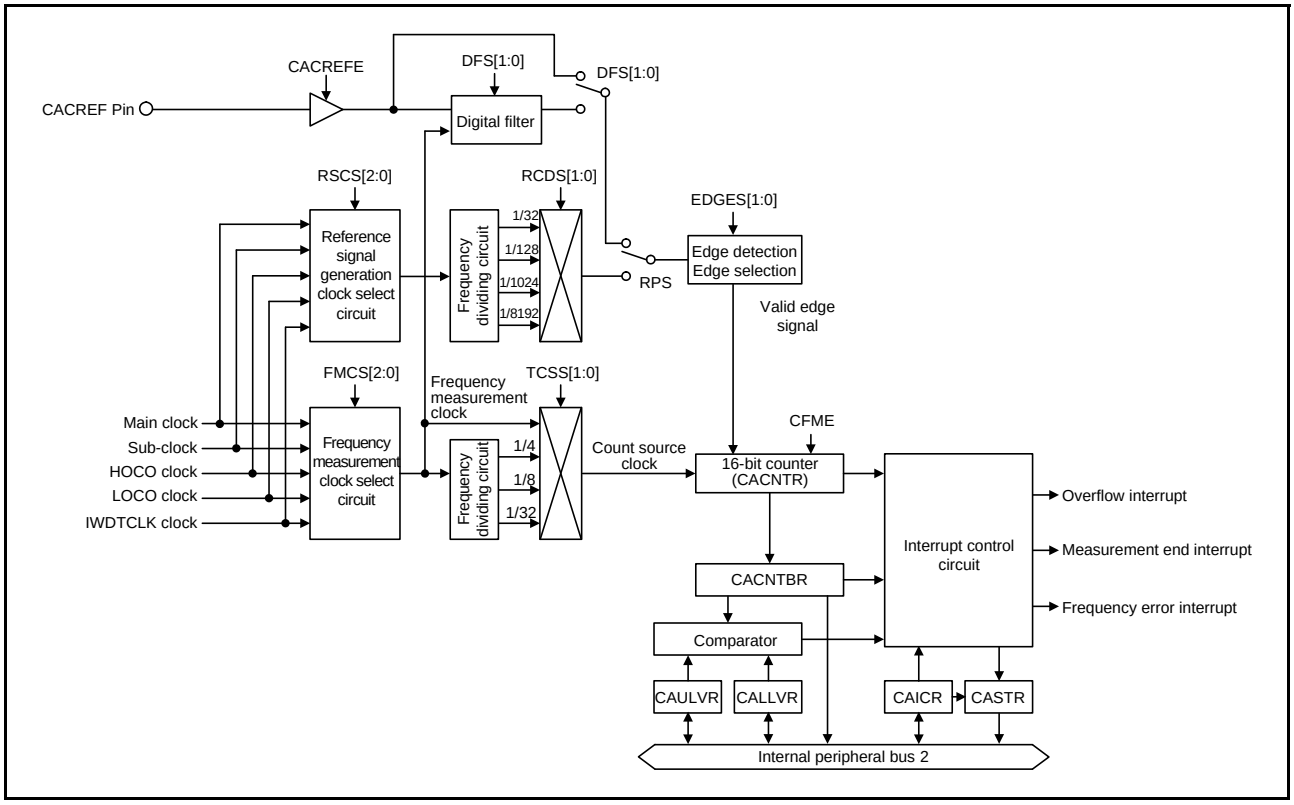


Figure 10.1 Block Diagram of CAC

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Clock frequency accuracy measurement circuit 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

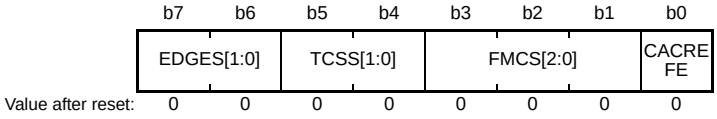
CFME Bit (Clock Frequency Measurement Enable)

When this bit is 1, clock frequency measurement is enabled.

After a new value is written to this bit, reflecting the value in the internal circuitry takes a certain amount of time. Note that writing to this bit is ineffective if a previously written value has not yet been reflected in the internal circuitry. Read the bit to confirm whether a value written has actually been reflected.

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]	Frequency Measurement Clock Select	b3 b1 0 0 0: Output clock of main clock oscillator 0 0 1: Output clock of sub-clock oscillator 0 1 0: Output clock of high-speed on-chip oscillator 0 1 1: Output clock of low-speed on-chip oscillator 1 0 0: Output clock of IWDT-dedicated on-chip oscillator 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. CACR1 should be set when the CFME bit in CACR0 is 0.

CACREFE Bit (CACREF Pin Input Enable)

When this bit is 1, the CACREF pin input is enabled.

FMCS[2:0] Bits (Frequency Measurement Clock Select)

These bits select the 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: Internally generated signal	R/W
b3 to b1	RSCS[2:0]	Reference Signal Generation Clock Select	b3 b1 0 0 0: Output clock of main clock oscillator 0 0 1: Output clock of sub-clock oscillator 0 1 0: Output clock of high-speed on-chip oscillator 0 1 1: Output clock of low-speed on-chip oscillator 1 0 0: Output clock of IWDT-dedicated on-chip oscillator Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Reference Signal Generation Clock Frequency Division Ratio 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 Selection	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. CACR2 should be set when the CFME bit in CACR0 is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internally generated signal as the reference signal.

RSCS[2:0] Bits (Reference Signal Generation Clock Select)

These bits select the clock source for generating the reference signal.

RCDS[1:0] Bits (Reference Signal Generation Clock Frequency Division Ratio Select)

These bits select the frequency division ratio of the reference signal generation clock.

DFS[1:0] Bits (Digital Filter Selection)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Control 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 Enable	0: Frequency error interrupt is disabled. 1: Frequency error interrupt is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Enable	0: Measurement end interrupt is disabled. 1: Measurement end interrupt is enabled.	R/W
b2	OVFIE	Overflow Interrupt Enable	0: Overflow interrupt is disabled. 1: Overflow interrupt 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 FERRF bit is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the MENDF bit is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the OVFF bit 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 Enable)

When this bit is 1, a frequency error interrupt is enabled.

MENDIE Bit (Measurement End Interrupt Enable)

When this bit is 1, a measurement end interrupt is enabled.

OVFIE Bit (Overflow Interrupt Enable)

When this bit is 1, an overflow interrupt is enabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the FERRF bit.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the MENDF bit.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the OVFF bit.

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	[Setting condition] • The clock frequency is outside of the setting range. [Clearing condition] • 1 is written to the FERRFCL bit.	R
b1	MENDF	Measurement End Flag	[Setting condition] • Measurement has finished. [Clearing condition] • 1 is written to the MENDFCL bit.	R
b2	OVFF	Overflow Flag	[Setting condition] • The counter has overflowed. [Clearing condition] • 1 is written to the OVFFCL bit.	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 bit is set to 1 when the clock frequency is outside of the setting range.

MENDF Flag (Measurement End Flag)

This bit is set to 1 when measurement has finished.

OVFF Flag (Overflow Flag)

This bit is set to 1 when the counter has overflowed.

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 stores the upper-limit value of the frequency.
This register should be set when the CFME bit in CACR0 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 stores the lower-limit value of the frequency.
This register should be set when the CFME bit in CACR0 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 Based on CACREF Pin Input

Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit based on the CACREF pin input.
The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

(1) Writing 1 to the CFME bit in CACR0 while the RPS bit in CACR2 is 0 and the CACREFE bit in CACR1 is 1 enables clock-frequency measurement based on the CACREF pin input.
(2) After 1 is written to the CFME bit, the timer starts up-counting when the valid edge selected by the EDGES[1:0] bits in CACR1 is input from the CACREF pin.
(3) When the next valid edge is input, the counter value is retained 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 will occur.
(4) When the next valid edge is input, the counter value is retained 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 will occur. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt will occur.
(5) When the next valid edge is input, the counter value is retained 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 will occur. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt will occur.
(6) While the CFME bit in CACR0 is 1, the counter value is retained 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.

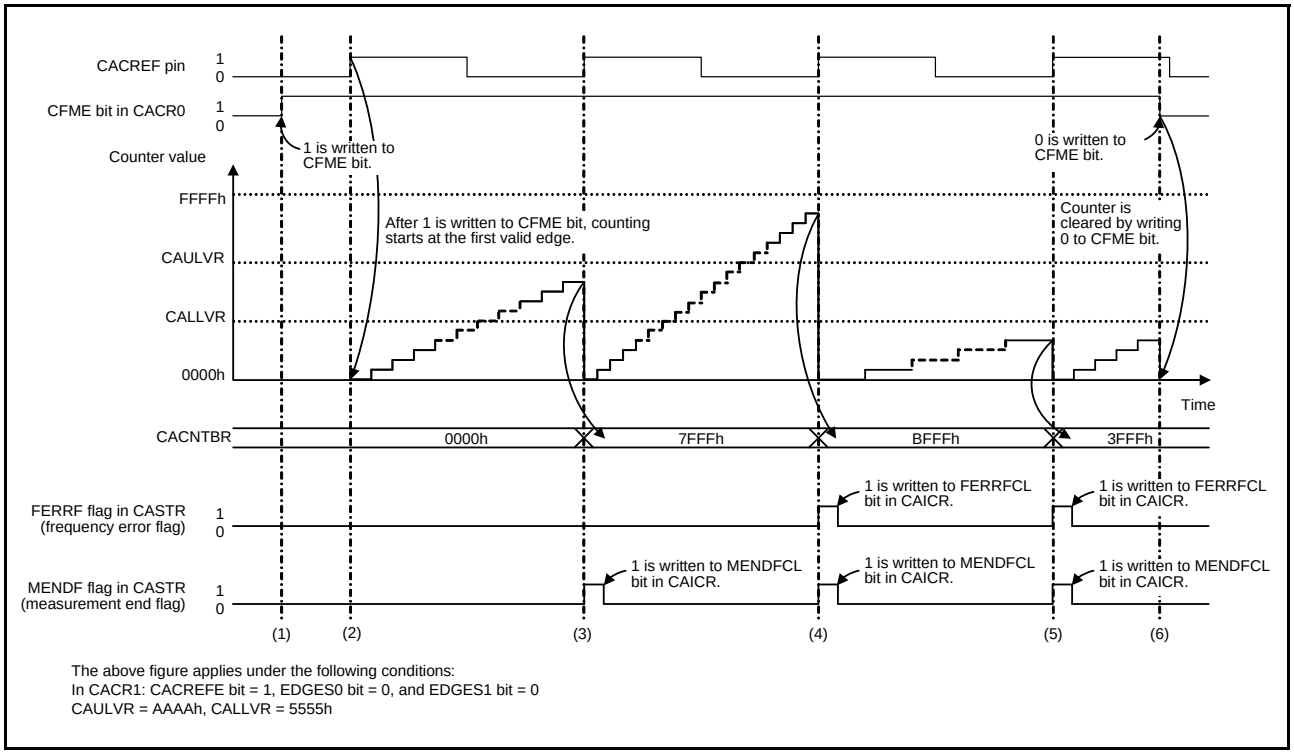


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit Based on CACREF Pin Input

10.3.2 Measuring Clock Frequency Based on Another Clock Source

Figure 10.3 shows an operating example of the clock frequency accuracy measurement circuit based on another clock source.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

(1) When 1 is written to the CFME bit in CACR0 with the RPS bit in CACR2 set to 1, clock frequency measurement based on another clock source is enabled.

(2) After 1 is written to the CFME bit, the timer starts up-counting when the valid edge selected by the EDGES[1:0] bits in CACR1 is input based on the clock source selected by the RSCS[2:0] bits in CACR2.

(3) When the next valid edge is input, the counter value is retained 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 will occur.

(4) When the next valid edge is input, the counter value is retained 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 will occur. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt will occur.

(5) When the next valid edge is input, the counter value is retained 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 will occur. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt will occur.

(6) While the CFME bit in CACR0 is 1, the counter value is retained 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.

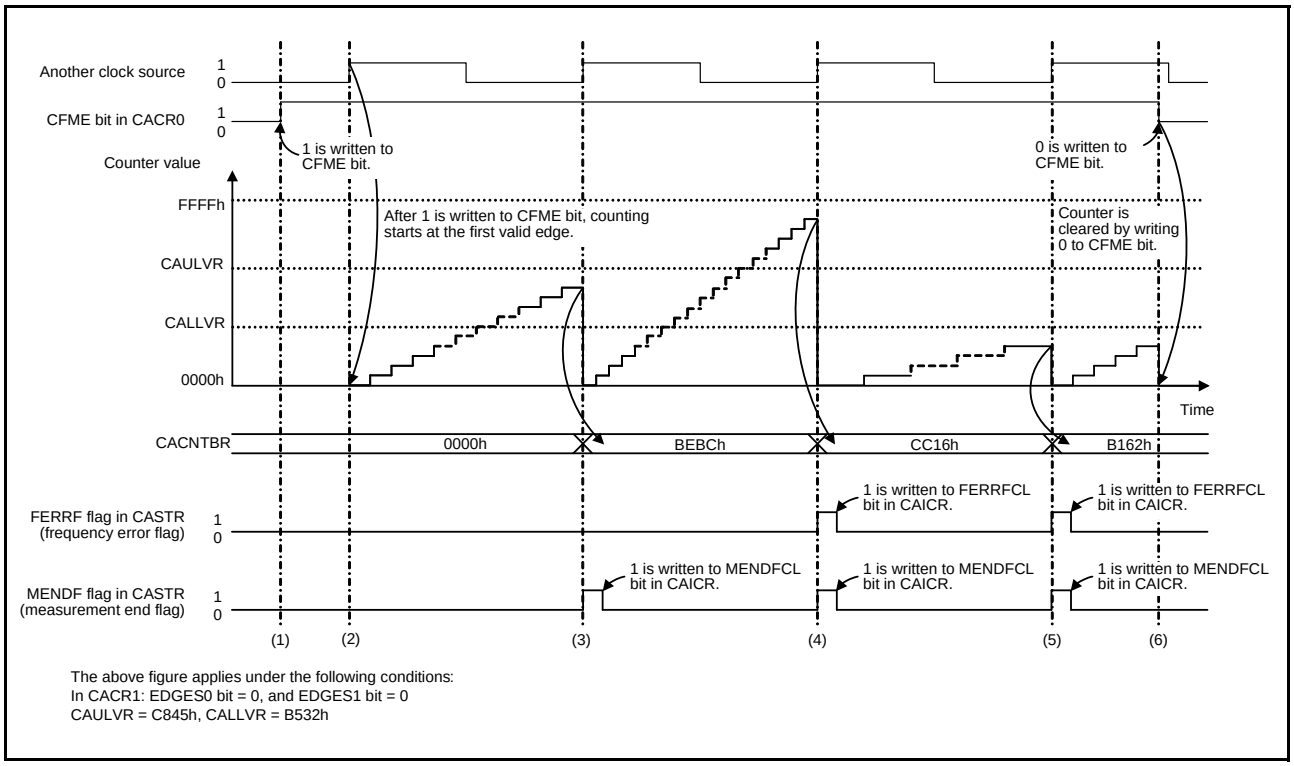


Figure 10.3 Operating Example of Clock Frequency Accuracy Measurement Circuit Based on Another Clock Source

10.3.3 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 times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three times.
Enabling and disabling of the digital filter and its sampling clock are selectable.
The counter value held 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 clock frequency accuracy measurement circuit issues three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag is set to 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 CFME bit in CACR0.
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 canceling the module stop state. For details, see section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

The RX220 Group has several functions for reducing power consumption, including switching clock signals to reduce power consumption, 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 DMAC, DTC, and RAM are stopped.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Description
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Functions can be stopped independently for each peripheral module.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • All-module clock stop mode • Software standby mode
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 control mode according to the operating frequency and operating voltage. • Four operating power control modes Middle-speed operating mode 1A Middle-speed operating mode 1B Low-speed operating mode 1 Low-speed operating mode 2

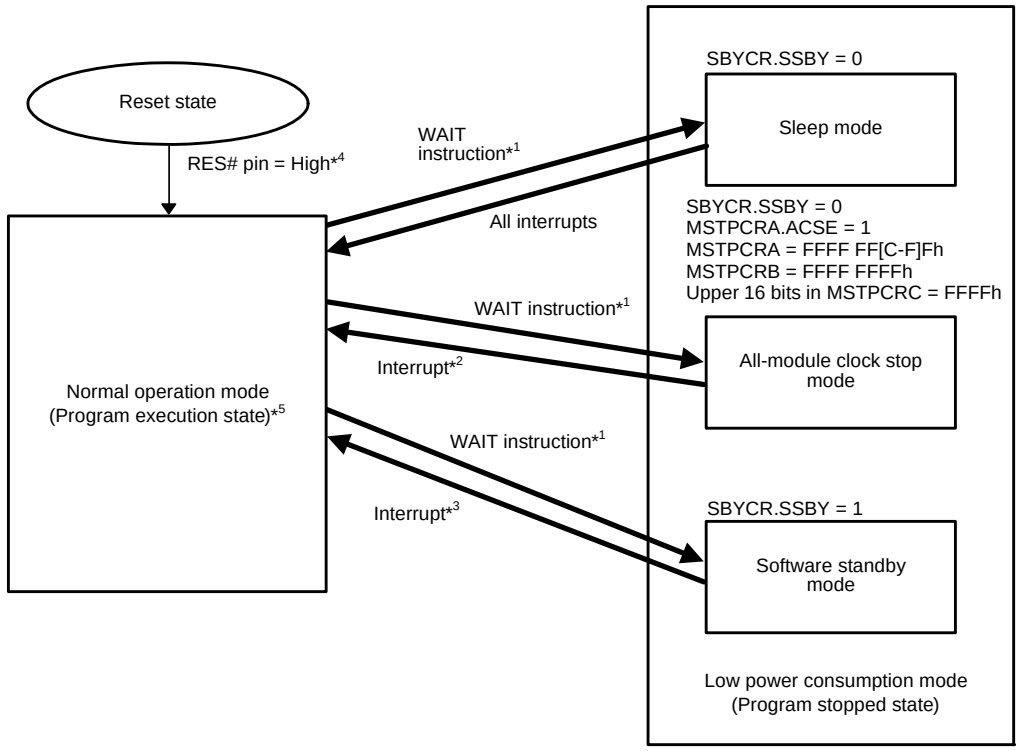
Note 1. For details, see section 9, Clock Generation Circuit.

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
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction
Canceling method other than reset	Interrupt	Interrupt*1	Interrupt*2
State after cancellation*3	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
Main clock oscillator	Operating possible	Operating possible	Stopped
Sub-clock oscillator	Operating possible	Operating possible	Operating possible*4
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*5	Operating possible*5	Operating possible*5
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0000 3FFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*5	Operating possible*5	Operating possible*5
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*6	Stopped (Retained)
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible*7
Power-on reset circuit	Operating	Operating	Operating*7
Peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)
I/O ports	Operating	Retained*8	Retained

“Operating possible” means that operating or stopped can be controlled by the control register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.
“Stopped (Undefined)” means that internal register values are undefined and power is not supplied to the internal circuit.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, 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 IRQ7) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. This does not include release initiated by a reset on the RES# pin, power-on reset, voltage monitoring reset, or independent watchdog timer reset. The transition is to the reset state when release is initiated by one of these reset sources.
Note 4. Operation or stopping is selected by the sub-clock oscillator control bit (RTCEN) in RTC control register 3 (RCR3).
Note 5. Operation or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operation or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSTPR).
Note 6. Stopping or operation is controlled by the module-stop setting bits (MSTPA4 and MSTPA5, respectively) in module stop control register A (MSTPCRA) for 8-bit timers 0 and 1 (unit 0) and 2 and 3 (unit 1).
Note 7. In the case of a transition to software standby mode when the setting of the software cut bit in the flash HOCO software standby control register (FHSSBYCR.SOFTCUT2) is 1, the voltage monitoring circuits are stopped and the low power consumption function of the power-on reset circuit is enabled.
Note 8. While the 8-bit timer and RTC are operated, the related pins continue operation.



Note 1. If an interrupt that becomes a canceling source is accepted while making a transition to the program stop state after the WAIT instruction has been executed, the transition to the program stop state is canceled and the interrupt exception handling is executed

Note 2. "Interrupts" here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, voltage monitoring 1, voltage monitoring 2, and oscillator-stopped detection interrupts). However, an 8-bit timer interrupt is only effective when 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 IRQ7) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts).

Note 4. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode

Note 5. Makes a transition from sleep mode, all-module clock stop mode, or software standby mode to normal operating mode by an interrupt. In the case of recovery from sleep mode, the clock source for subsequent use is selectable. For details, see section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). For software standby mode, the clock source after returning is the same as that before returning.

When a RES# pin reset, power-on reset, voltage monitoring 0 reset, or independent watchdog timer reset is generated in any state, a transition to the reset state is made.

Figure 11.1 Mode Transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

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

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

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

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	—	—	—	—	—	—	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	—	—	—	—	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	Multi-Function Timer Pulse Unit 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
b13 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: 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 Module Stop	Target module: S12AD 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b23 to b18	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b24	MSTPA24	Module Stop A24	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b26, b25	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPA27	Module Stop A27	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: 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, be sure that 1 has been written to this bit.	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	ACSE	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

Note: • 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 MSTPA 4 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
	—	MSTPB30	—	—	—	MSTPB26	MSTPB25	—	MSTPB23	—	MSTPB21	—	—	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MSTPB9	—	—	MSTPB6	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCIf Module Stop	Target module: SCIf (SCI12) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	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 to 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
b16 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b20 to	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: 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	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: 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
b29 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: 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.

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
	—	—	—	—	—	MSTPC26	—	—	—	—	—	MSTPC20	MSTPC19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0000 3FFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop*2	Target module: CAC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b20	MSTPC20	IrDA Module Stop	Target module: IRDA 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b25 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	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
b31 to b27	—	Reserved	These bits are 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 MSTPC0 bit should not be set to 1 during access to the corresponding RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after 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	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 1 0: Middle-speed operating mode 1A 0 1 1: Middle-speed operating mode 1B 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 transition status flag (OPCMTSF) is 1 (operating power control mode switching is in progress)
- When the ROM P/E mode entry i bit in the flash P/E mode entry register (FENTRYR.FENTRYi) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (i = 0, 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 ROM 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 an operational power control mode, refer to section 11.5, Function for Lower Operating Power Consumption.

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 Voltage Range	Operating Frequency Range					Power Consumption
			During Flash Memory Read				During Flash Memory Programming/ Erasure	
			ICLK	FCLK	PCLKD	PCLKB	FCLK	
Middle-speed operating mode 1A	010b	3.6 V to 5.5 V	32 MHz max	32 MHz max	32 MHz max	32 MHz max	4 MHz to 32 MHz	Large
		2.7 V to 3.6 V					4 MHz to 32 MHz	
		1.62 V to 2.7 V	8 MHz max	8 MHz max	8 MHz max	8 MHz max	—	
Middle-speed operating mode 1B	011b	3.6 V to 5.5 V	32 MHz max	32 MHz max	32 MHz max	32 MHz max	—	
		2.7 V to 3.6 V					4 MHz to 32 MHz	
		1.62 V to 2.7 V	8 MHz max	8 MHz max	8 MHz max	8 MHz max	4 MHz to 8 MHz	
Low-speed operating mode 1	110b	3.6 V to 5.5 V	8 MHz max	8 MHz max	8 MHz max	8 MHz max	—	
		2.7 V to 3.6 V					—	
		1.8 V to 2.7 V	4 MHz max	4 MHz max	4 MHz max	4 MHz max	—	
		1.62 V to 1.8 V	2 MHz max	2 MHz max	2 MHz max	2 MHz max	—	
Low-speed operating mode 2	111b	3.6 V to 5.5 V	32.768 kHz max	32.768 kHz max	32.768 kHz max	32.768 kHz max	—	
		2.7 V to 3.6 V					—	
		1.8 V to 2.7 V					—	
		1.62 V to 1.8 V					—	Small

Each operating power control mode is described below.

- Middle-Speed Operating Mode 1A

This mode can be operated in a wide voltage range.
The maximum operating frequency during FLASH read is 32 MHz for ICLK, FCLK, PCLKD, and PCLKB. The operating voltage range is 1.62 to 5.5 V during FLASH read. However, for ICLK, FCLK, PCLKD, and PCLKB, the maximum operating frequency during FLASH read is 8 MHz when the operating voltage is 1.62 V or larger and smaller than 2.7 V.
During FLASH programming/erasure, the operating frequency range is 4 to 32 MHz and the operating voltage range is 2.7 to 5.5 V.
After a reset is canceled, operation is started in this mode.

- Middle-Speed Operating Mode 1B

This mode is basically the same as the middle-speed operating mode 1A but provides further reduced power consumption during low-voltage FLASH programming/erasure.
The operating frequency range and operating voltage range during FLASH read are the same as those in middle-speed operating mode 1A.
However, during FLASH programming/erasure, the operating frequency range is 4 to 32 MHz and the operating voltage range is 1.62 to 3.6 V.
The maximum operating frequency during FLASH programming/erasure is 8 MHz when the operating voltage is 1.62 V or larger and smaller than 2.7 V.
Figure 11.2 shows the relationship between the operating voltages and operating frequencies in middle-speed operating modes 1A and 1B.

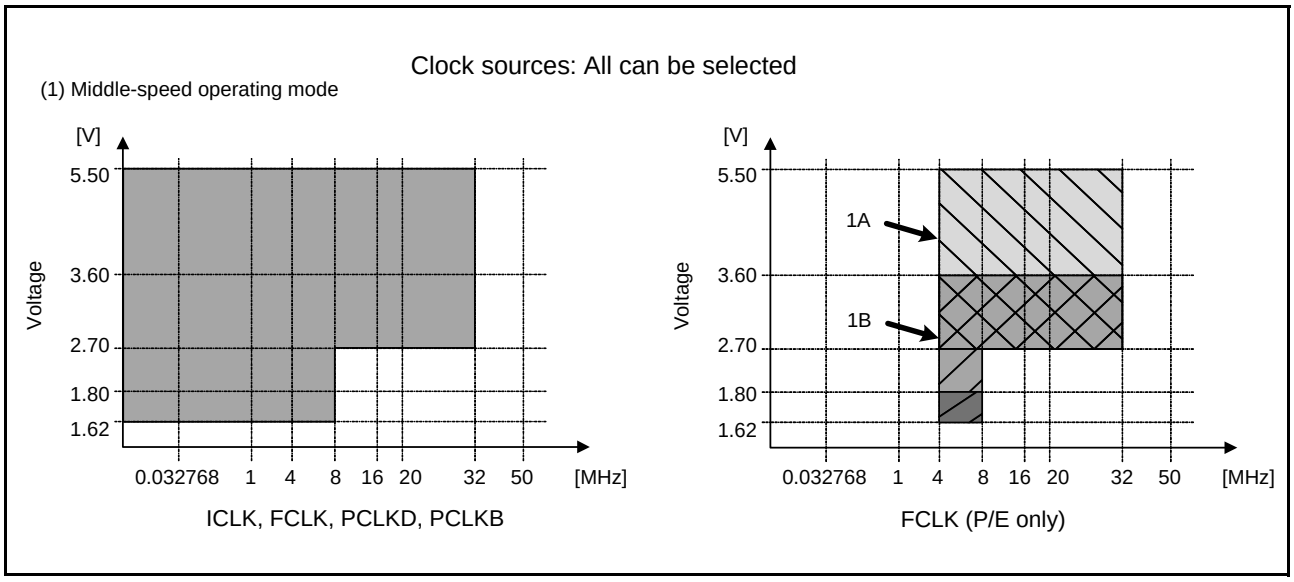


Figure 11.2 Relationship between the Operating Voltages and Operating Frequencies in Middle-Speed Operating Modes 1A and 1B

- Low-Speed Operating Mode 1

As compared to middle-speed operating mode 1A/1B, this mode reduces power consumption for low-speed operation. During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKD, and PCLKB is 8 MHz. The operating voltage is in the range of 1.62 to 5.5 V. The maximum operating frequency during FLASH read is 4 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.7 V. Furthermore, voltage below the range from 1.62 V to 1.8 V limits the highest frequency for the ICLK, FCLK, PCLKD, PCLKB, and BCLK signals when flash memory is to be read to 2 MHz.

In low-speed operating mode 1, P/E operation of FLASH is disabled.

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

Figure 11.3 shows the relationship between the operating voltages and operating frequencies in low-speed operating mode 1.

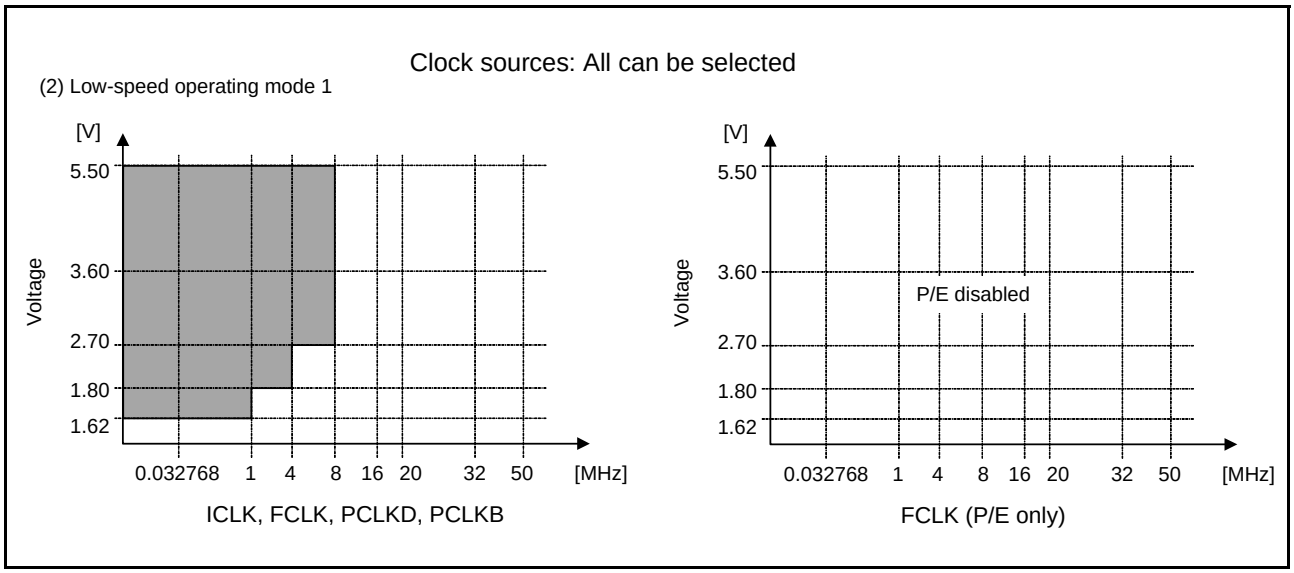


Figure 11.3 Relationship between the Operating Voltages and Operating Frequencies in Low-Speed Operating Mode 1

• Low-Speed Operating Mode 2

As compared to low-speed operating mode 1, this mode reduces power consumption for low-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKD, and PCLKB is 32.768 kHz. The operating voltage is in the range of 1.62 to 5.5 V.
The following restrictions apply when low-speed operating mode 2 is selected:

- P/E operations for flash memory are prohibited.
- Using the HOCO is prohibited.
- Using the oscillation stop detection function of the main clock oscillator 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).

Note: • The OPCM[2:0] bits cannot be set to 111b (low-speed operating mode 2) when the HOCOCR.HCSTP bit is 0 (HOCO operated).

Figure 11.4 shows the relationship between the operating voltages and operating frequencies in low-speed operating mode 2.

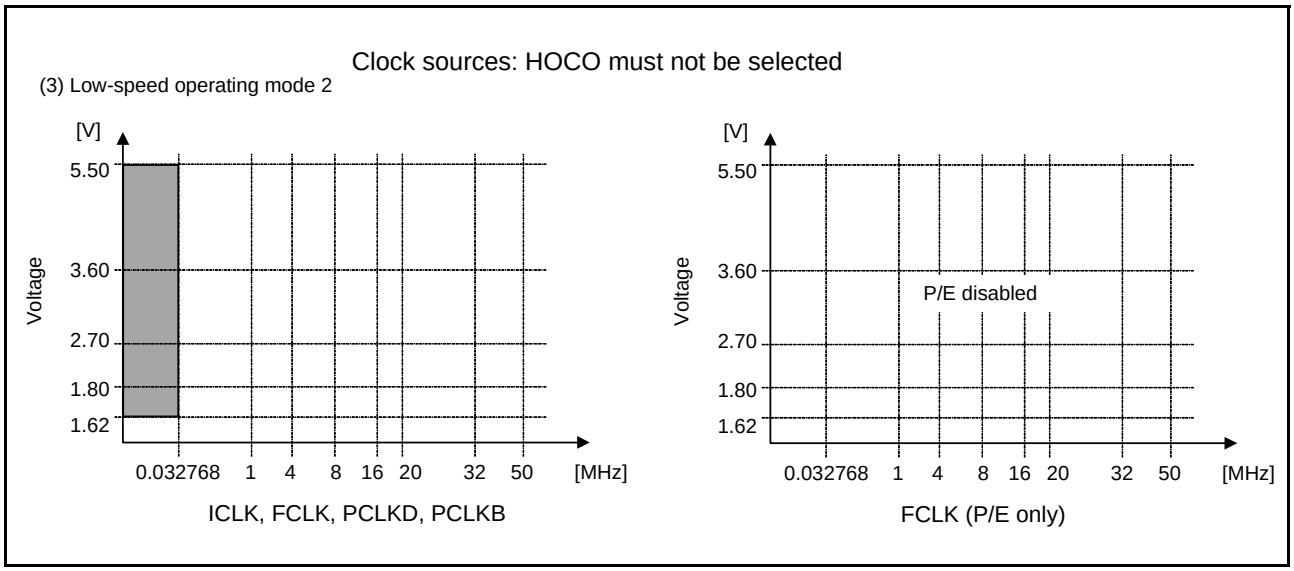


Figure 11.4 Relationship between the Operating Voltages and Operating Frequencies in Low-Speed Operating Mode 2

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 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching 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) and HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. The value of RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]).
When the setting of register RSTCKCR is for the HOCO to be used in recovery from sleep mode, the power supply for the HOCO is not automatically switched on. If the HOCO to be used in recovery from sleep mode, the power supply for the HOCO must be on when the transition to sleep mode takes place.
When returning from sleep mode while the RSTCKEN bit is 1, the OPCCR.OPCM[2:0] bits are automatically switched to middle-speed operating mode 1A (010b) according to the SCKCR register and RSTCKSEL[2:0] bit settings. When returning from sleep mode while the RSTCKEN bit is 0, the MCU returns to the operating mode it was in before transitioning to sleep mode.
When the RSTCKSEL[2:0] bits are set to 001b (HOCO is selected) and the sleep mode return clock source switching function is enabled, the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKD[3:0], and PCKB[3:0] bits should be set to divided-by-2 or more before a transition is made to sleep mode.

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.
Setting RSTCKSEL to 001b (HOCO is selected) is prohibited when one of the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKD[3:0], and PCKB[3:0] bits is set to 0000b (1/1 frequency division).

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 or sub-clock is selected as a clock for a transition to sleep mode. To make a transition to sleep mode with HOCO, or main clock selected as the clock source, the RSTCKEN bit should not be set to 1.
When the CPU is restored from the sleep mode with this bit set to “enable,” the OPCCR.OPCM[2:0] bits are automatically rewritten to turn on middle-speed operating mode 1A.

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	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time Select	b4 b0 0 0 0 0 0: Wait time = 2 cycles 0 0 0 0 1: Wait time = 4 cycles 0 0 0 1 0: Wait time = 8 cycles 0 0 0 1 1: Wait time = 16 cycles 0 0 1 0 0: Wait time = 32 cycles 0 0 1 0 1: Wait time = 256 cycles 0 0 1 1 0: Wait time = 512 cycles 0 0 1 1 1: Wait time = 1024 cycles 0 1 0 0 0: Wait time = 2048 cycles 0 1 0 0 1: Wait time = 4096 cycles 0 1 0 1 0: Wait time = 16384 cycles 0 1 0 1 1: Wait time = 32768 cycles 0 1 1 0 0: Wait time = 65536 cycles 0 1 1 0 1: Wait time = 131072 cycles 0 1 1 1 0: Wait time = 262144 cycles 0 1 1 1 1: Wait 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 oscillation stabilization wait time of the main clock oscillator.
Transmission of the main clock signal to the LSI internally starts after the number of cycles of the main clock set in the MOSCWTCR register have been counted.
Set the MSTS[4:0] bits so that the wait time is at least as long as the main clock oscillator stabilization time (tMAINOSC). For example, if the frequency of the oscillator in use is 10 MHz (so that the period is 100 ns) and the MSTS[4:0] bits are set to 01101b, the wait time will be 100 ns × 131072 cycles, which is approximately 13.11 ms.
The wait time is not required when the main clock is externally input.
MOSCWTCR can only be rewritten when the MOSCCR.MOSTP bit is 1; do not rewrite MOSCWTCR with other settings.

11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

Address(es): 0008 00A3h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	SSTS[4:0]	Sub-Clock Oscillator Wait Time Select	b4 b0 0 0 0 0 0: Wait time = 2 s + 2 cycles 0 0 0 0 1: Wait time = 2 s + 4 cycles 0 0 0 1 0: Wait time = 2 s + 8 cycles 0 0 0 1 1: Wait time = 2 s + 16 cycles 0 0 1 0 0: Wait time = 2 s + 32 cycles 0 0 1 0 1: Wait time = 2 s + 64 cycles 0 0 1 1 0: Wait time = 2 s + 512 cycles 0 0 1 1 1: Wait time = 2 s + 1024 cycles 0 1 0 0 0: Wait time = 2 s + 2048 cycles 0 1 0 0 1: Wait time = 2 s + 4096 cycles 0 1 0 1 0: Wait time = 2 s + 16384 cycles 0 1 0 1 1: Wait time = 2 s + 32768 cycles 0 1 1 0 0: Wait time = 2 s + 65536 cycles 0 1 1 0 1: Wait time = 2 s + 131072 cycles 0 1 1 1 0: Wait time = 2 s + 262144 cycles 0 1 1 1 1: Wait time = 2 s + 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.

SOSCWTCR is used to select the oscillation stabilization wait time of the sub-clock oscillator.
The sub-clock is provided to the LSI internally after the number of sub-clock cycles specified with SOSCWTCR has been counted.
Set the SSTS[4:0] bits so that the wait time is at least as long as the sub-clock oscillator stabilization time (tSUBOSC). For example, if the frequency of the oscillator in use is 32.768 kHz (so that the period is 30.5 μs) and the SSTS[4:0] bits are set to 01101b, the wait time will be 30.5 μs × 32768 cycles, which is approximately 2 s + 1 s = 3 s.
Writing to alter the setting in register SOSCWTCR is only possible when the SOSTP bit in SOSCCR is 1 (sub-clock oscillator is stopped). Do not attempt to change the setting if this is not the case.

11.2.9 HOCO Wait Control Register 2 (HOCOWTCR2)

Address(es): 0008 00A9h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	HSTS2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	HSTS2[4:0]	HOCO Wait Time Select 2	b4 b0 0 0 0 0 0: Wait time = 3072 cycles 0 0 0 0 1: Wait time = 5120 cycles 0 0 0 1 0: Wait time = 7168 cycles 0 0 0 1 1: Wait time = 9216 cycles 0 0 1 0 0: Wait time = 11264 cycles 0 0 1 0 1: Wait time = 13312 cycles 0 0 1 1 0: Wait time = 15360 cycles 0 0 1 1 1: Wait time = 17408 cycles 0 1 0 0 0: Wait time = 19456 cycles 0 1 0 0 1: Wait time = 21504 cycles 0 1 0 1 0: Wait time = 23552 cycles 0 1 0 1 1: Wait time = 25600 cycles 0 1 1 0 0: Wait time = 27648 cycles 0 1 1 0 1: Wait time = 29696 cycles 0 1 1 1 0: Wait time = 31744 cycles 0 1 1 1 1: Wait time = 33792 cycles 1 0 0 0 0: Wait time = 40 cycles 1 0 0 0 1: Wait time = 72 cycles 1 0 0 1 0: Wait time = 104 cycles 1 0 0 1 1: Wait time = 136 cycles 1 0 1 0 0: Wait time = 180 cycles These bits must be set to 10100b if the HOCO clock frequency is set to either of 32 MHz, 36.864 MHz, and 40MHz.*1 1 0 1 0 1: Wait time = 200 cycles These bits must be set to 10101b if the HOCO clock frequency is set to 50 MHz.*1 1 0 1 1 0: Wait time = 232 cycles 1 0 1 1 1: Wait time = 264 cycles 1 1 0 0 0: Wait time = 296 cycles 1 1 0 0 1: Wait time = 328 cycles 1 1 0 1 0: Wait time = 360 cycles 1 1 0 1 1: Wait time = 392 cycles 1 1 1 0 0: Wait time = 424 cycles 1 1 1 0 1: Wait time = 456 cycles 1 1 1 1 0: Wait time = 488 cycles 1 1 1 1 1: Wait time = 520 cycles	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.

Note 1. When this value is set, the HOCO oscillation stabilization time 2 (tHOCO2) is established and a clock whose HOCO oscillation frequency (fHOCO) accuracy is described in the electrical characteristics section is present immediately after the supply of that clock is started.
The supply of this clock is possible even when a setting that provides a smaller cycle count than this setting is specified. In such case, however, the HOCO frequency accuracy that is described in the electrical characteristics section is not guaranteed at the beginning of the supply of the clock because the HOCO oscillation stabilization time 2 (tHOCO2) cannot be established. In either case, the HOCO frequency accuracy described in the electrical characteristics section is established in the tHOCO2 time after the beginning of the clock supply.

HOCOWTCR2 is used to select the oscillation stabilization wait time of the HOCO.

The supply of the HOCO clock to the internal LSI is started after the number of HOCO clocks equivalent to the number of cycles specified in this register are counted.

HOCOWTCR2 can only be rewritten when the HOCOCR.HCSTP bit is 1 (HOCO stopped); do not rewrite HOCOWTCR2 with other settings.

11.2.10 Flash HOCO Software Standby Control Register (FHSSBYCR)

Address: 0008 C28Fh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SOFTCUT[2:0]	Software Cut 0	b2 b0 0 0 0: Power is supplied to HOCO in software standby mode. The voltage detection circuit (LVD) is active and the low power consumption function by the power-on reset circuit (POR) is disabled. 0 1 x: Power is not supplied to HOCO in software standby mode. The voltage detection circuit (LVD) is active and the low power consumption function by the power-on reset circuit (POR) is disabled. 1 0 0: Power is supplied to HOCO in software standby mode. The voltage detection circuit (LVD) is stopped and the low power consumption function by the power-on reset circuit (POR) is enabled. 1 1 x: Power is not supplied to HOCO in software standby mode. The voltage detection circuit (LVD) is stopped and the low power consumption function by the power-on reset circuit (POR) is enabled. Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

To use the voltage detection circuit (LVD) in software standby mode, set the SOFTCUT[2] bit to 0. To stop the voltage detection circuit (LVD) for reducing power consumption and enable the low power consumption function of the power-on reset circuit, set the SOFTCUT[2] bit to 1.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[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], and PCKD[3:0] bits.
The flash memory interface operates on the operating clock specified by the FCK[3:0] bits.
For details, see section 9, Clock Generation Circuit.

11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to C, i = 0 to 31) 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 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 Control Mode

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

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

Example: From middle-speed operating mode 1A to low-speed operating mode 1

(High-speed operation in the operating power control mode used before mode-switching)
↓
Set to switch from the HOCO 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 control mode)

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

Example: From low-speed operating mode 2 to middle-speed operating mode 1A

(Low-speed operation in the operating power 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 HOCO clock (clock source and frequency division ratio)
↓
(High-speed operation in the switched operating power control mode)

The method described below can be used to shorten the time taken by transitions between modes.

- In cases where the transition is from a mode with higher to a mode with lower power consumption, the transition takes the shortest time when the system clock is set to the value for the highest available frequency after the transition.
- In cases where the transition is from a mode with lower to a mode with higher power consumption, the transition also takes the shortest time when the system clock is set to the value for the highest available frequency before the transition.

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.
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 destination to be used for recovery from sleep mode to the CPU.
(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, 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
  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 stabilization wait time, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR) and section 11.2.9, HOCO Wait Control Register 2 (HOCOWTCR2).

If a transition is made to sleep mode with the SBYCR.SSBY bit being 1 (oscillation stop detection function enabled), the sleep mode return clock source switching function is disabled.

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*5, IWDT, RTC, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode*2.

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*3 of the CPU to 0.
(2) Set the interrupt destination to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*4 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*3 of the CPU.
(4) Set the IERm.IENj bit*4 for the interrupt to be used for recovery from all-module clock stop 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*3 of the CPU to 1).

Note 1. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.
Note 2. Transitions to all-module clock stop mode are not to be made in some states of DTC or DMAC operations. Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC so that the DTC and DMAC are not activated.
Note 3. For details, see section 2, CPU.
Note 4. For details, see section 14, Interrupt Controller (ICUb).
Note 5. 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 IRQ7), a peripheral interrupt (8-bit timer*1, RTC alarm, RTC periodic, IWDT*2, voltage monitoring 1, voltage monitoring 2, or oscillator-stopped detection interrupt), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset.

However, note that in cases where a maskable interrupt has been masked by the CPU (priority level*3 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*4 of the CPU) or a maskable interrupt has been set up as a trigger to 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. For details, see section 14, Interrupt Controller (ICUb).

Note 4. 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, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, 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. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Further reduction of power consumption is possible by using the FHSSBYCR register.

In software standby mode, when 000b is set to the FHSSBYCR.SOFTCUT[2:0] bits, internal power supply to the high-speed on-chip oscillator and the power-on reset circuit continues. Since power supply start-up stabilization time is not required at resumption, resumption time from software standby mode is quicker compared to other modes.
When 010b is set to the SOFTCUT[2:0] bits, power supply to the high-speed on-chip oscillator stops, and low power consumption function of the internal power supply is enabled, so the current consumption is reduced.
When 100b is set to the SOFTCUT[2:0] bits, the voltage detection circuit (LVD) stops and the low power consumption function of the power-on reset circuit is enabled, so the current consumption is reduced. At this time, the voltage detection characteristics of the power-on reset circuit is changed. For details, refer to section 38, Electrical Characteristics.
When 110b is set to the SOFTCUT[2:0] bits, power supply to the high-speed on-chip oscillator stops and the low power consumption function of the internal power supply is enabled. Furthermore, the voltage detection circuit (LVD) stops and the low power consumption function of the power-on reset circuit is enabled, so the current consumption is significantly reduced. At this time, the voltage detection characteristics of the power-on reset circuit is changed. For details, refer to section 38, Electrical Characteristics.
When the high-speed on-chip oscillator is not used, the power supply can be turned off with HOCOPCR.HOCOPCNT settings, so the current consumption is reduced even more. For details, refer to section 9, Clock Generation Circuit.

Clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.
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 destination to be used for recovery from software standby mode to the CPU.
(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).

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 IRQ7), peripheral interrupts (the RTC alarm, RTC periodic, IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When an interrupt initiates release from software standby, the oscillators which were 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 among the NMI, IRQ0 to IRQ7, RTC alarm, RTC periodic, IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the time set by the MOSCWTCR.MSTS[4:0], SOSCWTCR.SSTS[4:0], or HOCOWTCR2.HSTS2[4:0] bits has elapsed, the chip is released from software standby mode and starts interrupt exception processing.
- Release due to a reset on the RES# pin
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the 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 11.5 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).

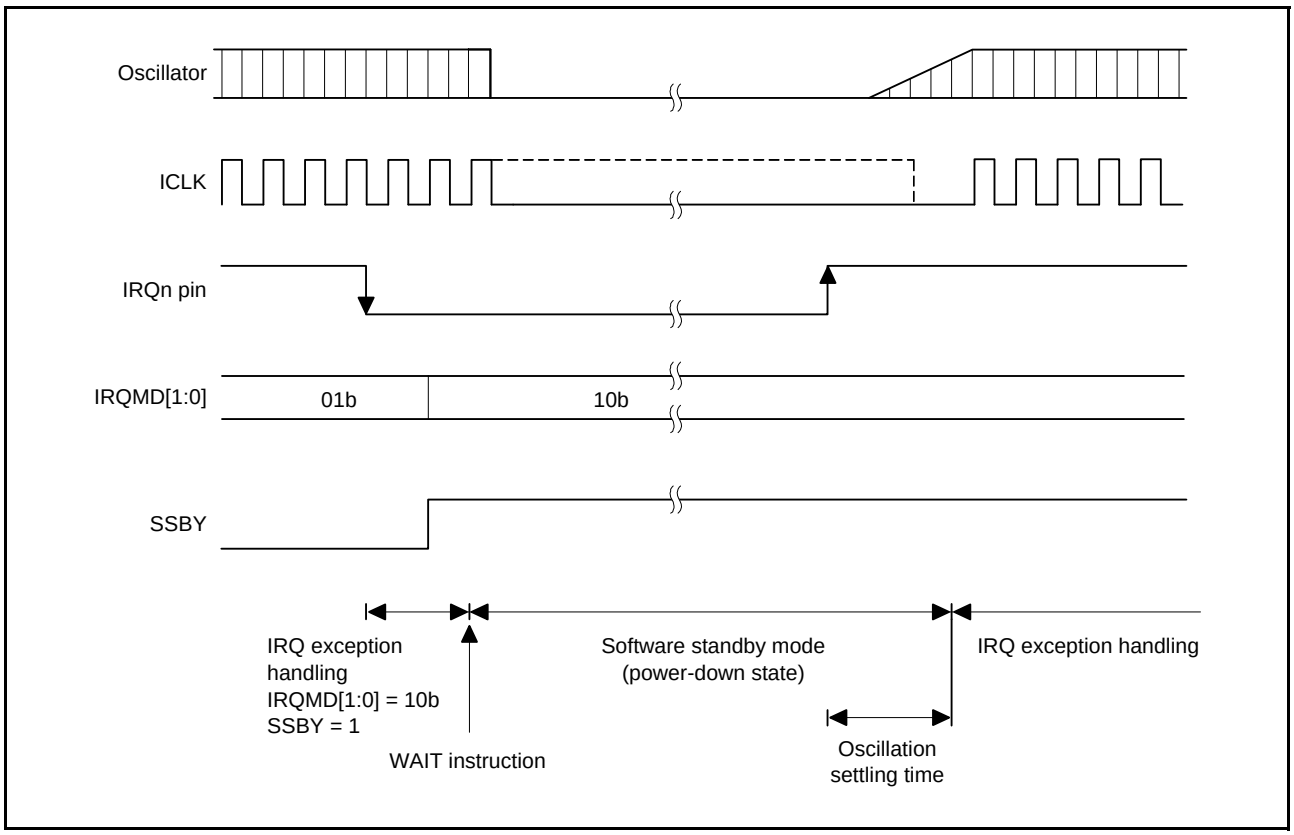


Figure 11.5 Example of Software Standby Mode Application

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in 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 16, DMA Controller (DMACA) and section 17, 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 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction may be executed before the register setting modification is reflected, causing unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last write to the register has completed.

11.7.6 Rewrite the Register by DMAC and DTC 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.

11.7.7 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 canceling source, change the ICLK so as to be faster than the PCLKB before all-module clock stop mode is entered.

11.7.8 Point for Caution when Using the Sub-Clock as the Source of the System Clock

If the sub-clock is in use as the source of the system clock, make sure that the RTC or the low-speed clock oscillator is operating (by setting the RCR3.RTCEN = 1 or the LOCOCR.LCSTP = 0, respectively) for a transition to software standby mode.

12. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 12.1 lists the association between the PRCR bits and the registers to be protected.

Table 12.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR3, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, OSTDCR, OSTDSR, HOCOCR2
PRC1	• Registers related to the operating modes: SYSCR1 • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR, RSTCKCR, MOSCWTCR, SOSCWTCR, FHSSBYCR, HOCOWTCR2 • Registers related to clock generation circuit: MOFCR, HOCOPCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

12.1 Register Descriptions

12.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, 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 enable and disable 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 six 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.

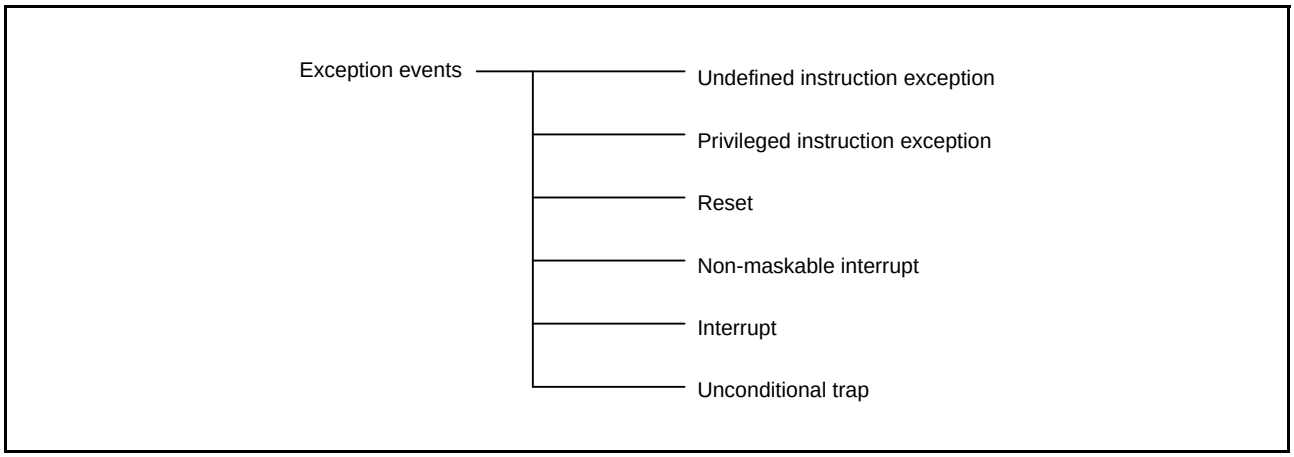


Figure 13.1 Types of Exception Events

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 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.

13.1.4 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.

13.1.5 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.

13.1.6 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.

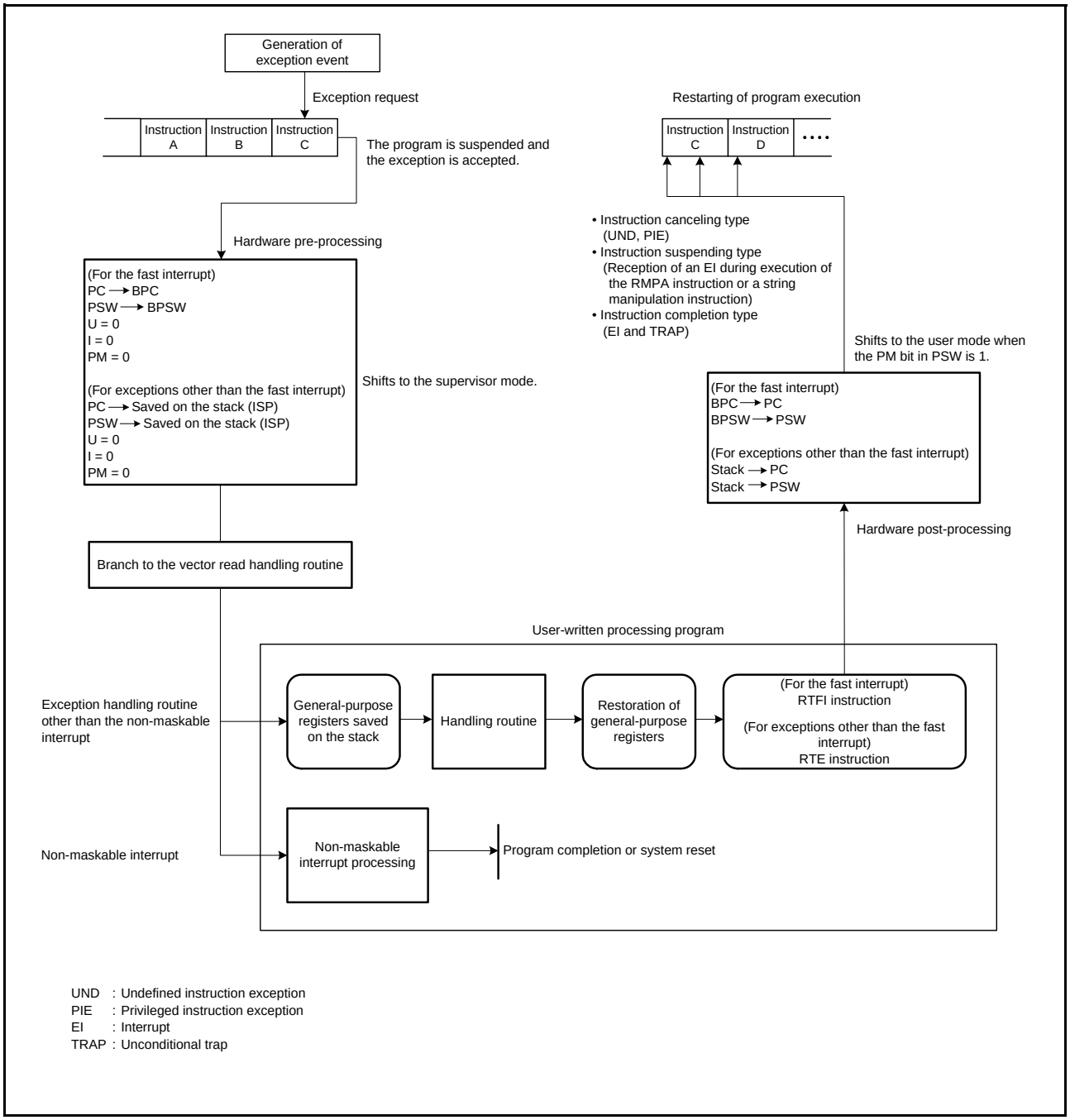


Figure 13.2 Outline of Exception Handling Procedure

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
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
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

(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 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.

13.5.4 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.

13.5.5 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.

13.5.6 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
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	Undefined instruction exception Privileged instruction exception
Low	5	Unconditional trap

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 IRQ7 • Number of sources: 8 • 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 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
	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 IRQ7 interrupts, TMR interrupts, or RTC alarm/periodic interrupts. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ7 interrupts, or RTC alarm/periodic interrupts.

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

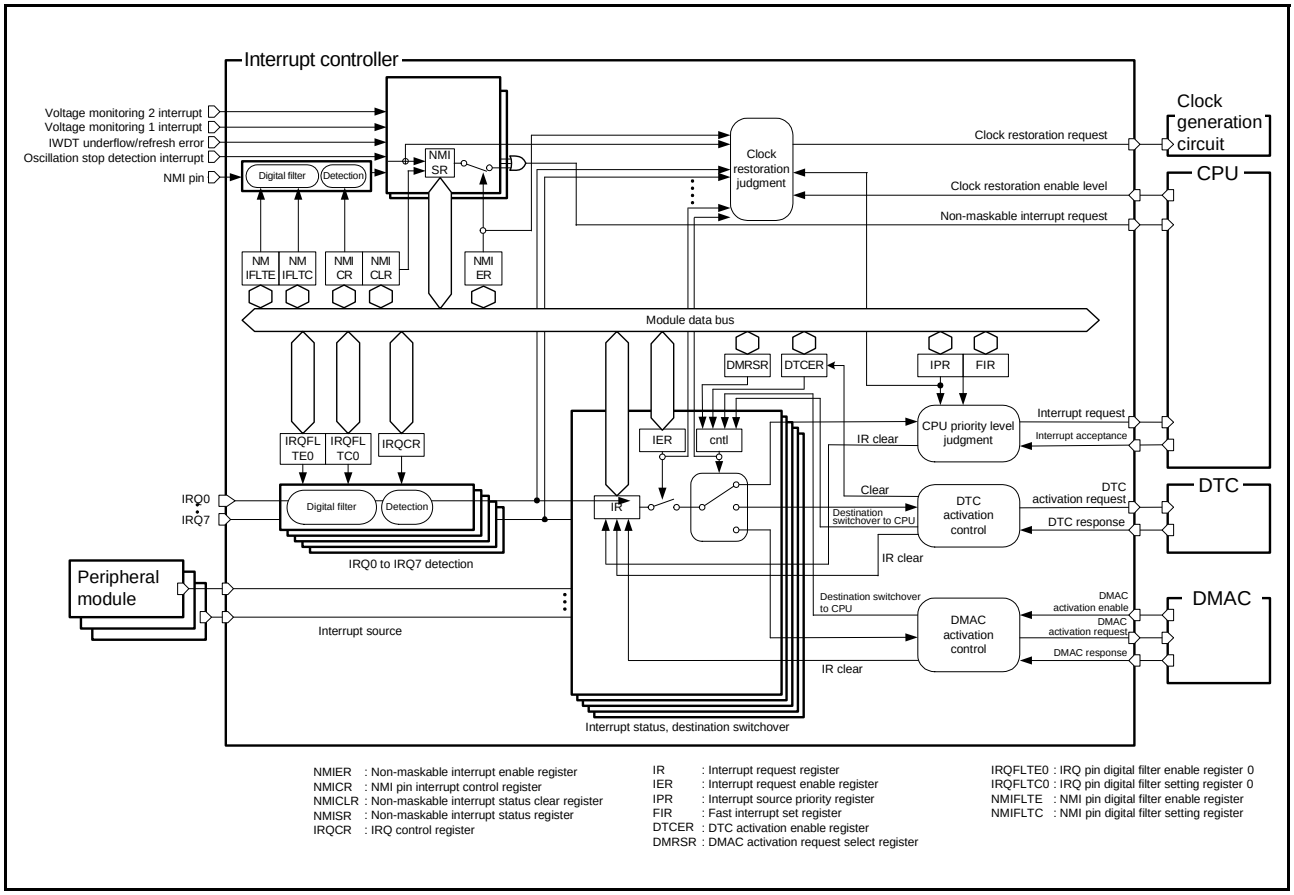


Figure 14.1 Block Diagram of Interrupt Controller

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 IRQ7	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 70F9h

	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 7). 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 249)

Address(es): 0008 7300h to 0008 73F9h

	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]

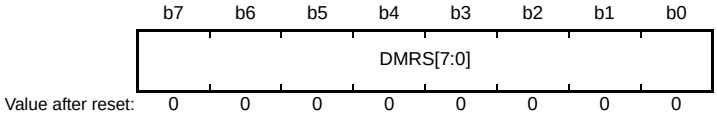
- When 1 is written to the DTCE bit

[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 7)

Address(es): 0008 7500h to 0008 7507h

	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 IRQ7.
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.

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

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 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: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/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 PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

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

14.2.11 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2ST	LVD1ST	IWDTST	—	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	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	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

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

14.2.12 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2EN	LVD1EN	IWDTEN	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/(W)
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.

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.

14.2.13 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2CLR	LVD1CLR	IWDTCLR	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/(W)
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.

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.

14.2.14 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.

14.2.15 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.

14.2.16 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: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/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 PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/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	—	×	×	×	×	×	—	—	—

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
—	Reserved	39	009Ch	—	×	×	×	×	×	—	—	—
—	Reserved	40	00A0h	—	×	×	×	×	×	—	—	—
—	Reserved	41	00A4h	—	×	×	×	×	×	—	—	—
—	Reserved	42	00A8h	—	×	×	×	×	×	—	—	—
—	Reserved	43	00ACh	—	×	×	×	×	×	—	—	—
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		—
—	Reserved	48	00C0h	—	×	×	×	×	×	—	—	—
—	Reserved	49	00C4h	—	×	×	×	×	×	—	—	—
—	Reserved	50	00C8h	—	×	×	×	×	×	—	—	—
—	Reserved	51	00CCh	—	×	×	×	×	×	—	—	—
—	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	—	×	×	×	×	×	—	—	—
RTC	CUP	63	00FCh	Edge	○	×	×	×	×	IER07.IEN7	IPR063	—
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
—	Reserved	72	0120h	—	×	×	×	×	×	—	—	—
—	Reserved	73	0124h	—	×	×	×	×	×	—	—	—
—	Reserved	74	0128h	—	×	×	×	×	×	—	—	—
—	Reserved	75	012Ch	—	×	×	×	×	×	—	—	—
—	Reserved	76	0130h	—	×	×	×	×	×	—	—	—
—	Reserved	77	0134h	—	×	×	×	×	×	—	—	—
—	Reserved	78	0138h	—	×	×	×	×	×	—	—	—
—	Reserved	79	013Ch	—	×	×	×	×	×	—	—	—
—	Reserved	80	0140h	—	×	×	×	×	×	—	—	—
—	Reserved	81	0144h	—	×	×	×	×	×	—	—	—
—	Reserved	82	0148h	—	×	×	×	×	×	—	—	—
—	Reserved	83	014Ch	—	×	×	×	×	×	—	—	—
—	Reserved	84	0150h	—	×	×	×	×	×	—	—	—
—	Reserved	85	0154h	—	×	×	×	×	×	—	—	—

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	86	0158h	—	×	×	×	×	×	—	—	—
—	Reserved	87	015Ch	—	×	×	×	×	×	—	—	—
LVD/CMPA	LVD1/CMPA1	88	0160h	Edge	○	×	×	○	○	IER0B.IEN0	IPR088	—
	LVD2/CMPA2	89	0164h	Edge	○	×	×	○	○	IER0B.IEN1	IPR089	—
—	Reserved	90	0168h	—	×	×	×	×	×	—	—	—
—	Reserved	91	016Ch	—	×	×	×	×	×	—	—	—
RTC	ALM	92	0170h	Edge	○	×	×	○	○	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	○	×	×	○	○	IER0B.IEN5	IPR093	—
—	Reserved	94	0178h	—	×	×	×	×	×	—	—	—
—	Reserved	95	017Ch	—	×	×	×	×	×	—	—	—
—	Reserved	96	0180h	—	×	×	×	×	×	—	—	—
—	Reserved	97	0184h	—	×	×	×	×	×	—	—	—
—	Reserved	98	0188h	—	×	×	×	×	×	—	—	—
—	Reserved	99	018Ch	—	×	×	×	×	×	—	—	—
—	Reserved	100	0190h	—	×	×	×	×	×	—	—	—
—	Reserved	101	0194h	—	×	×	×	×	×	—	—	—
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
—	Reserved	107	01ACh	—	×	×	×	×	×	—	—	—
—	Reserved	108	01B0h	—	×	×	×	×	×	—	—	—
—	Reserved	109	01B4h	—	×	×	×	×	×	—	—	—
—	Reserved	110	01B8h	—	×	×	×	×	×	—	—	—
—	Reserved	111	01BCh	—	×	×	×	×	×	—	—	—
—	Reserved	112	01C0h	—	×	×	×	×	×	—	—	—
—	Reserved	113	01C4h	—	×	×	×	×	×	—	—	—
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		—

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
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	—
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
—	Reserved	142	0238h	—	×	×	×	×	×	—	—	—
—	Reserved	143	023Ch	—	×	×	×	×	×	—	—	—
—	Reserved	144	0240h	—	×	×	×	×	×	—	—	—
—	Reserved	145	0244h	—	×	×	×	×	×	—	—	—
—	Reserved	146	0248h	—	×	×	×	×	×	—	—	—
—	Reserved	147	024Ch	—	×	×	×	×	×	—	—	—
—	Reserved	148	0250h	—	×	×	×	×	×	—	—	—
—	Reserved	149	0254h	—	×	×	×	×	×	—	—	—
—	Reserved	150	0258h	—	×	×	×	×	×	—	—	—
—	Reserved	151	025Ch	—	×	×	×	×	×	—	—	—
—	Reserved	152	0260h	—	×	×	×	×	×	—	—	—
—	Reserved	153	0264h	—	×	×	×	×	×	—	—	—
—	Reserved	154	0268h	—	×	×	×	×	×	—	—	—
—	Reserved	155	026Ch	—	×	×	×	×	×	—	—	—
—	Reserved	156	0270h	—	×	×	×	×	×	—	—	—
—	Reserved	157	0274h	—	×	×	×	×	×	—	—	—
—	Reserved	158	0278h	—	×	×	×	×	×	—	—	—
—	Reserved	159	027Ch	—	×	×	×	×	×	—	—	—
—	Reserved	160	0280h	—	×	×	×	×	×	—	—	—
—	Reserved	161	0284h	—	×	×	×	×	×	—	—	—
—	Reserved	162	0288h	—	×	×	×	×	×	—	—	—
—	Reserved	163	028Ch	—	×	×	×	×	×	—	—	—
—	Reserved	164	0290h	—	×	×	×	×	×	—	—	—
—	Reserved	165	0294h	—	×	×	×	×	×	—	—	—
—	Reserved	166	0298h	—	×	×	×	×	×	—	—	—
—	Reserved	167	029Ch	—	×	×	×	×	×	—	—	—
—	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		—

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
TMR1	CMIA1	177	02C4h	Edge	○	○	×	×	○	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	○	○	×	×	○	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	○	×	×	×	○	IER16.IEN3		—
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		—
—	Reserved	186	02E8h	—	×	×	×	×	×	—	—	—
—	Reserved	187	02ECh	—	×	×	×	×	×	—	—	—
—	Reserved	188	02F0h	—	×	×	×	×	×	—	—	—
—	Reserved	189	02F4h	—	×	×	×	×	×	—	—	—
—	Reserved	190	02F8h	—	×	×	×	×	×	—	—	—
—	Reserved	191	02FCh	—	×	×	×	×	×	—	—	—
—	Reserved	192	0300h	—	×	×	×	×	×	—	—	—
—	Reserved	193	0304h	—	×	×	×	×	×	—	—	—
—	Reserved	194	0308h	—	×	×	×	×	×	—	—	—
—	Reserved	195	030Ch	—	×	×	×	×	×	—	—	—
—	Reserved	196	0310h	—	×	×	×	×	×	—	—	—
—	Reserved	197	0314h	—	×	×	×	×	×	—	—	—
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
—	Reserved	202	0328h	—	×	×	×	×	×	—	—	—
—	Reserved	203	032Ch	—	×	×	×	×	×	—	—	—
—	Reserved	204	0330h	—	×	×	×	×	×	—	—	—
—	Reserved	205	0334h	—	×	×	×	×	×	—	—	—
—	Reserved	206	0338h	—	×	×	×	×	×	—	—	—
—	Reserved	207	033Ch	—	×	×	×	×	×	—	—	—
—	Reserved	208	0340h	—	×	×	×	×	×	—	—	—
—	Reserved	209	0344h	—	×	×	×	×	×	—	—	—
—	Reserved	210	0348h	—	×	×	×	×	×	—	—	—
—	Reserved	211	034Ch	—	×	×	×	×	×	—	—	—
—	Reserved	212	0350h	—	×	×	×	×	×	—	—	—
—	Reserved	213	0354h	—	×	×	×	×	×	—	—	—
—	Reserved	214	0358h	—	×	×	×	×	×	—	—	—
—	Reserved	215	035Ch	—	×	×	×	×	×	—	—	—
—	Reserved	216	0360h	—	×	×	×	×	×	—	—	—
—	Reserved	217	0364h	—	×	×	×	×	×	—	—	—
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		—

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
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		—
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		—
—	Reserved	230	0398h	—	×	×	×	×	×	—	—	—
—	Reserved	231	039Ch	—	×	×	×	×	×	—	—	—
—	Reserved	232	03A0h	—	×	×	×	×	×	—	—	—
—	Reserved	233	03A4h	—	×	×	×	×	×	—	—	—
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	—
—	Reserved	250	03E8h	—	×	×	×	×	×	—	—	—
—	Reserved	251	03ECh	—	×	×	×	×	×	—	—	—
—	Reserved	252	03F0h	—	×	×	×	×	×	—	—	—
—	Reserved	253	03F4h	—	×	×	×	×	×	—	—	—
—	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.

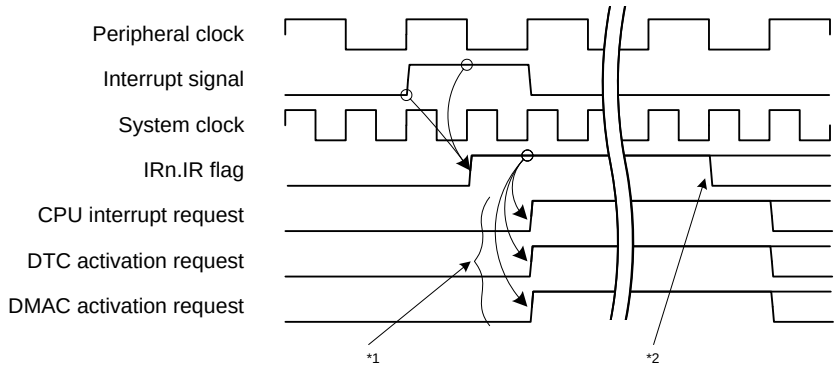
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation, or DMAC activation)
- Determining priority

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 7) 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 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK 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.

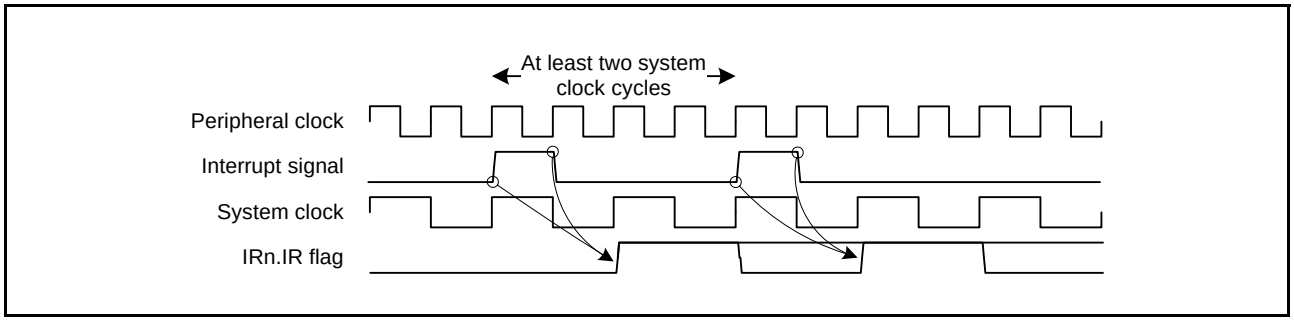


Figure 14.3 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Clock)

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 27, Serial Communications Interface (SCIe, SCIf), section 29, I2C Bus Interface (RIIC), and section 30, Serial Peripheral Interface (RSPI).

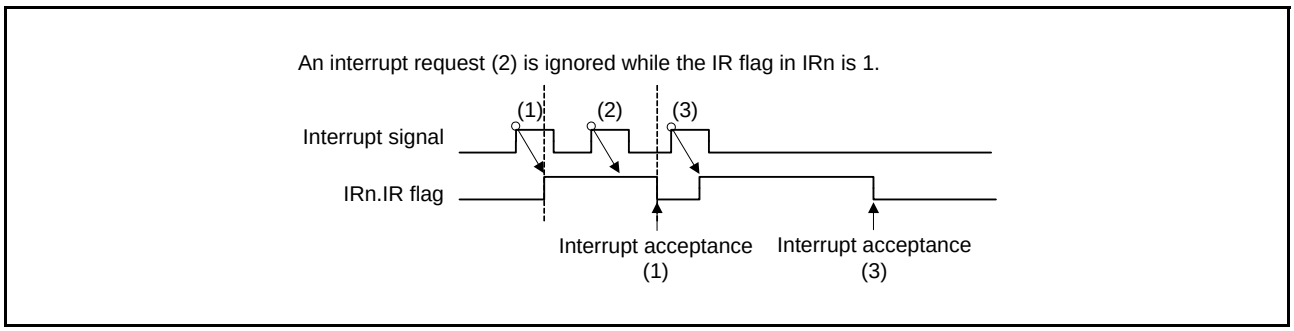


Figure 14.4 Timing for IRn.IR Flag Re-Setting

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.

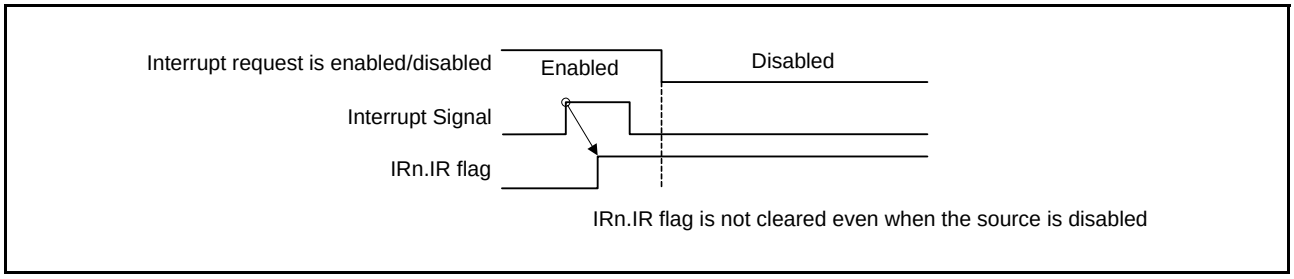


Figure 14.5 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

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.

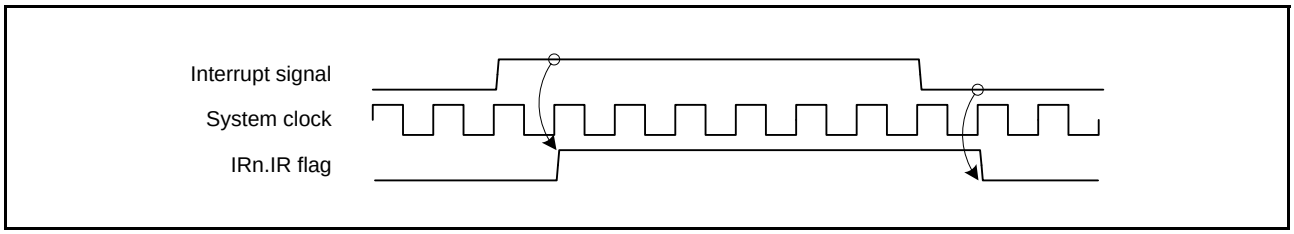


Figure 14.6 IRn.IR Flag Operation for Level Detection Interrupts

Figure 14.7 shows the procedure for handling level detection interrupts.

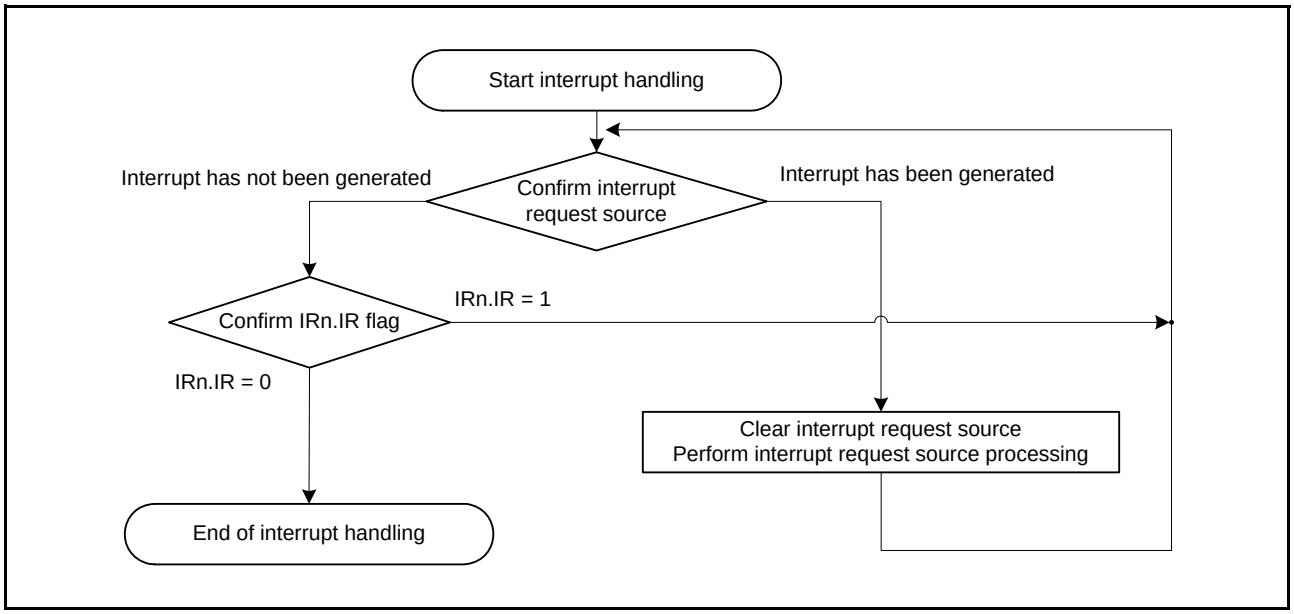


Figure 14.7 Procedure for Handling Level Detection Interrupts

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 27, Serial Communications Interface (SCIe, SCIf), section 29, I2C Bus Interface (RIIC), and section 30, 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 16.3.7, Activating the DMAC in section 16, 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 17.5, DTC Setting Procedure, in section 17, 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 17.3, Chain Transfer Conditions in section 17, 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 16, 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 7) and NMI pin interrupt.
The digital filter samples input signals at the filter sampling clock (PCLK) 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 (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits (i = 0 to 7) and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/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.

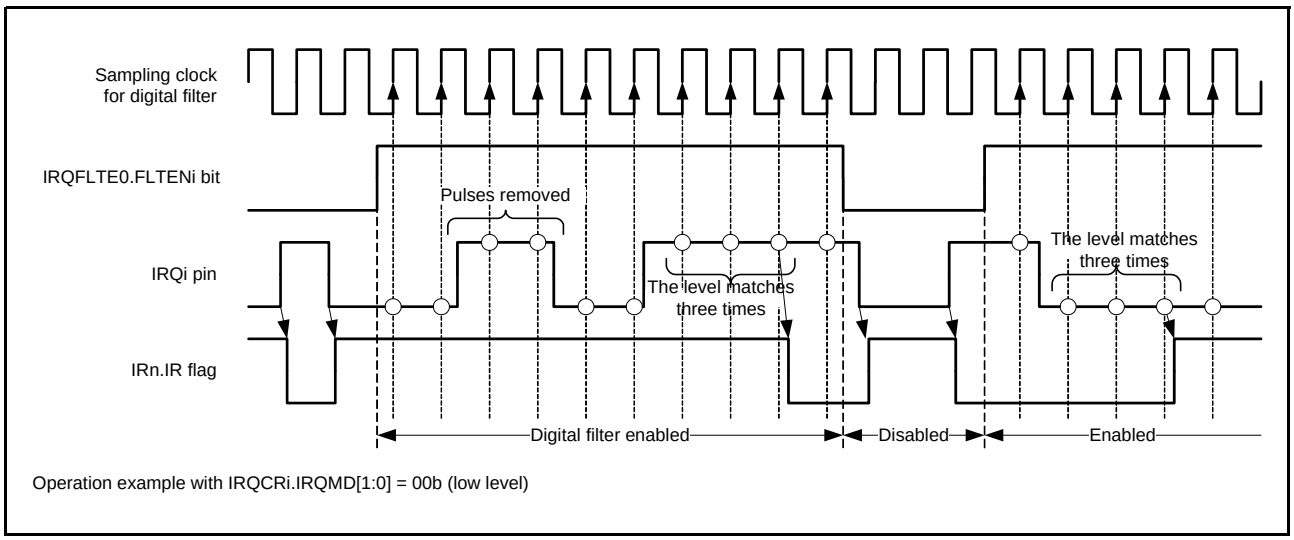


Figure 14.8 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.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.FLTENi bit (i = 0 to 7) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTC0.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.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DMAC activation, set the DMRSRm.DMRS[7:0] bits. If the interrupt is to be used for DTC activation, set the DTCERn.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, 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.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.FLTENi = 0, NMIFLTE.NFLTEN = 0).
2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE0.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 bus 1) • Operates in synchronization with the peripheral-module clock (PCLKB and PCLKD*1)
	Internal peripheral bus 6	• Connected to ROM (P/E) and E2 DataFlash memory • Operates in synchronization with the FlashIF clock (FCLK)

Note 1. The peripheral module clock used as the operating clock is PCLKD for S12AD.
P/E: Programming/Erasure

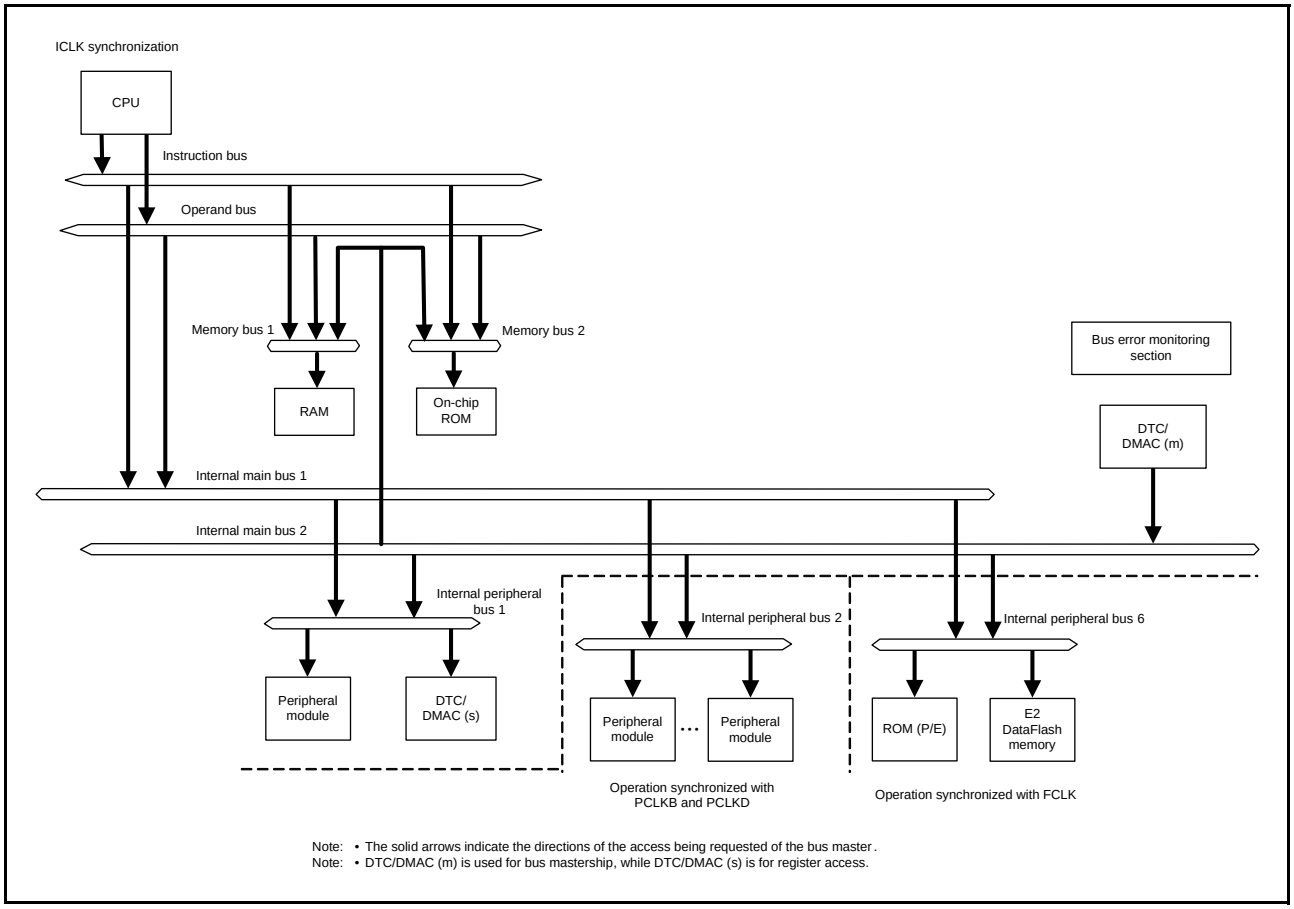


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0001 0000h to 0007 FFFFh		Reserved area
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	E2 DataFlash memory and ROM (for programming/erasure)
0500 0000h to 7FFF FFFFh	Reserved area	Reserved area
8000 0000h to FEFF FFFFh	Memory bus 2	ROM (for reading only)
FF00 0000h to FFFF FFFFh		

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 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, 2, and 6), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Table 15.3 Order of Priority for Bus Masters

Priority	Bus Master
High ↑ 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 bus 1
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, 2, 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).

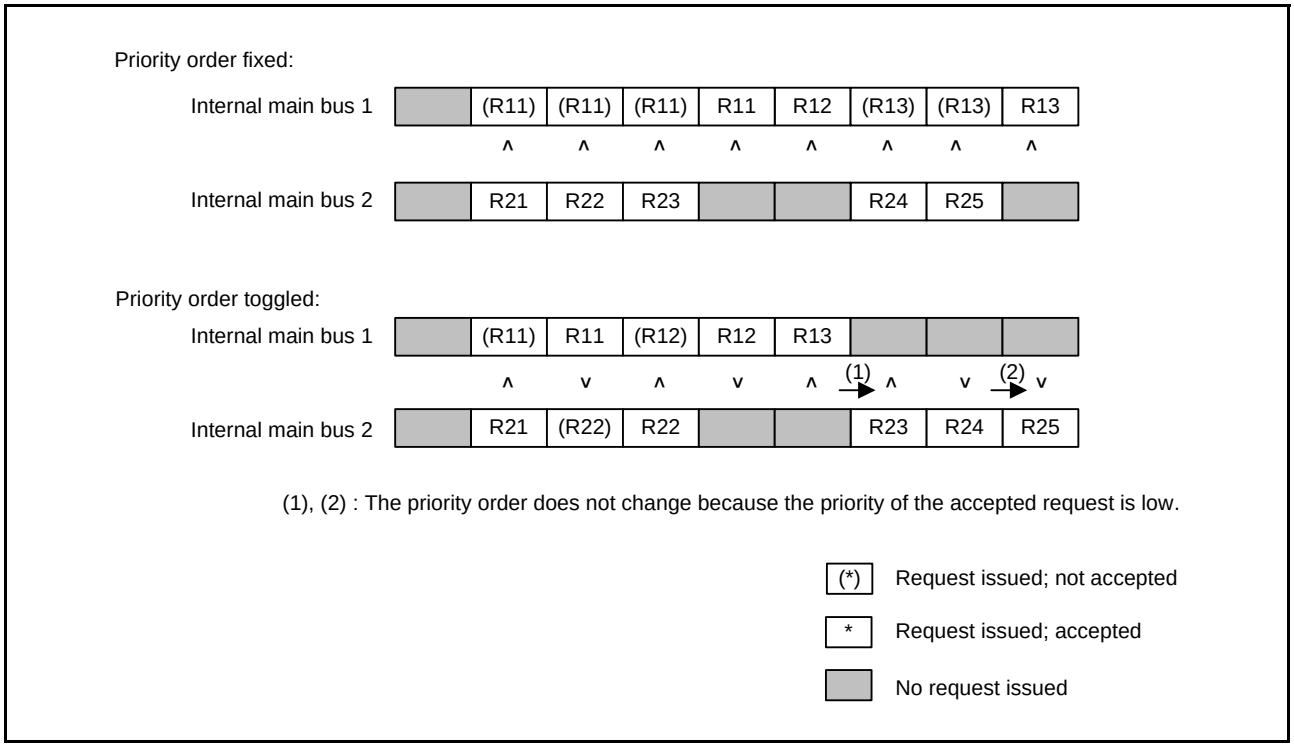


Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses

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).

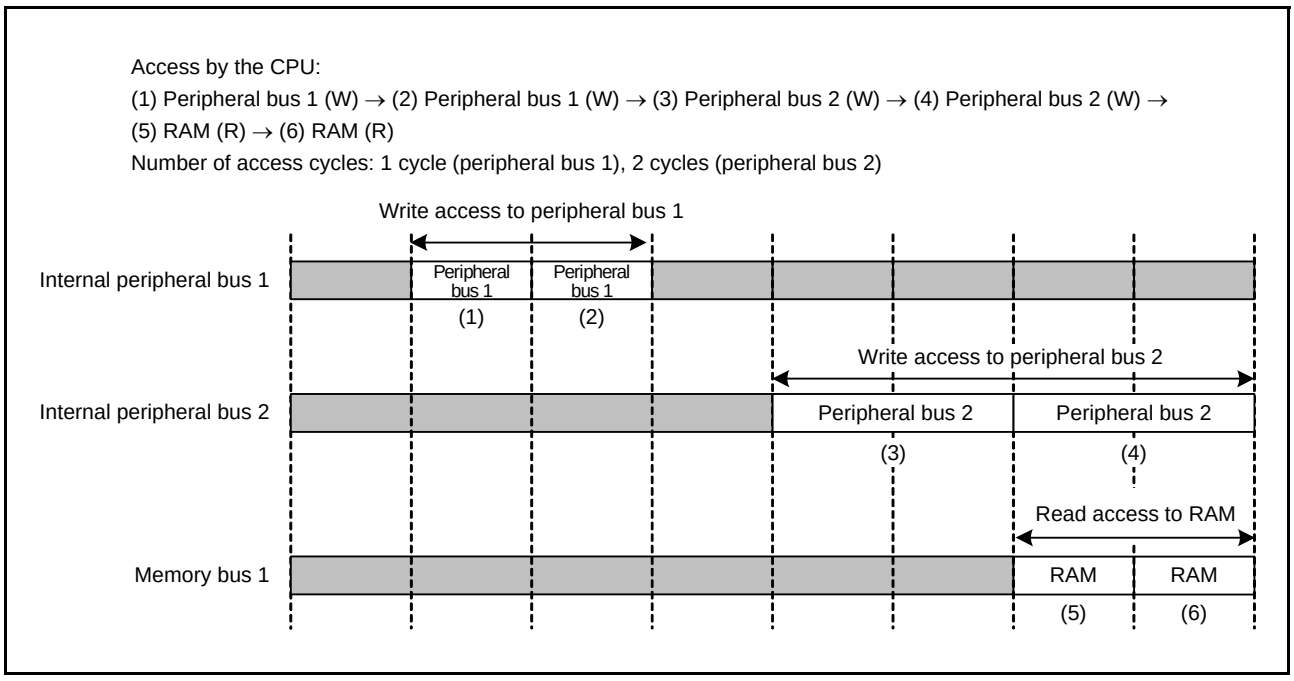


Figure 15.3 Write Buffer Function

15.2.6 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 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 during access to RAM and ROM by the CPU.

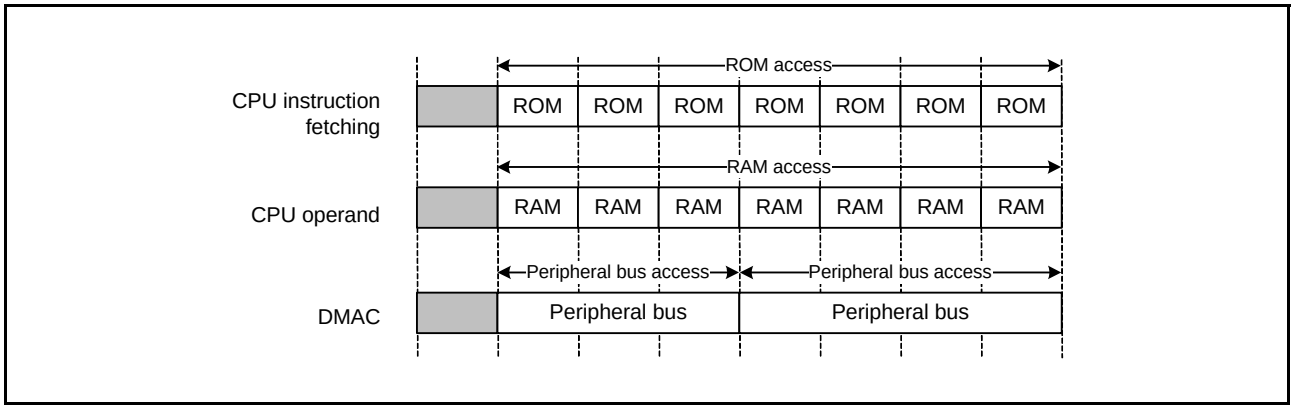


Figure 15.4 Example of Parallel Operations

15.2.7 Restrictions

(1) Prohibition of Access that Spans Multiple 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. Ensure that a single word or long-word access does not span across two areas by crossing address space area boundaries.

(2) Restrictions in relation to RMPA and string-manipulation instructions

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

15.3 Register Descriptions

15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

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

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

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

15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	—	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
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC/DMAC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BPFB[1:0]		—	—	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 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
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC and DMAC are stopped. 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 Priority Control)

These bits specify the priority order for internal peripheral bus 2.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

15.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

15.4.1 Types of Bus Error

There is one type of bus error: illegal address access.
Illegal address access is the detection of illegal access to an area.

15.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access of the following types leads to illegal address access errors.

- The address ranges where access will lead to illegal address access errors are indicated in Table 15.5.

15.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The IERn register in the ICU can specify whether to generate an interrupt in the case of a bus error.

15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the types of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1, 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 15.5 Types of Bus Errors

Address	Type of Area	Type of Error
		Illegal Address Access
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	Reserved area	—
000C 0000h to 000D FFFFh	Reserved area	○
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Δ
0500 0000h to 07FF FFFFh	Reserved area	○
0800 0000h to 0FFF FFFFh	Reserved area	—
1000 0000h to 7FFF FFFFh	Reserved area	○
8000 0000h to FFFF FFFFh	Memory bus 2	—

—: Bus error not generated
Δ: Bus error generation behavior undefined
○: Bus error generated
Note: • The capacity of the RAM, E2 DataFlash, and ROM differs depending on the product. For details, see section 35, RAM, section 36, ROM (Flash Memory for Code Storage), and section 37, E2 DataFlash Memory (Flash Memory for Data Storage).

16. DMA Controller (DMACA)

The RX220 Group 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.

16.1 Overview

Table 16.1 lists the specifications of the DMAC, and Figure 16.1 shows a block diagram of the DMAC.

Table 16.1 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.

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

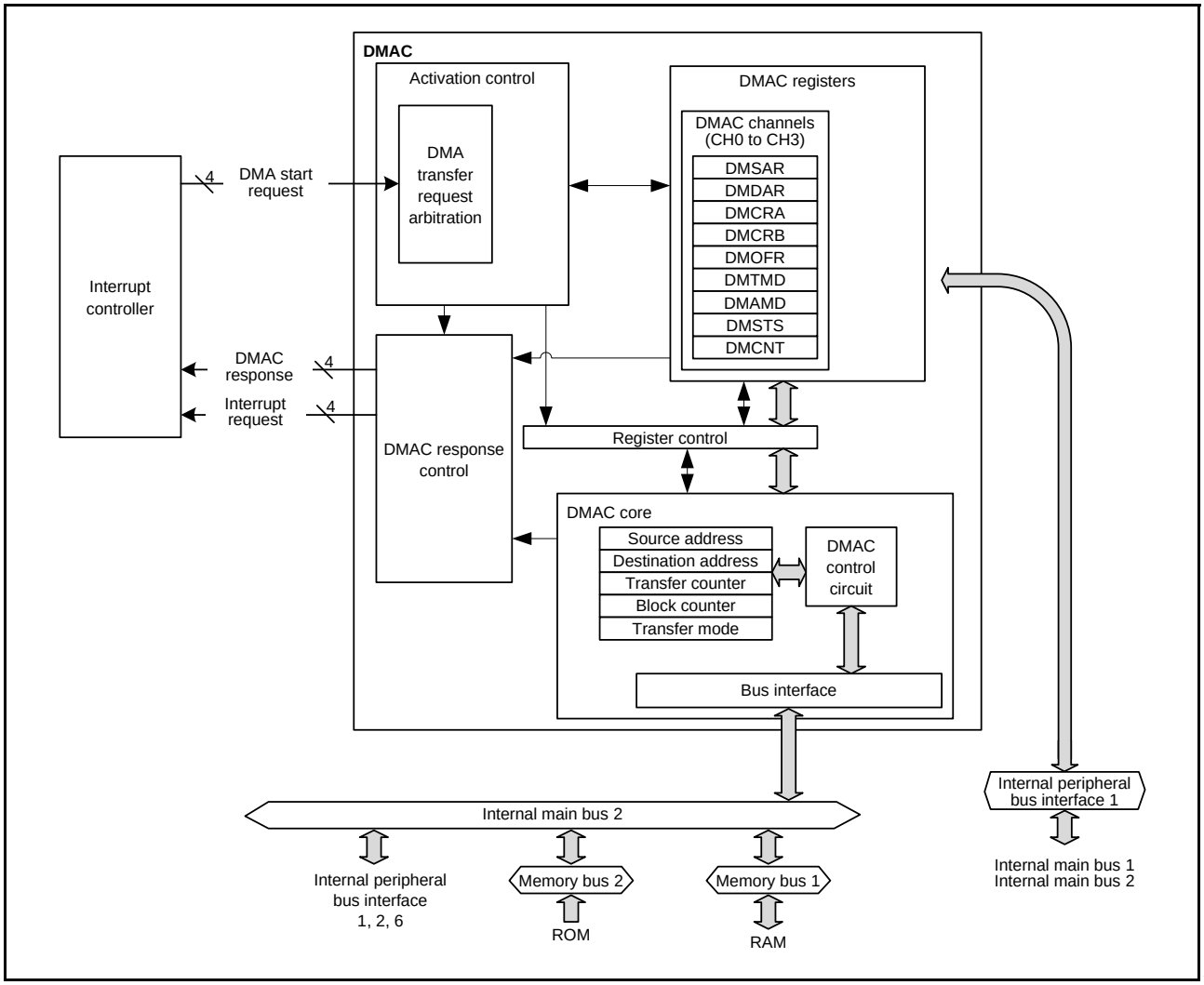


Figure 16.1 Block Diagram of DMAC

16.2 Register Descriptions

16.2.1 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.

16.2.2 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

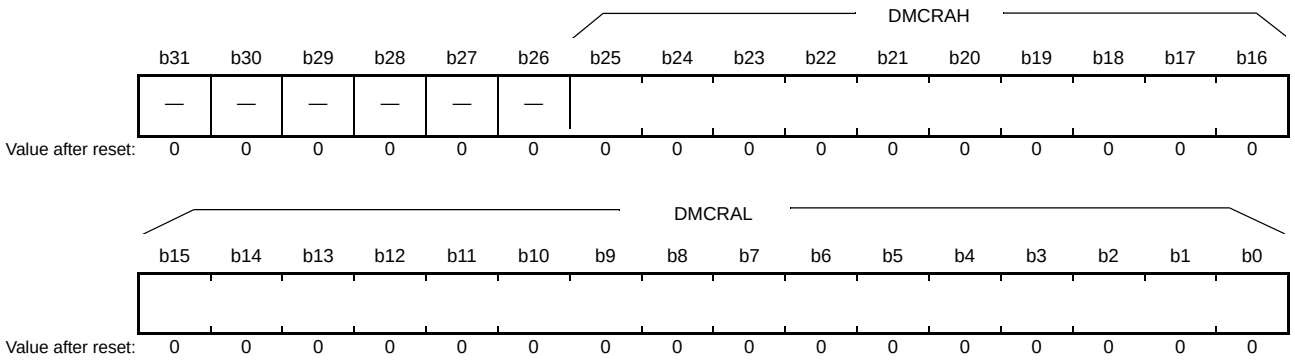
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

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.

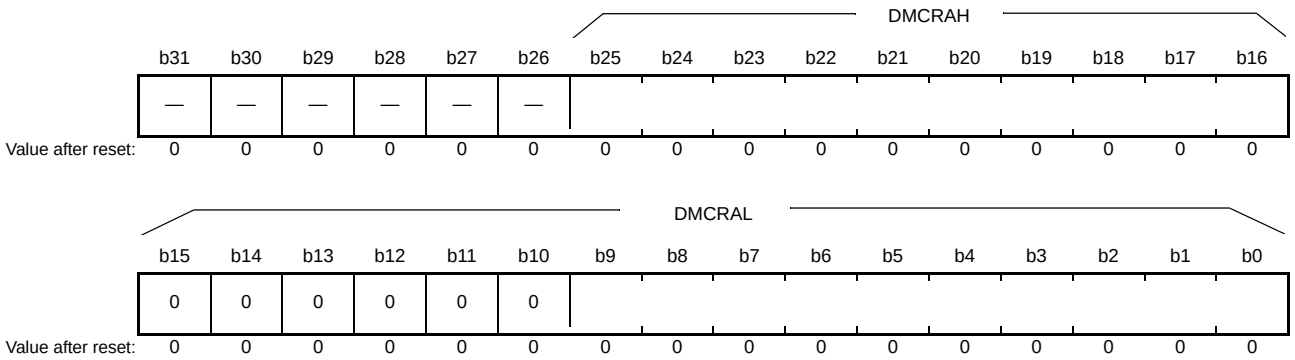
16.2.3 DMA Transfer Count Register (DMCRA)

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.

16.2.4 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.

16.2.5 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.

16.2.6 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.

16.2.7 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 16.2.	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 16.2.	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 16.2 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 16.2 lists the settings and the corresponding extended repeat areas.

SM 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 16.2 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.

16.2.8 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.

16.2.9 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.

16.2.10 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.

16.2.11 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

16.2.12 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.

16.2.13 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

16.3 Operation

16.3.1 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 16.3 summarizes the register update operation in normal transfer mode, and Figure 16.2 shows the operation in normal transfer mode.

Table 16.3 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.

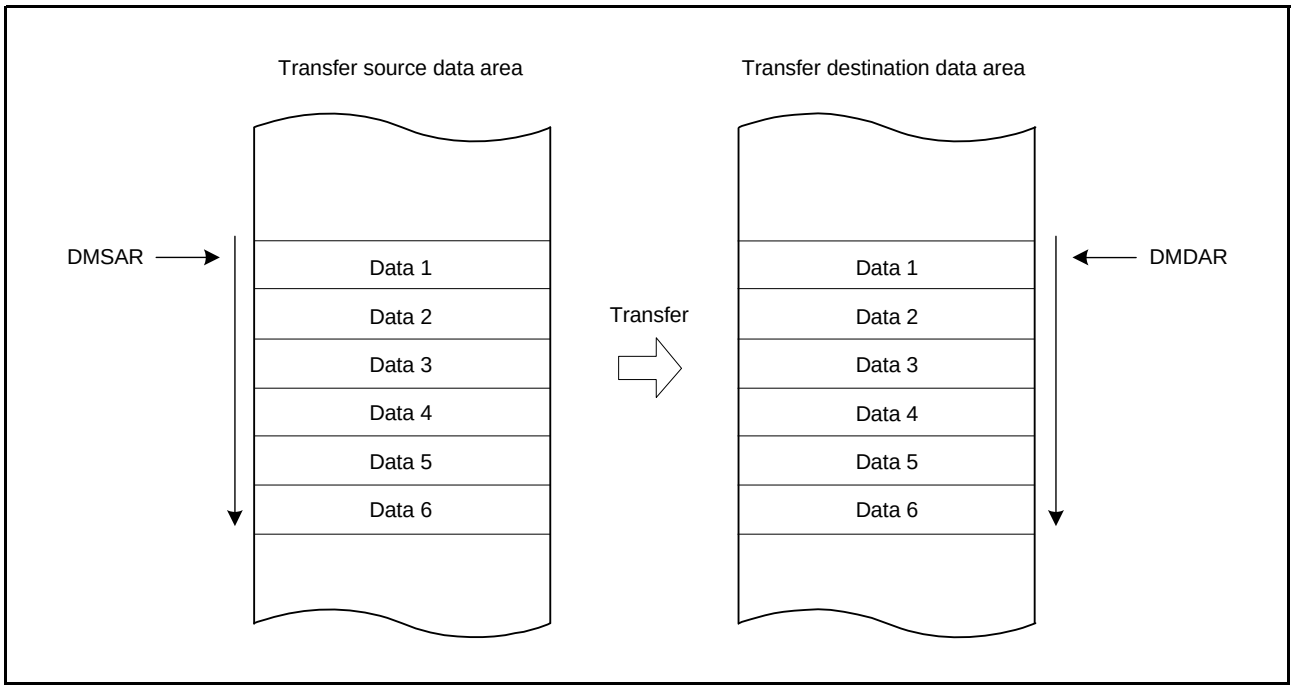


Figure 16.2 Operation in Normal Transfer Mode

(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 16.4 summarizes the register update operation in repeat transfer mode, and Figure 16.3 shows the operation in repeat transfer mode.

Table 16.4 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.

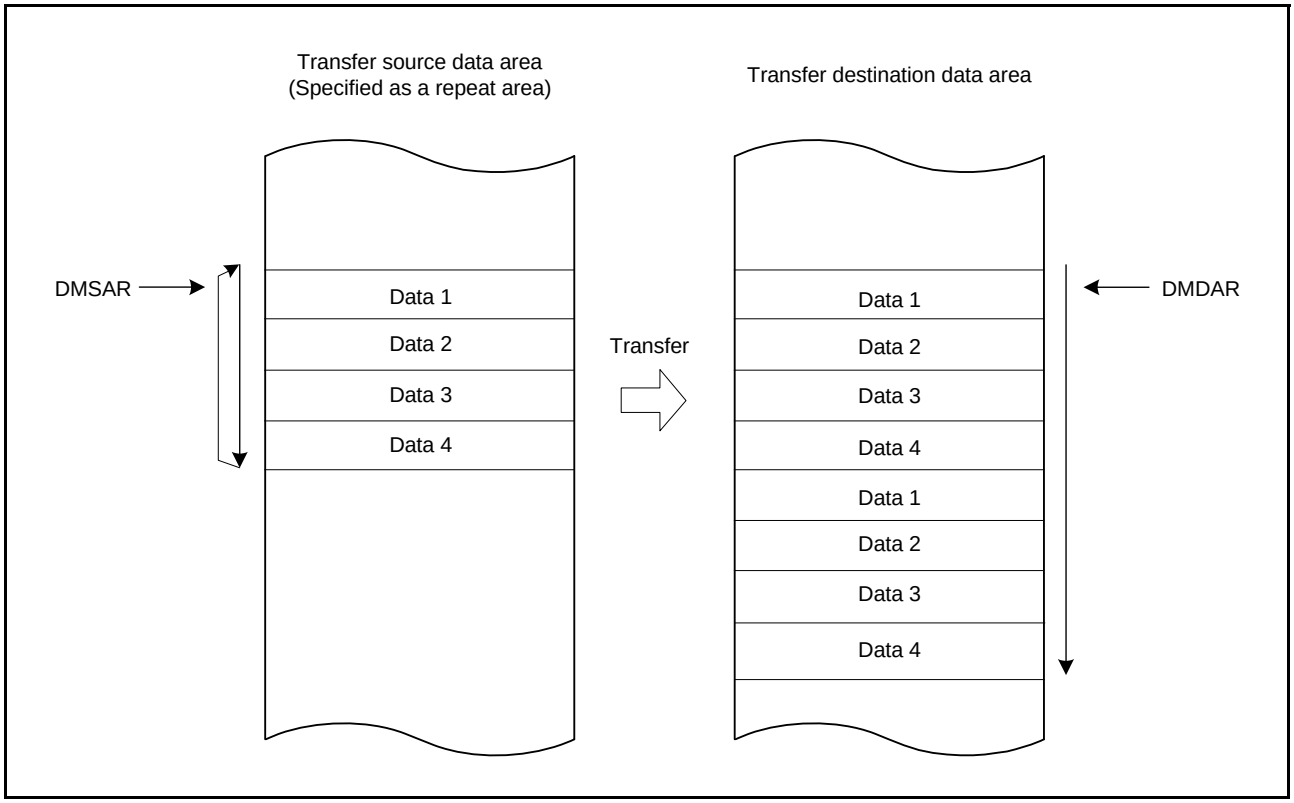


Figure 16.3 Operation in Repeat Transfer Mode

(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 16.5 summarizes the register update operation in block transfer mode, and Figure 16.4 shows the operation in block transfer mode.

Table 16.5 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.

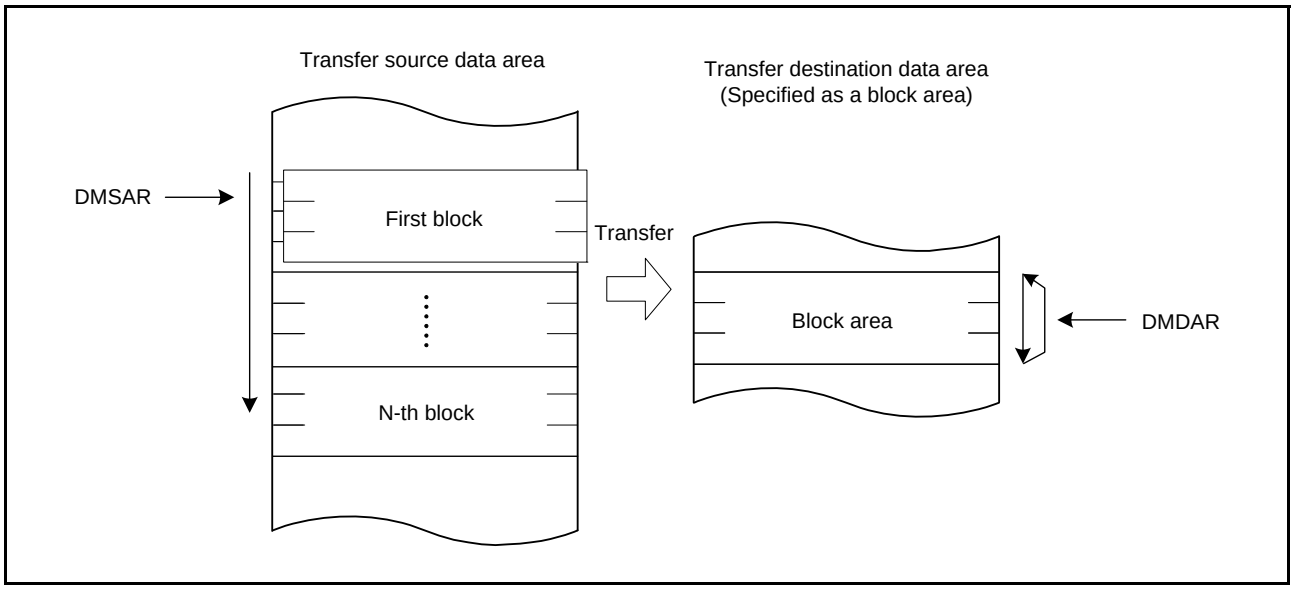


Figure 16.4 Operation in Block Transfer Mode

16.3.2 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 16.5 shows an example of the extended repeat area operation.

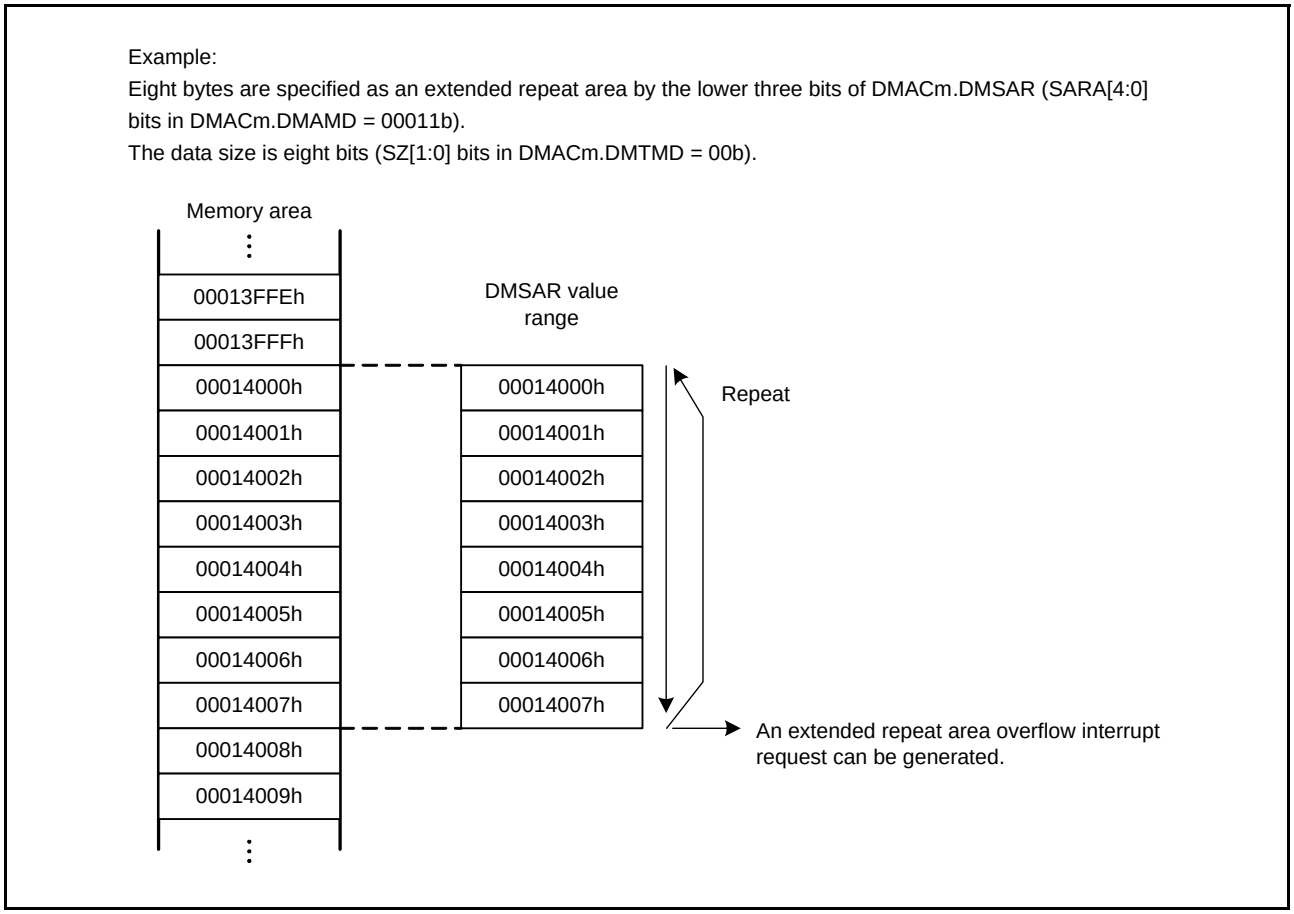


Figure 16.5 Example of Extended Repeat Area Operation

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 16.6 shows an example when the extended repeat area function is used in block transfer mode.

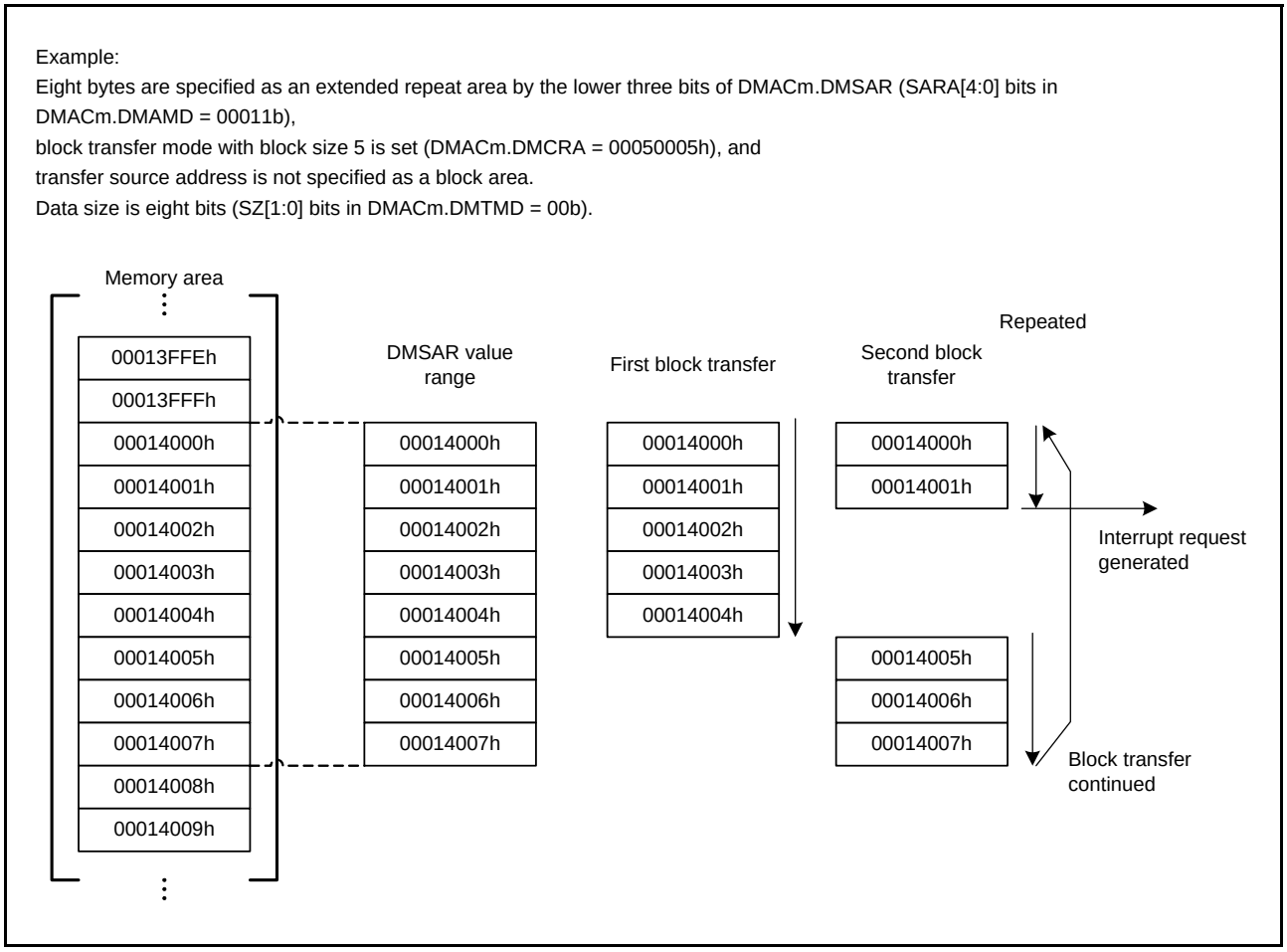


Figure 16.6 Example of Extended Repeat Area Function in Block Transfer Mode

16.3.3 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 16.6 shows the address update method in each address update mode.

Table 16.6 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 16.7 shows an example of address updating using offset addition.

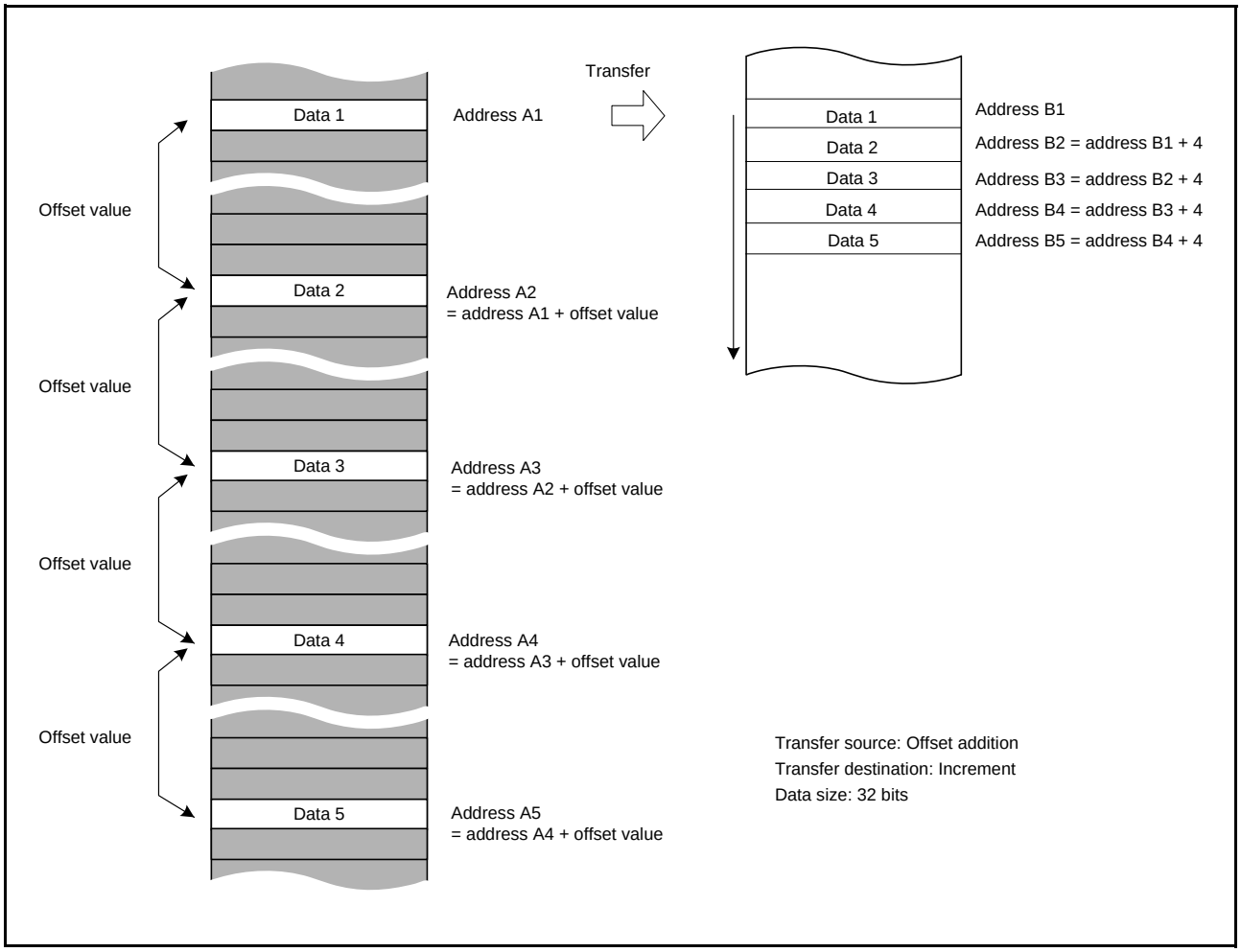


Figure 16.7 Example of Address Updating by Offset Addition

In Figure 16.7, 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 16.8 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.

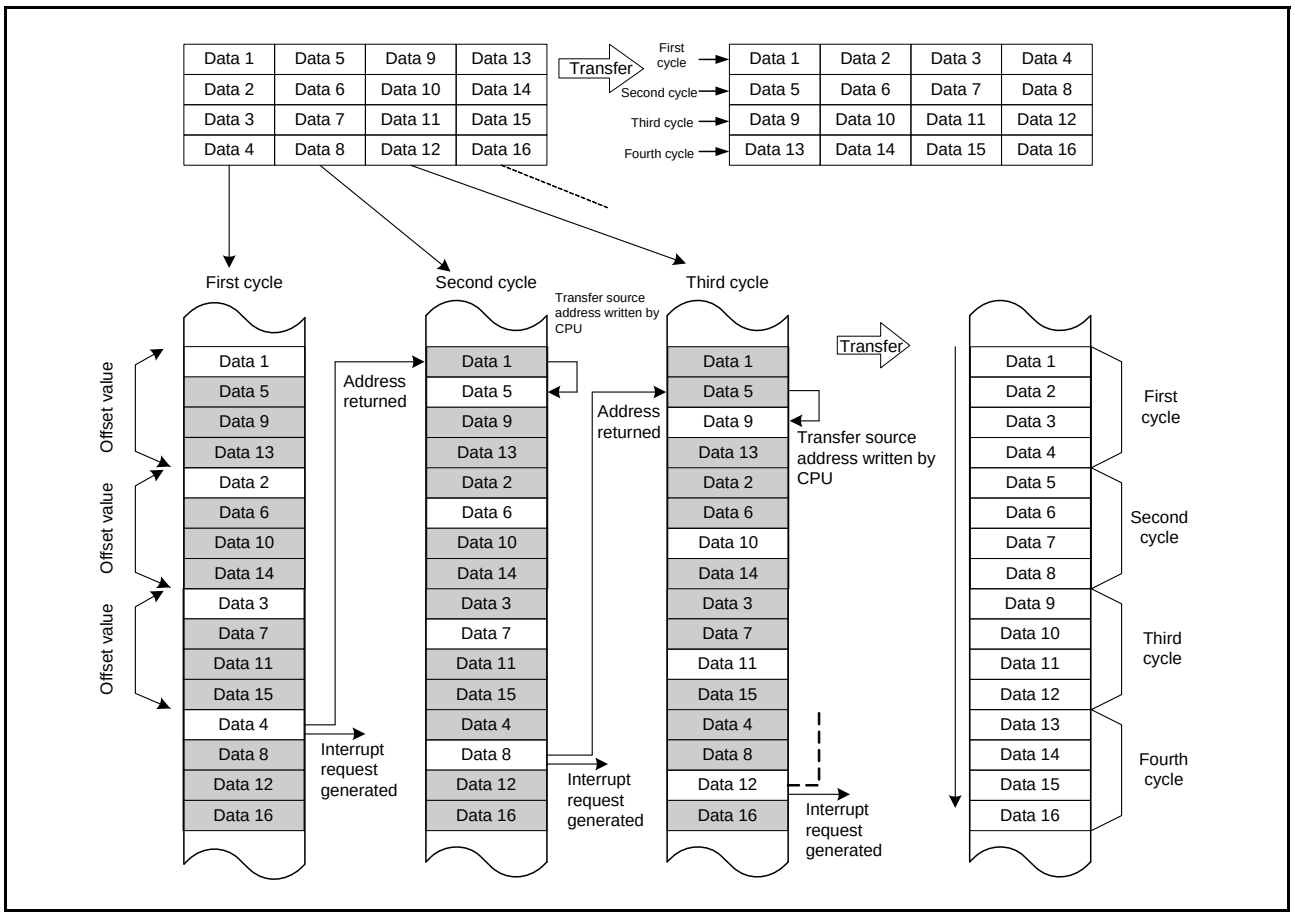


Figure 16.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

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 16.9 shows a flowchart of the XY conversion.

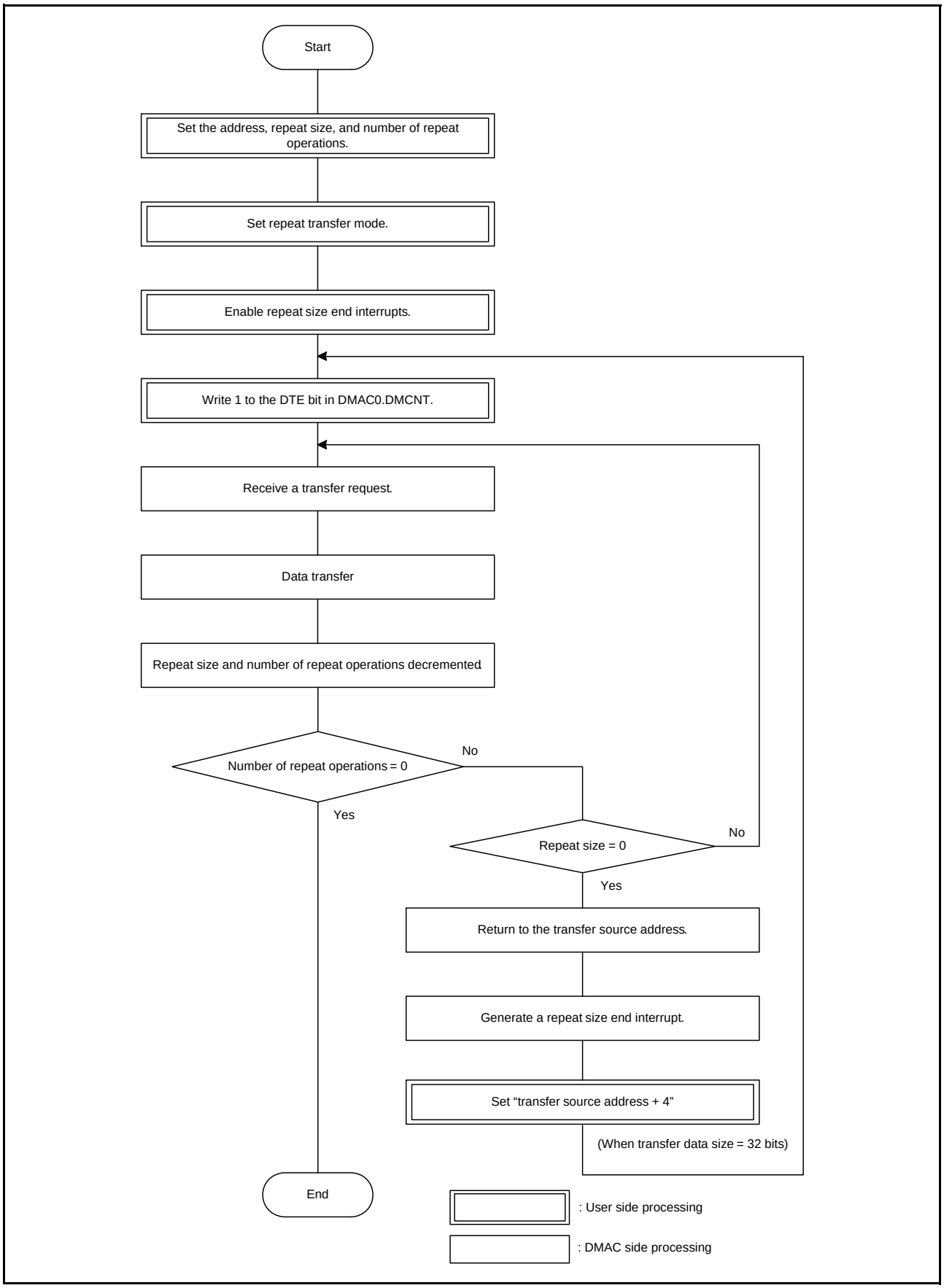


Figure 16.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

16.3.4 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).

16.3.5 Operation Timing

Figure 16.10 and Figure 16.11 show DMAC operation timing examples.

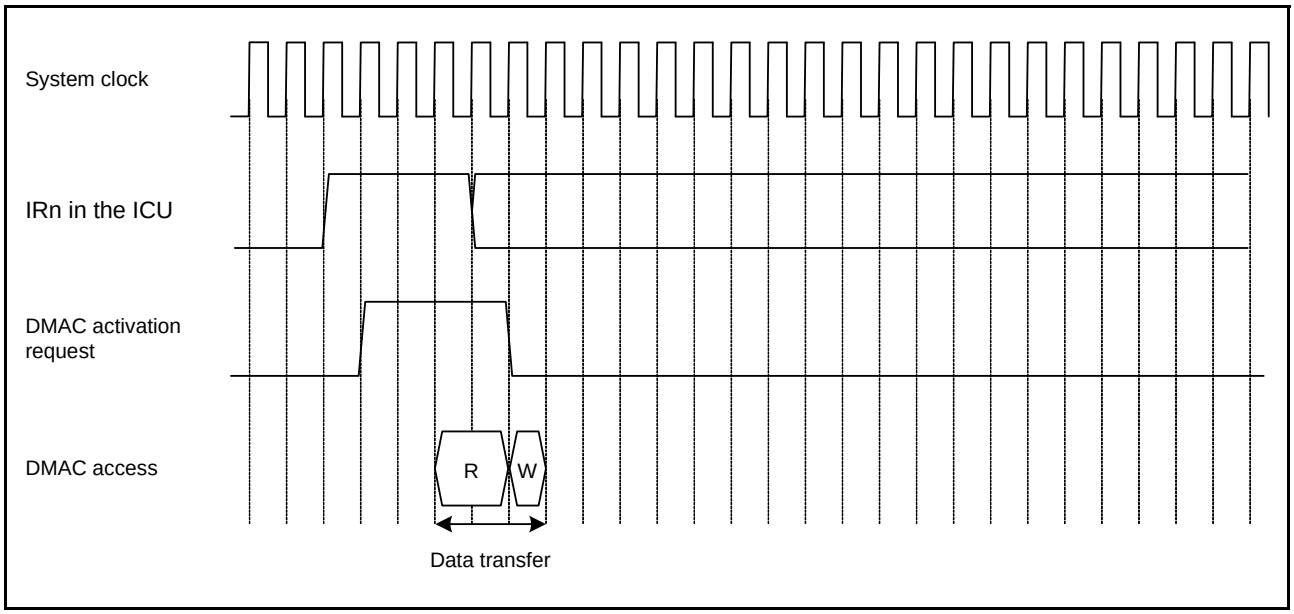


Figure 16.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

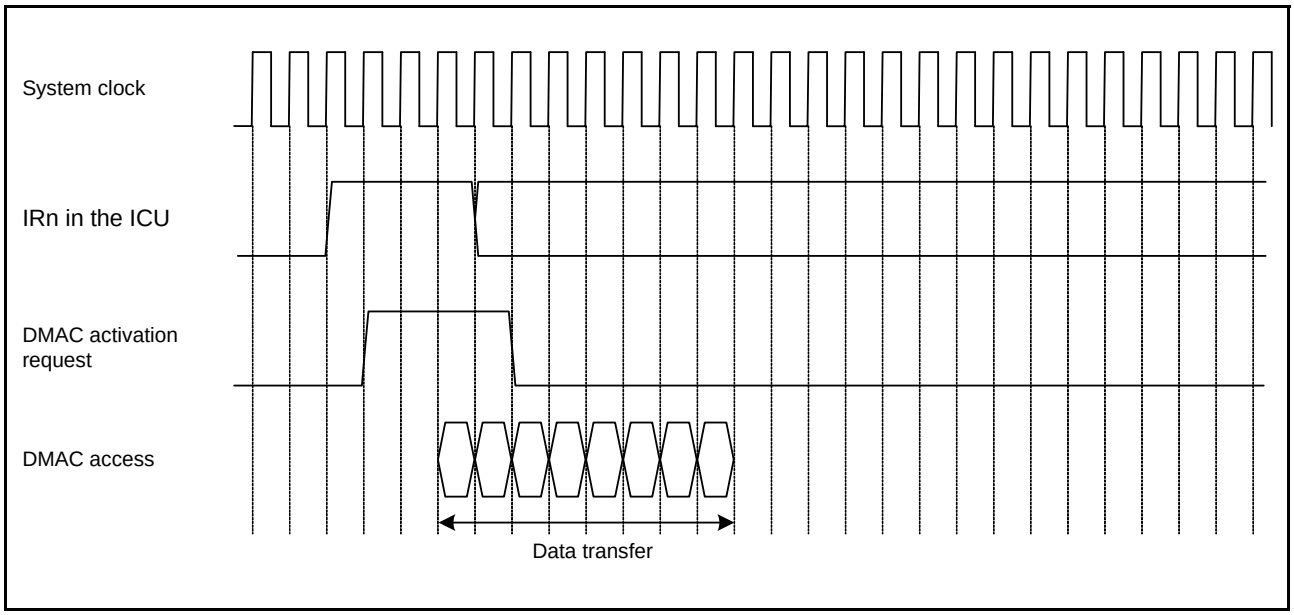


Figure 16.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

16.3.6 DMAC Execution Cycles

Table 16.7 lists execution cycles in one DMAC data transfer operation.

Table 16.7 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 35, RAM, section 36, ROM (Flash Memory for Code Storage), section 5, I/O Registers.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 16.3.5, Operation Timing.

16.3.7 Activating the DMAC

Figure 16.12 shows the register setting procedure.

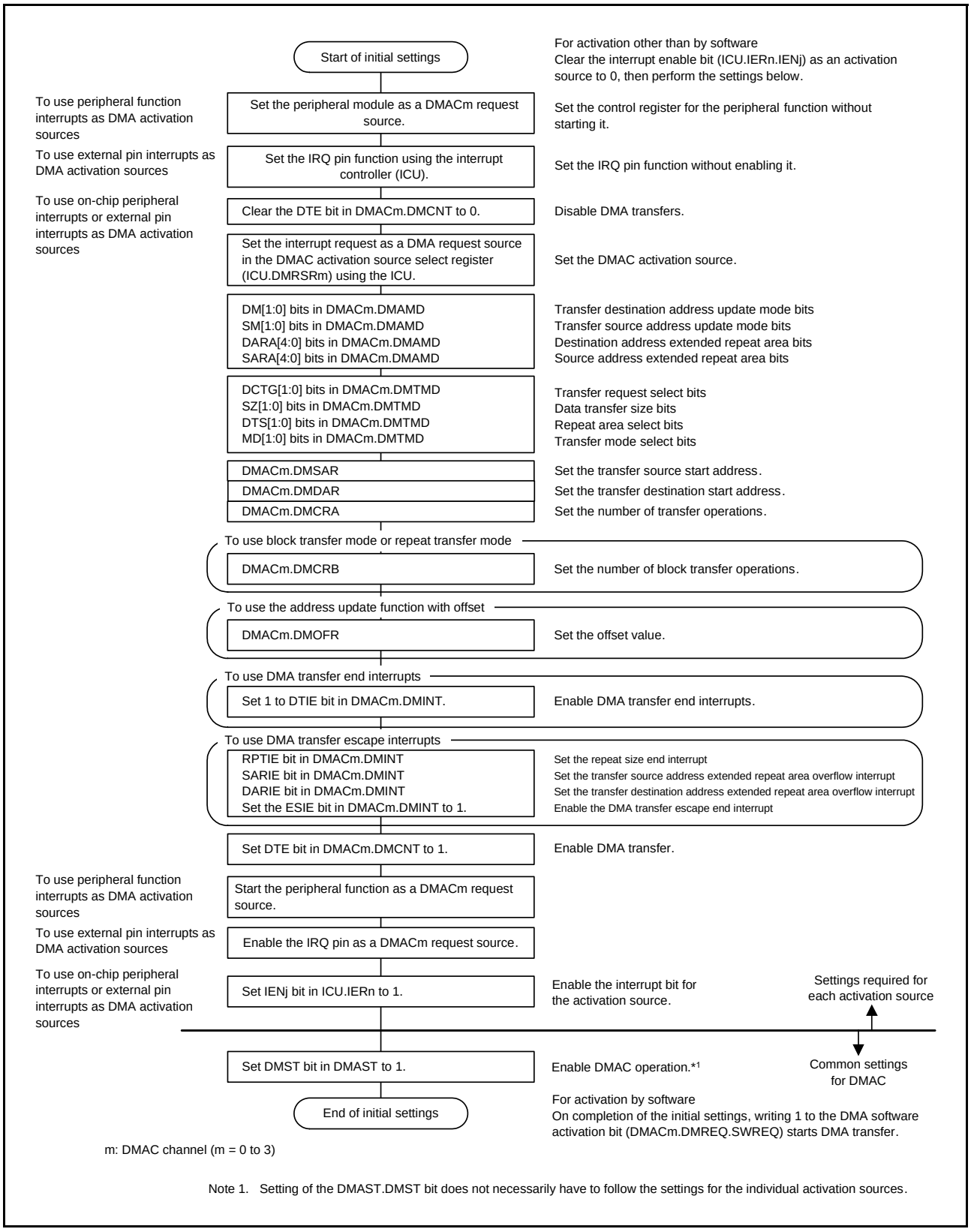


Figure 16.12 Register Setting Procedure

16.3.8 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).

16.3.9 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 16.3 to Table 16.5.

(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 16.3 to Table 16.5.

(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 16.3 to Table 16.5.

(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 16.3 to Table 16.5.

(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).

16.3.10 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.

16.4 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.

16.4.1 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).

16.4.2 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).

16.4.3 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).

16.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 16.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 16.13 shows the schematic logic diagram of interrupt outputs. Figure 16.14 shows the DMAC interrupt handling routine to resume or terminate DMA transfer.

Table 16.8 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		

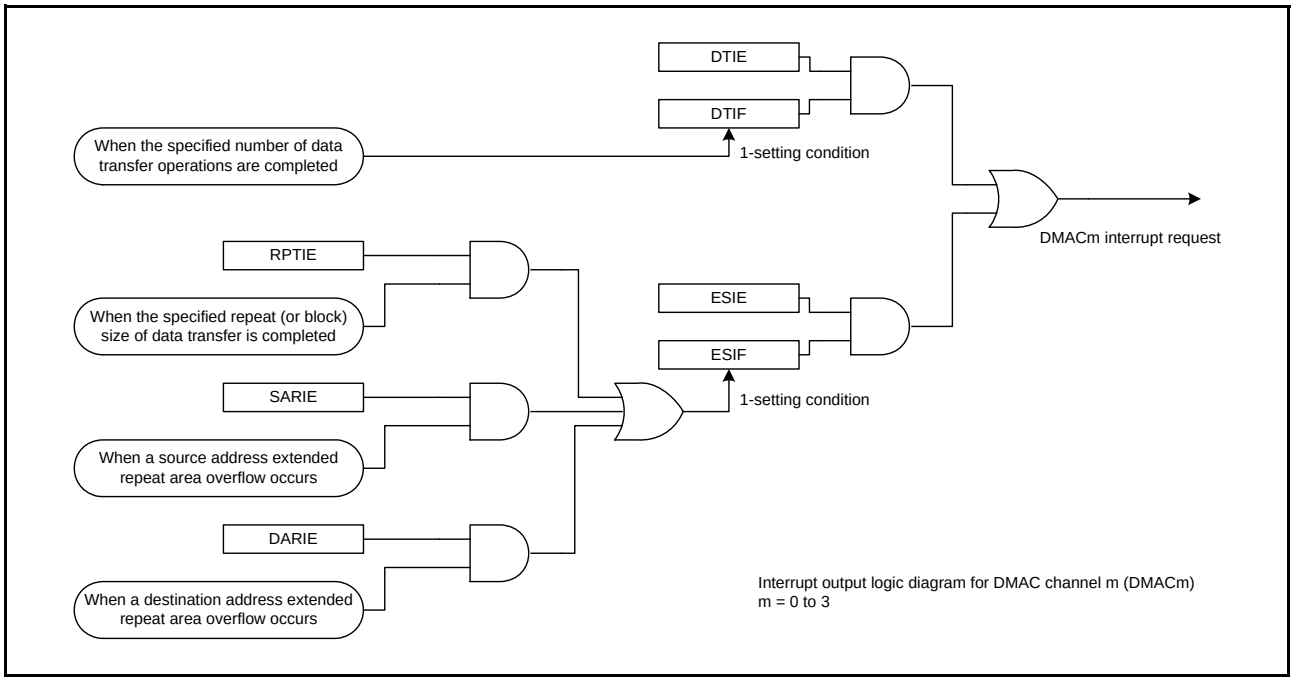


Figure 16.13 Schematic Logic Diagram of Interrupt Outputs

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.

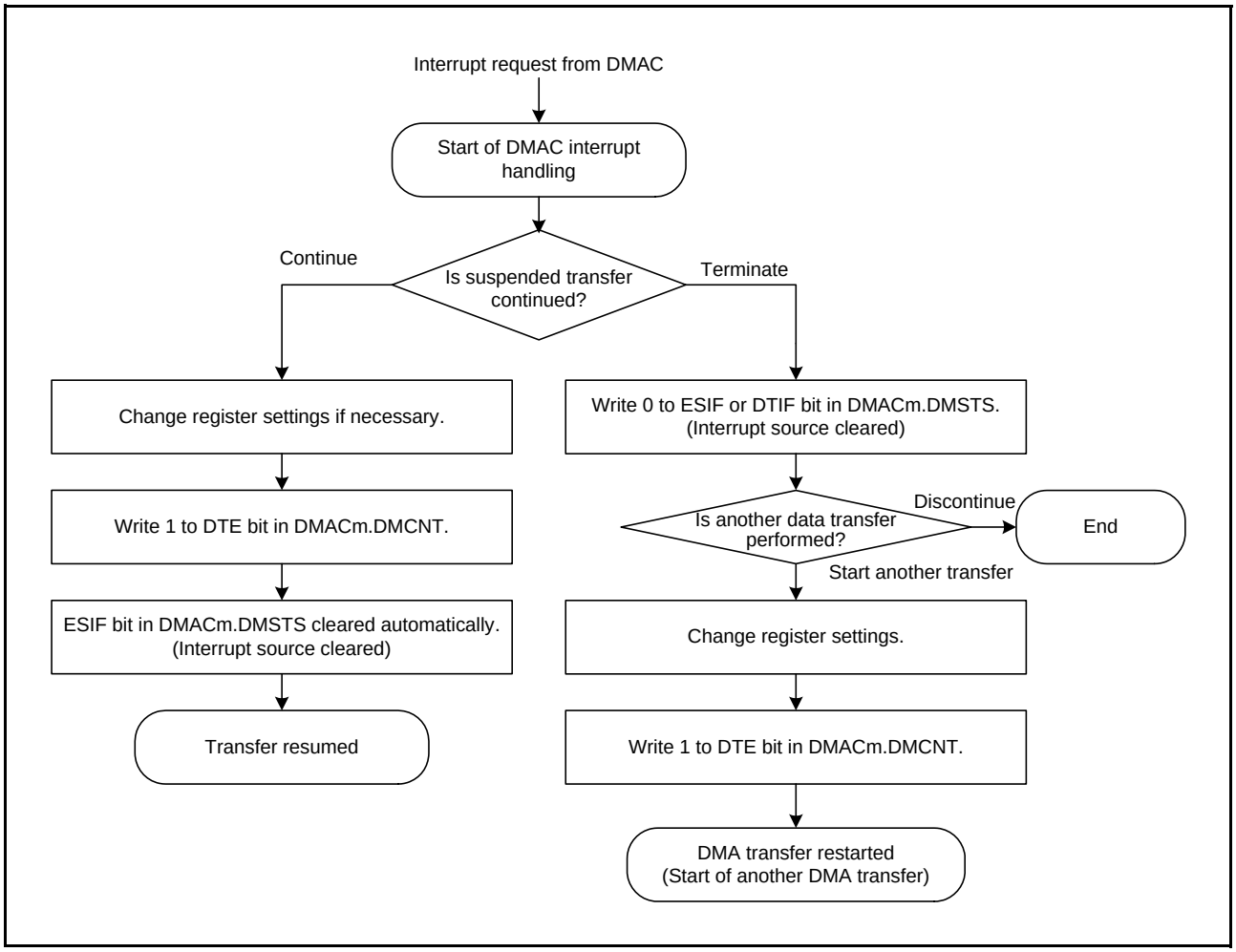


Figure 16.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

16.6 Low Power Consumption Function

Before transition to the module stop state, all-module clock stop mode, or 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 Mode

Make settings in accord with the procedure under section 11.6.3.1, Transition to Software Standby Mode, in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby follows the completion of DMA transfer.

(4) Note on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.7.5, 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.

16.7 Usage Notes

16.7.1 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.

16.7.2 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).

16.7.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

16.7.4 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 16.2.12, DMA Activation Source Flag Control Register (DMCSL).

16.7.5 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).

16.7.6 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 16.3.7, Activating the DMAC.

17. Data Transfer Controller (DTCa)

The RX220 Group incorporates a data transfer controller (DTC).
The DTC is activated by an interrupt request to control data transfer.

17.1 Overview

Table 17.1 lists the specifications of the DTC, and Figure 17.1 shows a block diagram of the DTC.

Table 17.1 DTC Specifications

Item	Description
Transfer mode	• 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 repeat size is 256. • Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 256 data.
Transfer channel	• Channel transfer corresponding to the interrupt source is possible (transferred by DTC activation request from the ICU). • Data of multiple channels 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 excepting reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh excepting reserved areas)
Data transfer units	• Length of a single data: 8, 16, or 32 bits • Number of data for a single block: 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	Event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Transfer data read skip can be specified.
Write-back skip	When “fixed” is selected for transfer source address and/or transfer destination address, write-back skip execution is provided.
Lower power consumption function	Module stop state can be specified.

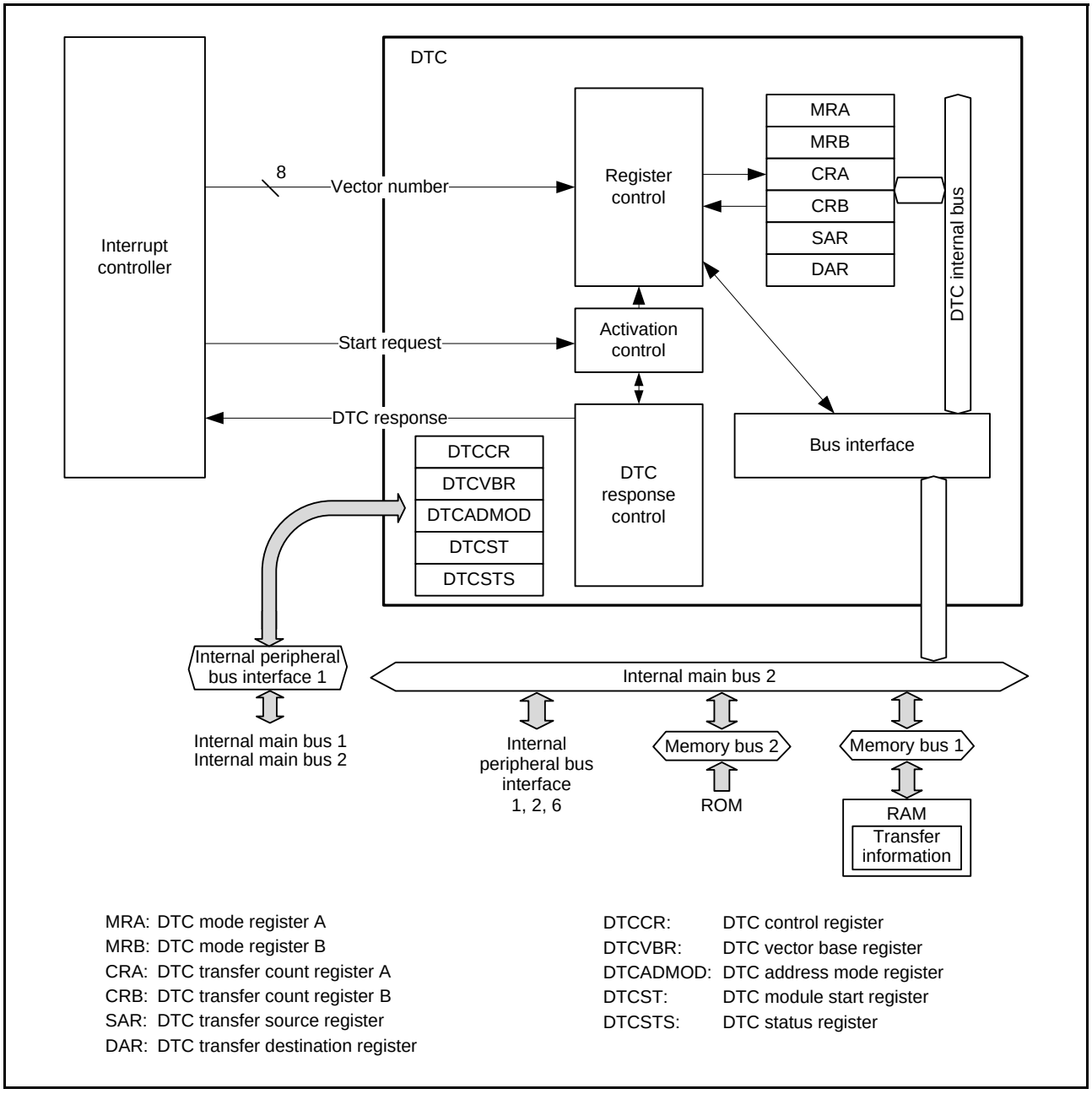


Figure 17.1 Block Diagram of DTC

17.2 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 the DTC internal registers are placed in the RAM area as transfer information data. When an activation request is generated, the DTC reads the transfer information data 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 data.

17.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
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	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 transfer 0 1: Word transfer 1 0: Longword 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 is used to select the operating mode of the DTC.
MRA cannot be accessed directly from the CPU.

SM[1:0] Bits (Transfer Source Address Addressing Mode)

These bits specify the SAR operation after data transfer.

SZ[1:0] Bits (DTC Data Transfer Size)

These bits specify the transfer data size.

MD[1:0] Bits (DTC Transfer Mode Select)

These bits specify the transfer mode of the DTC.

17.2.2 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 SZ[1:0] bits in MRA = 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 in MRA = 00b, –2 when SZ[1:0] bits = 01b, –4 when 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 is used to select the operating mode of the DTC.
MRB cannot be accessed directly from the CPU.

DM[1:0] Bits (Transfer Destination Address Addressing Mode)

These bits specify the DAR operation after data transfer.

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.

DISEL Bit (DTC Interrupt Select)

The DISEL bit specifies when an interrupt request to the CPU is generated: each time DTC data transfer is performed or when specified data transfer is completed.

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, see Table 17.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the startup 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, see section 17.4.6, Chain Transfer.

17.2.3 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.

17.2.4 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

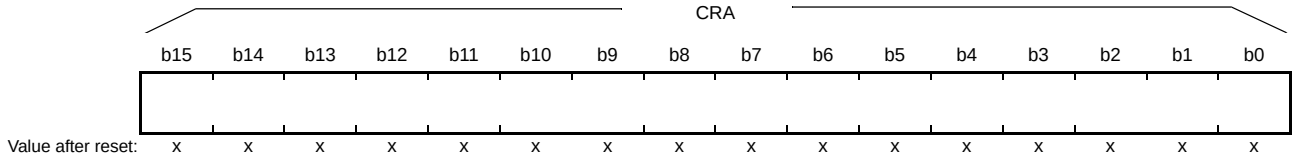
x: Undefined

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.

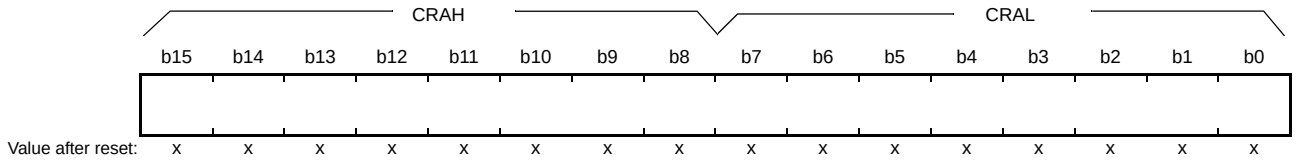
17.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



x: Undefined

Note: • The function depends on transfer mode.

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note: • Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA is used to set the transfer count of the DTC.
The function of this register depends on transfer mode.
CRA cannot be accessed directly from the CPU.

(1) Normal transfer mode (MD[1:0] bits in MRA = 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 (MD[1:0] bits in MRA = 01b)

The CRAH register retains 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 (MD[1:0] bits in MRA = 10b)

The CRAH register retains 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.

17.2.6 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.

17.2.7 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 Data Read Skip Enable	0: Transfer data read is not skipped 1: Transfer data 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

DTCCR is used to control the DTC.

RRS Bit (DTC Transfer Data Read Skip Enable)

The DTC vector number is always compared with the vector number in the previous startup process.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred data. However, when the previous transfer was chain transfer, the transferred data is always read regardless of the value of 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 data is always read regardless of the value of RRS bit.

17.2.8 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/W
b31 to b10	DTC Vector Base Address (Upper 22 bits)	The upper 4 bits (b31 to b28) are 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.

17.2.9 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	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)

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).

17.2.10 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 cleared to 0, transfer requests are no longer accepted.

If this bit is cleared to 0 during data transfer, the accepted transfer request is active until the processing is completed. Before making transition to the module stop state, all-module clock-stop mode, or 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, and software standby mode, refer to section 17.9, Low Power Consumption Function, and section 11, Low Power Consumption.

17.2.11 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 activating 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 activating 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 startup sources and the vector addresses, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

ACT Bit (DTC Active Flag)

This bit 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.

17.3 Sources of Activation

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (where n is the interrupt vector number of the given interrupt) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.
For the correspondence between the DTC startup sources and the vector addresses, see Table 14.3, Interrupt Vector Table. For startup by software, see section 14.2.5, Software Interrupt Activation Register (SWINTR).
Once the DTC has accepted a startup request, it does not accept another startup request until transfer for that single request is completed, regardless of the priority of the requests. When multiple startup requests are generated during DMAC/DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple startup requests are generated while the DTC module start (DTCST) bit in 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 each data transfer (or the last of the consecutive transfers in the case of a chained transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is cleared 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 startup source is cleared to 0 at the start of data transfer.

17.3.1 Allocating Transfer Data and DTC Vector Table

The DTC reads the start address of the transfer data corresponding to each startup source from the vector table and reads the transfer data 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 data is allocated in the RAM area. In the RAM area, the start address of the transfer data (n) with vector number n should be 4n added to the base address in the vector table.
Transfer data can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the SHORT bit in DTCADMOD to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 17.2 shows the relationship between the DTC vector table and transfer data.
Figure 17.3 shows the allocation of transfer data in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, see section 17.10.2, Allocating Transfer Data.

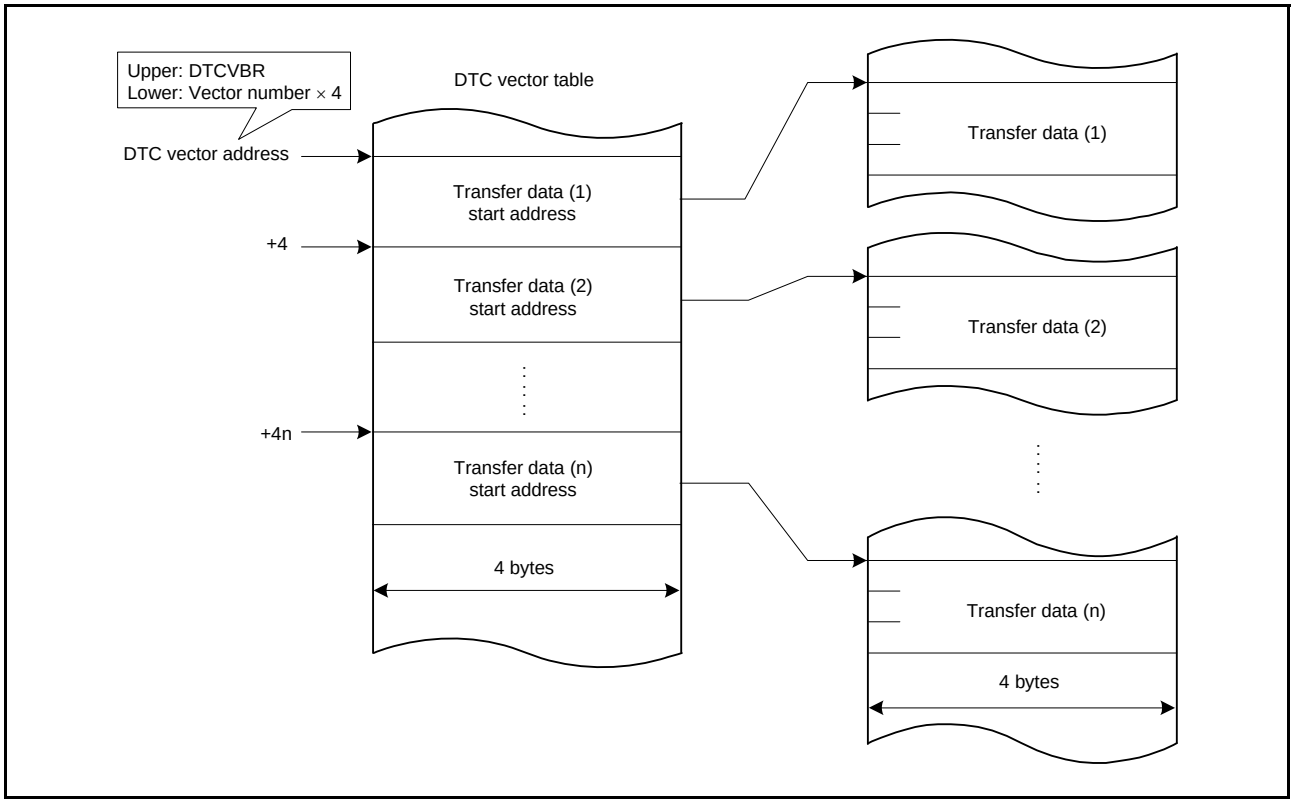


Figure 17.2 DTC Vector Table and Transfer Data

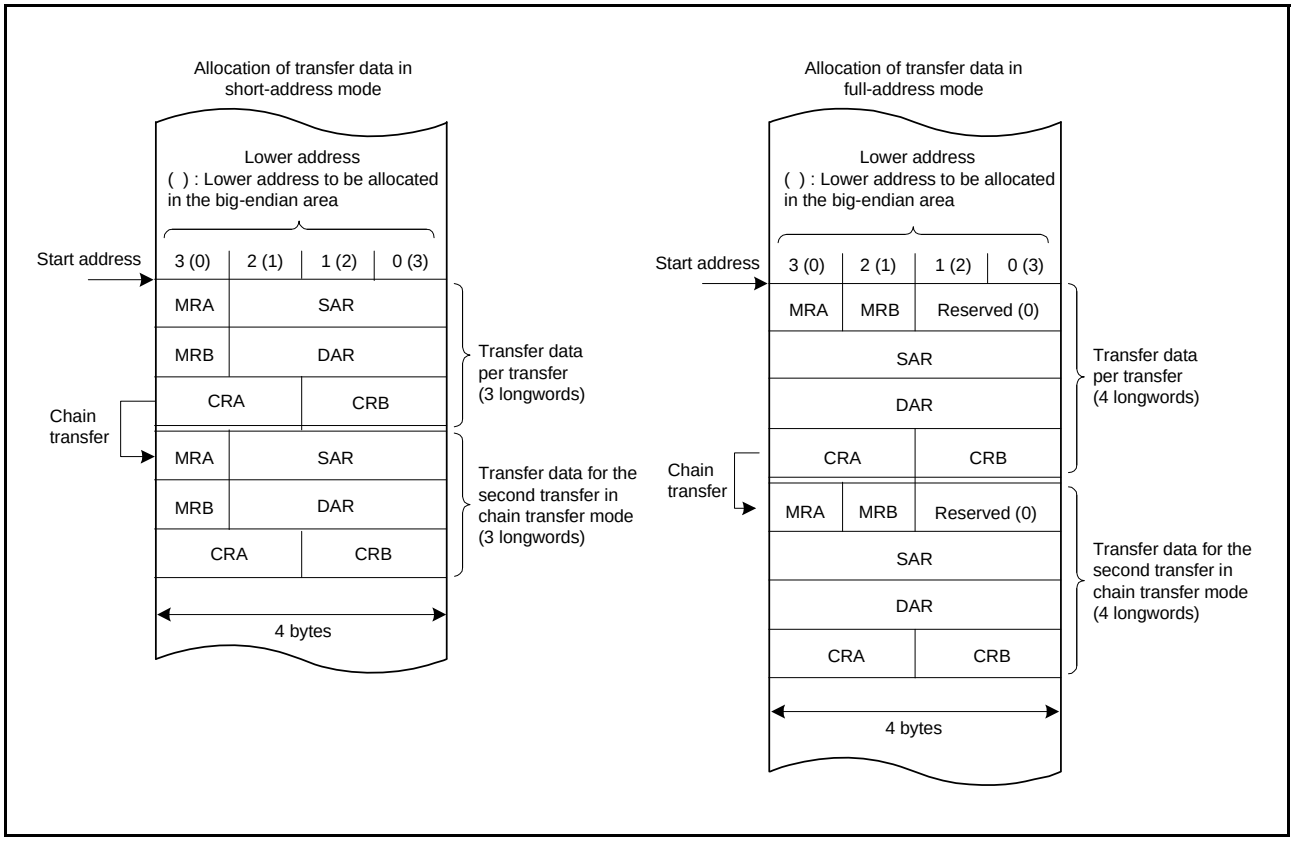


Figure 17.3 Allocation of Transfer Data in the RAM Area

17.4 Operation

The DTC transfers data in accordance with the transfer data. Storage of the transfer data in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Then the DTC reads transfer data from the transfer data store address pointed by the DTC vector, transfers data, and then writes back the transfer data after the data transfer. Storing transfer data 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 SAR and a transfer destination address in DAR. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 17.2 lists transfer modes of the DTC.

Table 17.2 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on a Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/word/longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/word/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/words/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 CHNE bit in MRB to 1 allows multiple transfers (chain transfer) on a single startup source. Setting the CHNS bit in MRB also enables chain transfer when specified data transfer is completed.

Figure 17.4 shows the operation flowchart of the DTC. Table 17.3 lists chain transfer conditions.

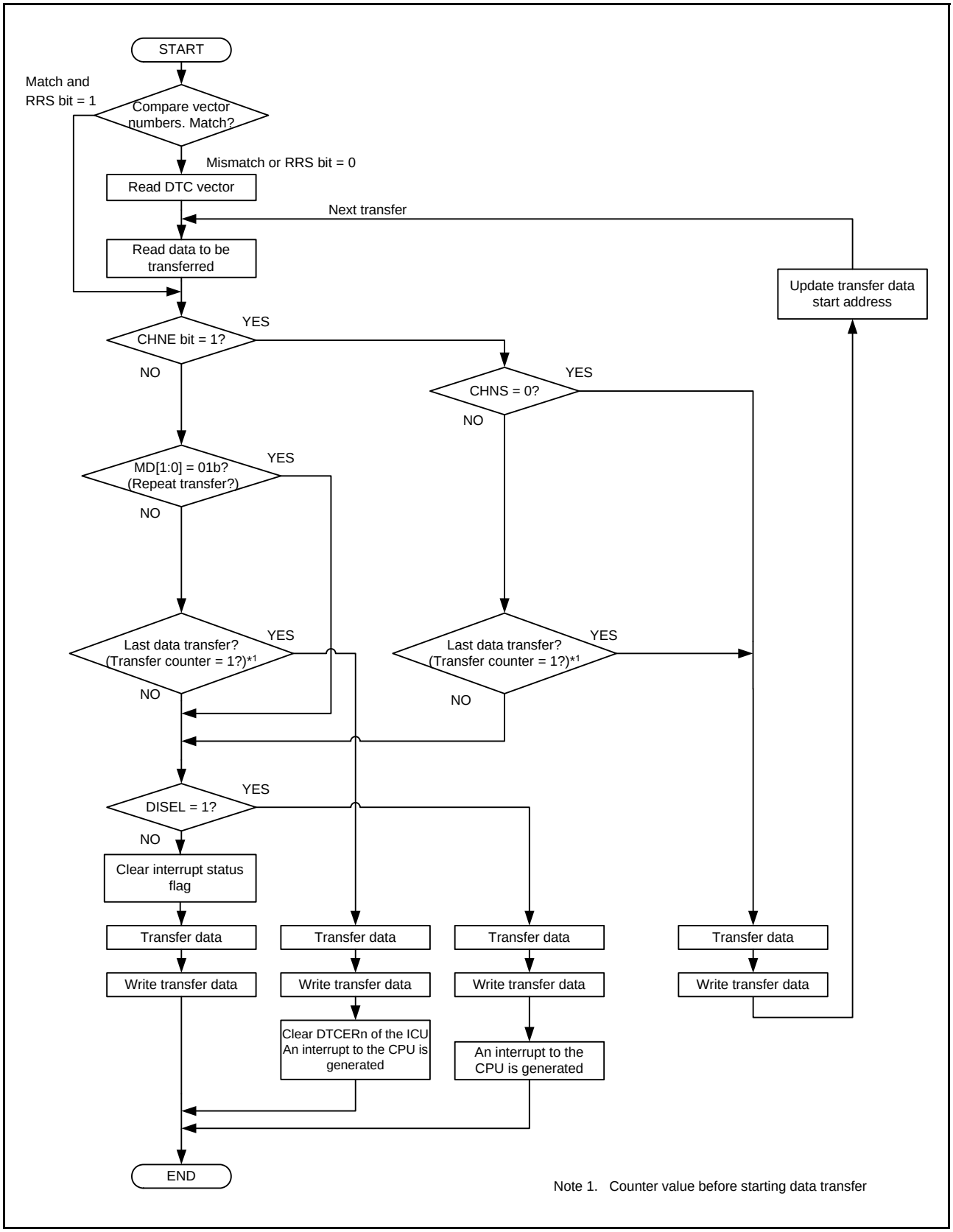


Figure 17.4 Operation Flowchart of the DTC

Table 17.3 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.

17.4.1 Transfer Data Read Skip Function

Vector address read and transfer data read can be skipped by the setting of the RRS bit in DTCCR.
When a DTC startup request is generated, the current DTC vector number is always compared with the DTC vector number in the previous startup 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 data. However, when the previous transfer was chain transfer, the vector address and transfer data are always 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 data is always read regardless of the value of RRS bit. Figure 17.13 shows an example of transfer data read skip.
To update the vector table and transfer data, set the RRS bit to 0, update the vector table and transfer data, and then set the RRS bit to 1. When the RRS bit is set to 0, the retained vector number is discarded and the vector table and transfer data that are updated in the following startup process are read.

17.4.2 Transfer Data Write-Back Skip Function

When the SM[1:0] bits in MRA or the DM[1:0] bits in MRB are set to “address fixed”, a part of transfer data is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 17.4 lists transfer data write-back skip conditions and applicable registers.
The CRA and CRB registers are always 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 always skipped.

Table 17.4 Transfer Data Write-Back Skip Conditions and Applicable Registers

SM[1:0] Bits in MRA		DM[1:0] Bits in MRB		SAR Register	DAR Register
b3	b2	b3	b2		
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		

17.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup 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 17.5 lists register functions in normal transfer mode, and Figure 17.5 shows the memory map of normal transfer mode.

Table 17.5 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
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.

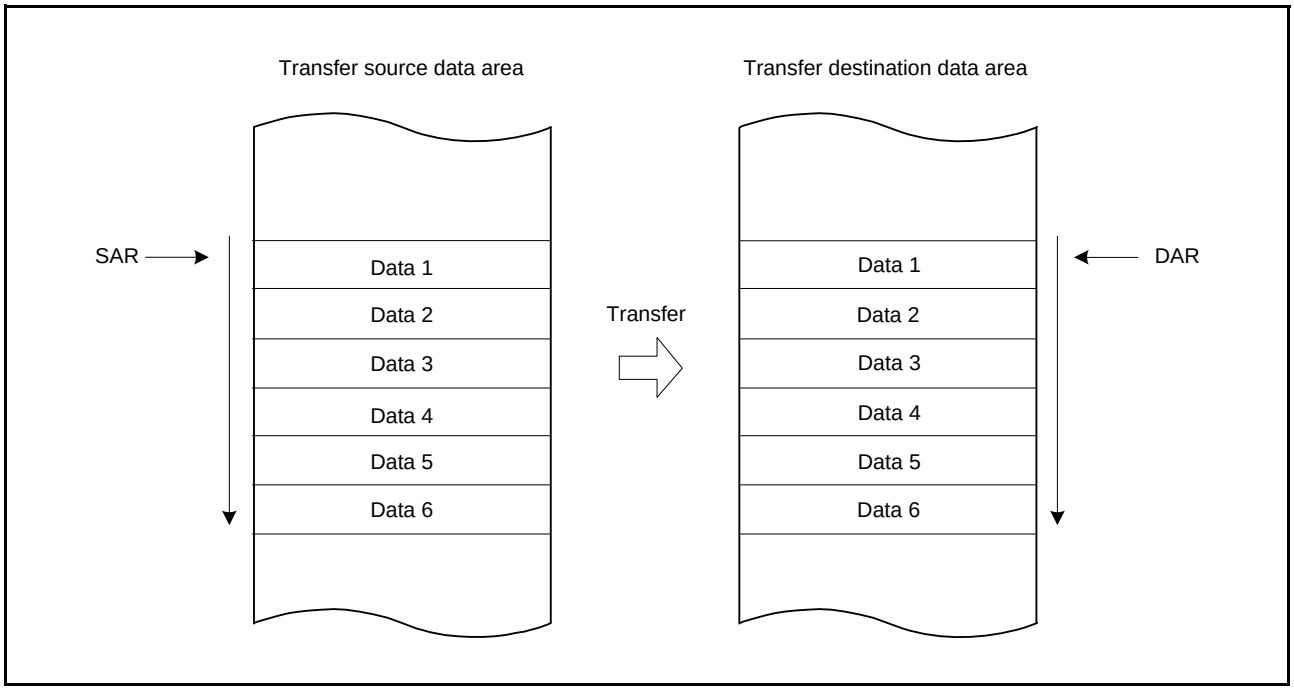


Figure 17.5 Memory Map of Normal Transfer Mode

17.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source.
Specify either transfer source or transfer destination for the repeat area by the DTS bit in MRB. 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 CRAH. Thus the transfer counter does not become 00h, which inhibits generation of interrupt request to the CPU when the DISEL bit in MRB is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
Table 17.6 lists the register functions in repeat transfer mode, and Figure 17.6 shows the memory map of repeat transfer mode.

Table 17.6 Register Functions in Repeat Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data	
		When CRAL is not 1	**When CRAL is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	(When the DTS bit in MRB is 0) Increment/decrement/fixed*1 (When the DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	(When the DTS bit in MRB is 0) DAR register initial value (When the DTS bit in MRB 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.

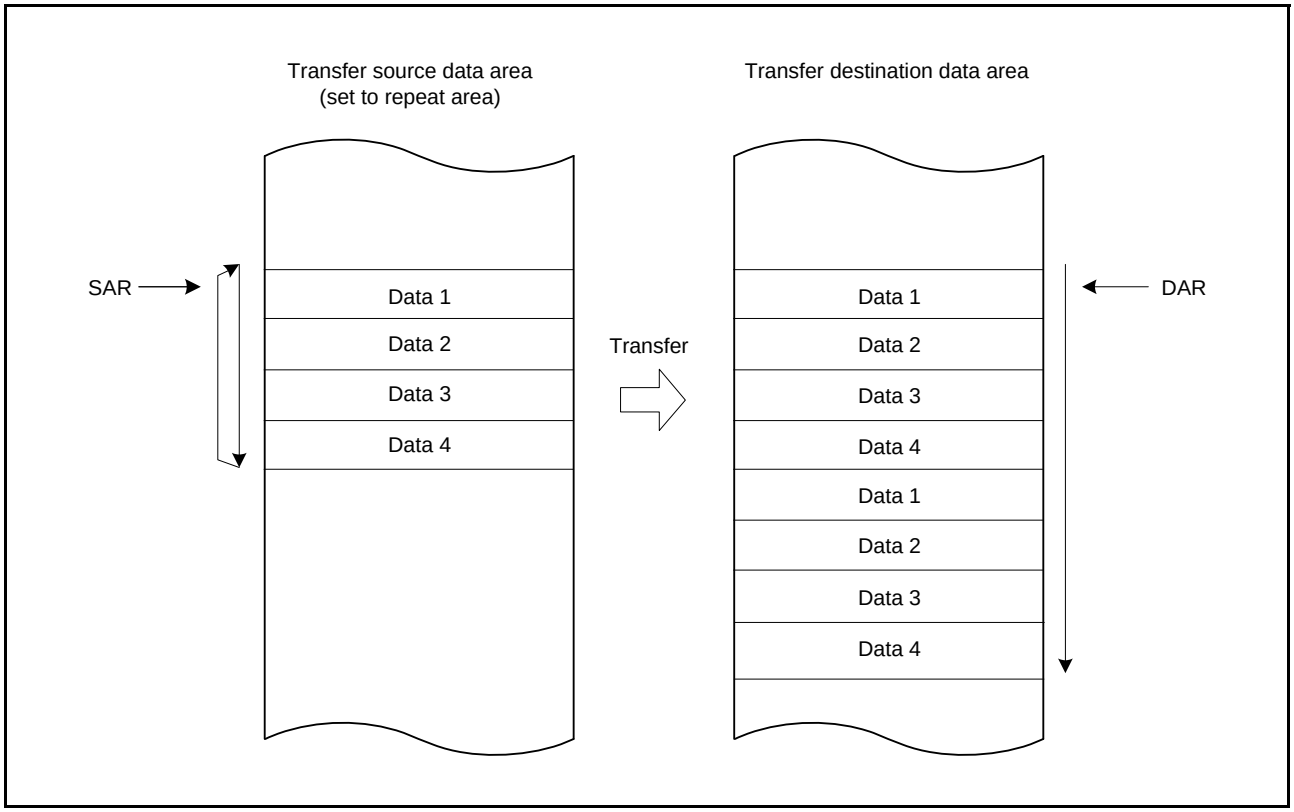


Figure 17.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

17.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single startup source.
Specify either transfer source or transfer destination for the block area by the DTS bit in MRB. The block size can be set to 1 to 256 bytes (or 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 (SAR when the DTS bit = 1 or DAR 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 17.7 lists register functions in block transfer mode, and Figure 17.7 shows the memory map of block transfer mode.

Table 17.7 Register Functions in Block Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	(When DTS bit in MRB is 0) Increment/decrement/fixed*1 (When DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	(When DTS bit in MRB is 0) DAR register initial value (When DTS bit in MRB 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.

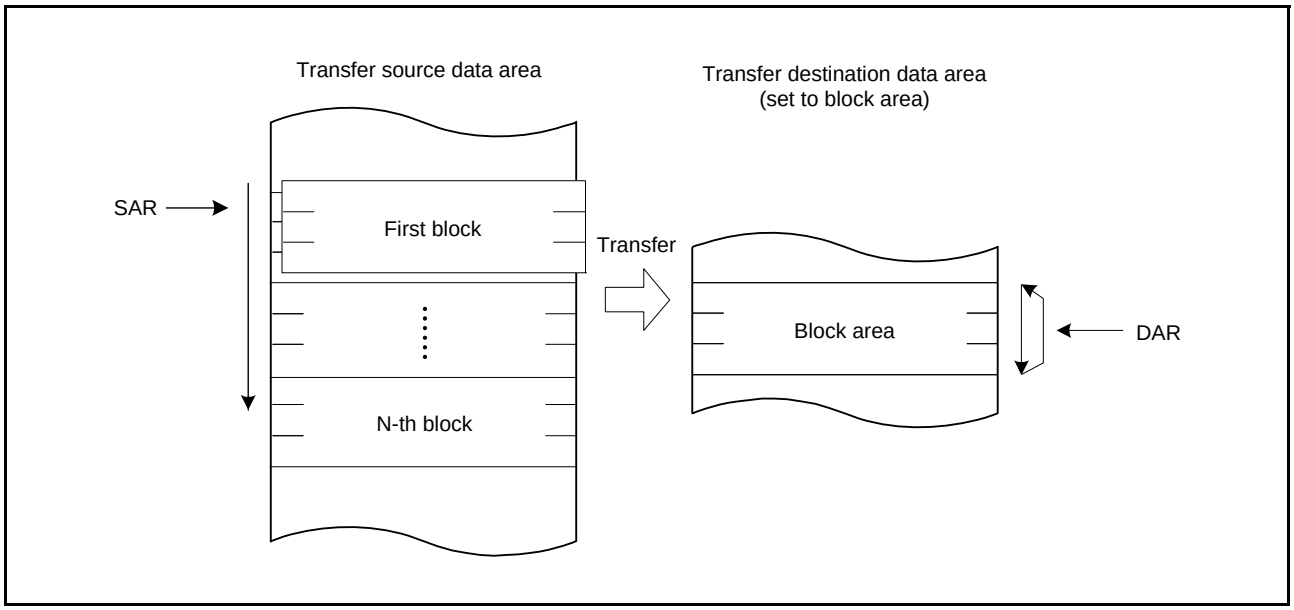


Figure 17.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

17.4.6 Chain Transfer

Setting the CHNE bit in MRB to 1 allows chain transfer to be performed continuously on a single startup source.
If the CHNE and CHNS bits in MRB 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 DISEL bit in MRB 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 that has started up the transfer.
The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer.
Figure 17.8 shows chain transfer operation.

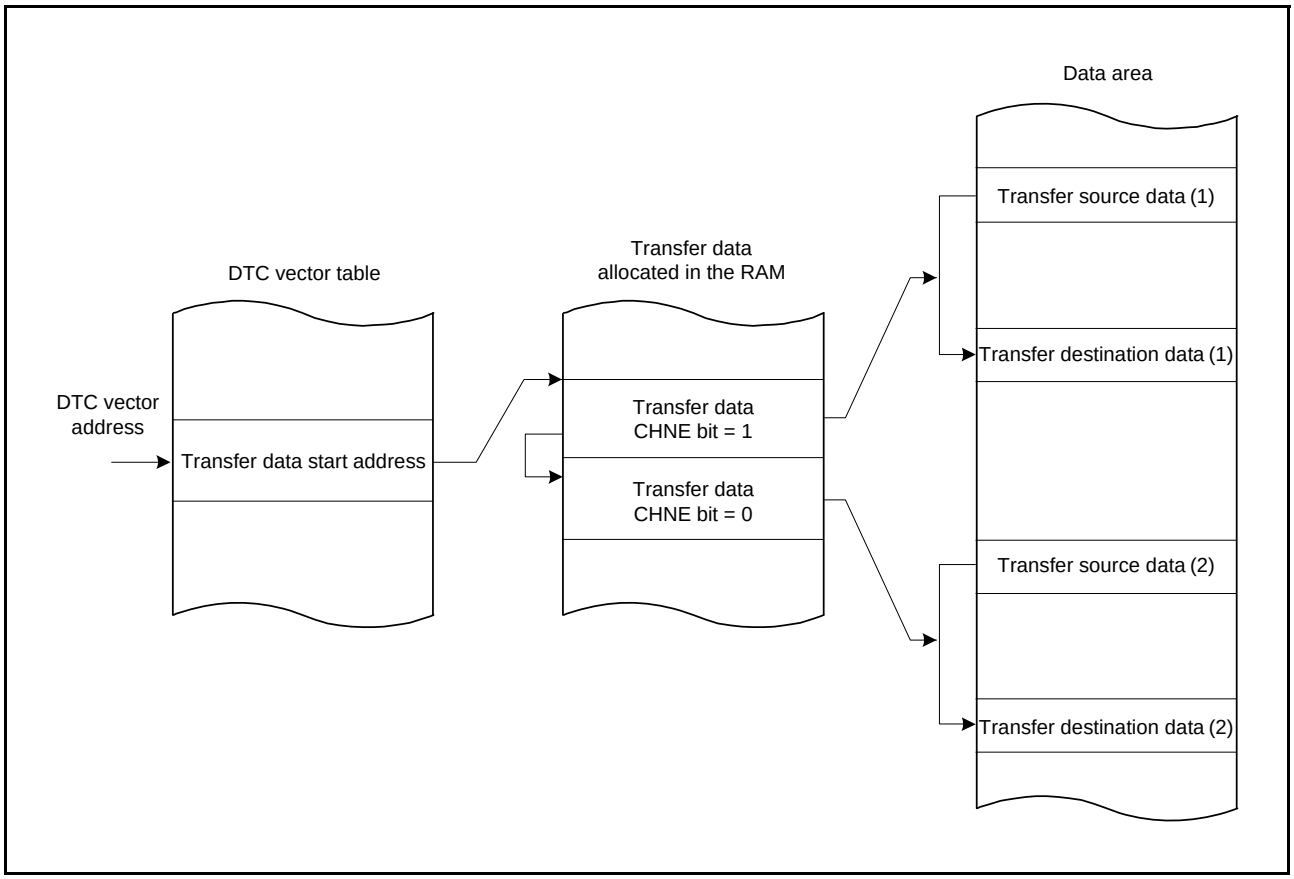


Figure 17.8 Chain Transfer Operation

Writing 1 to the CHNE and CHNS bits in MRB 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 17.3, Chain Transfer Conditions.

17.4.7 Operation Timing

Figure 17.9 to Figure 17.13 show examples of DTC operation timing.

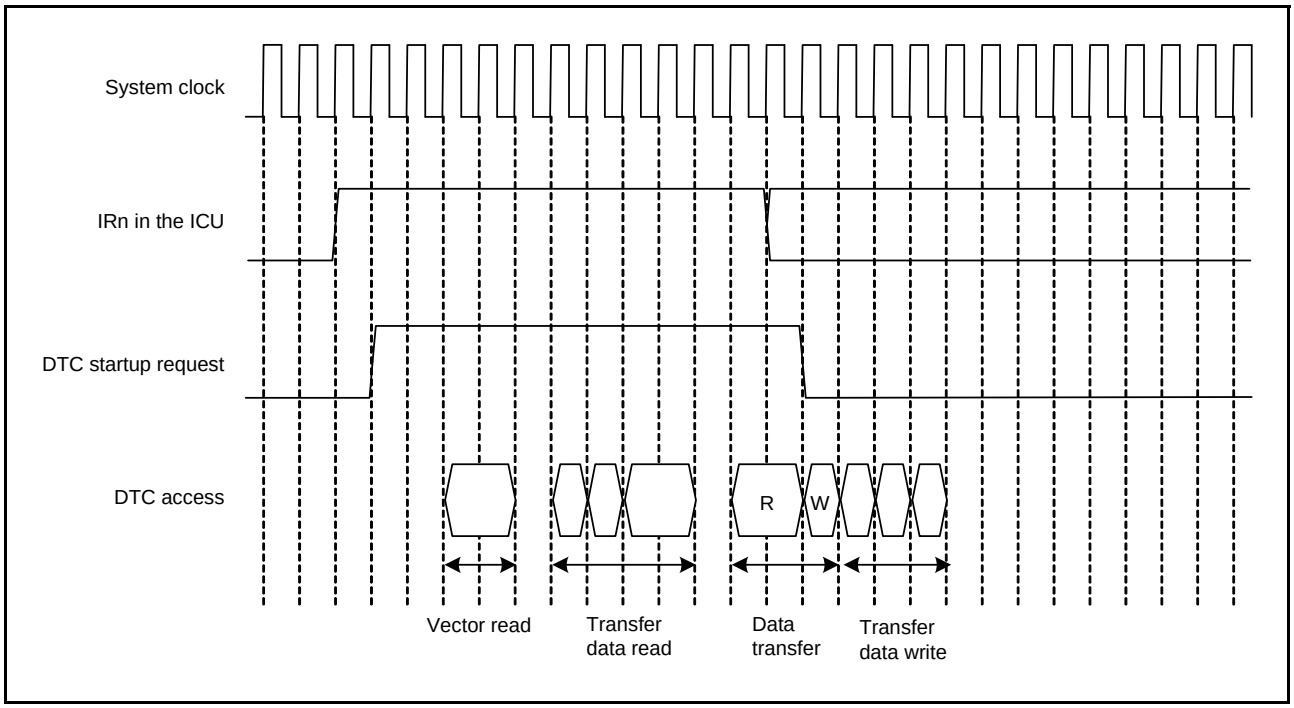


Figure 17.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

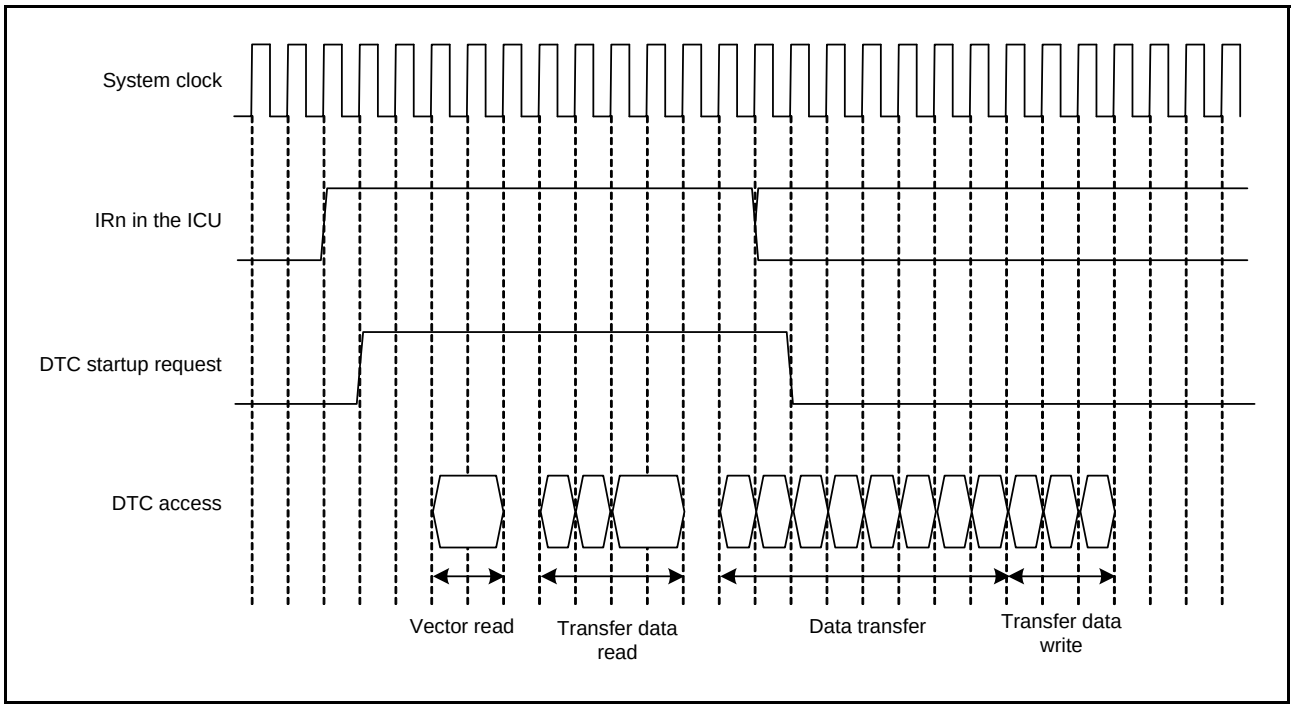


Figure 17.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

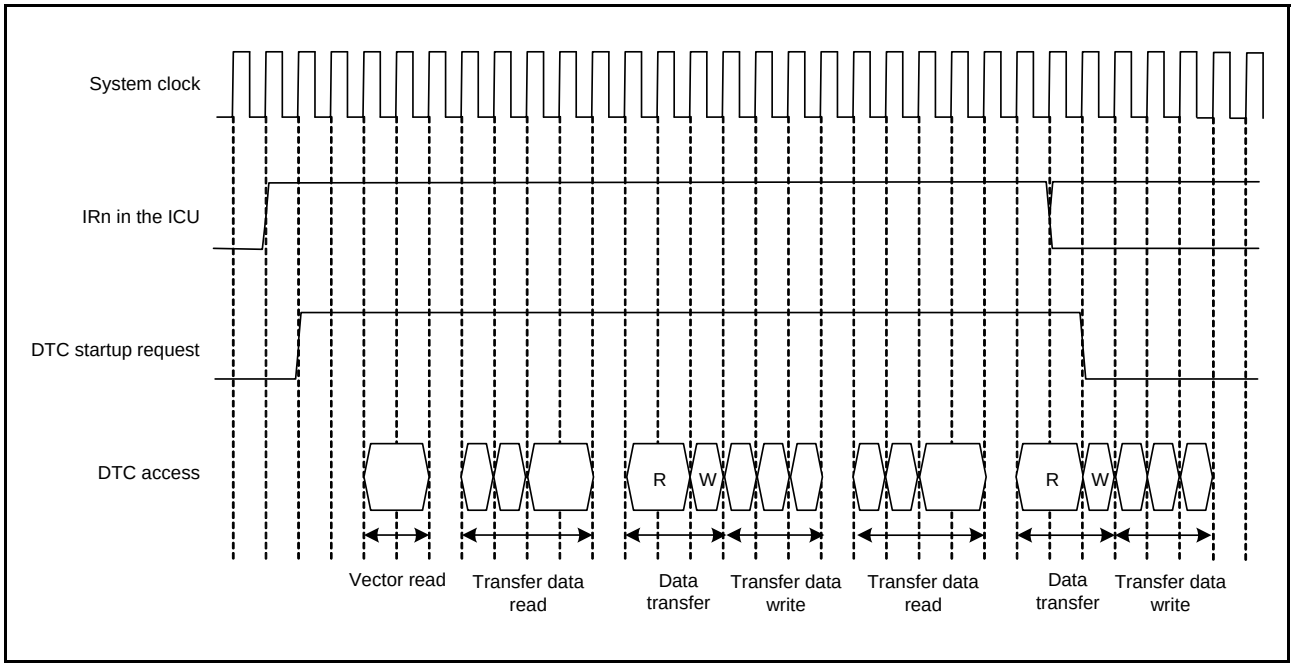


Figure 17.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

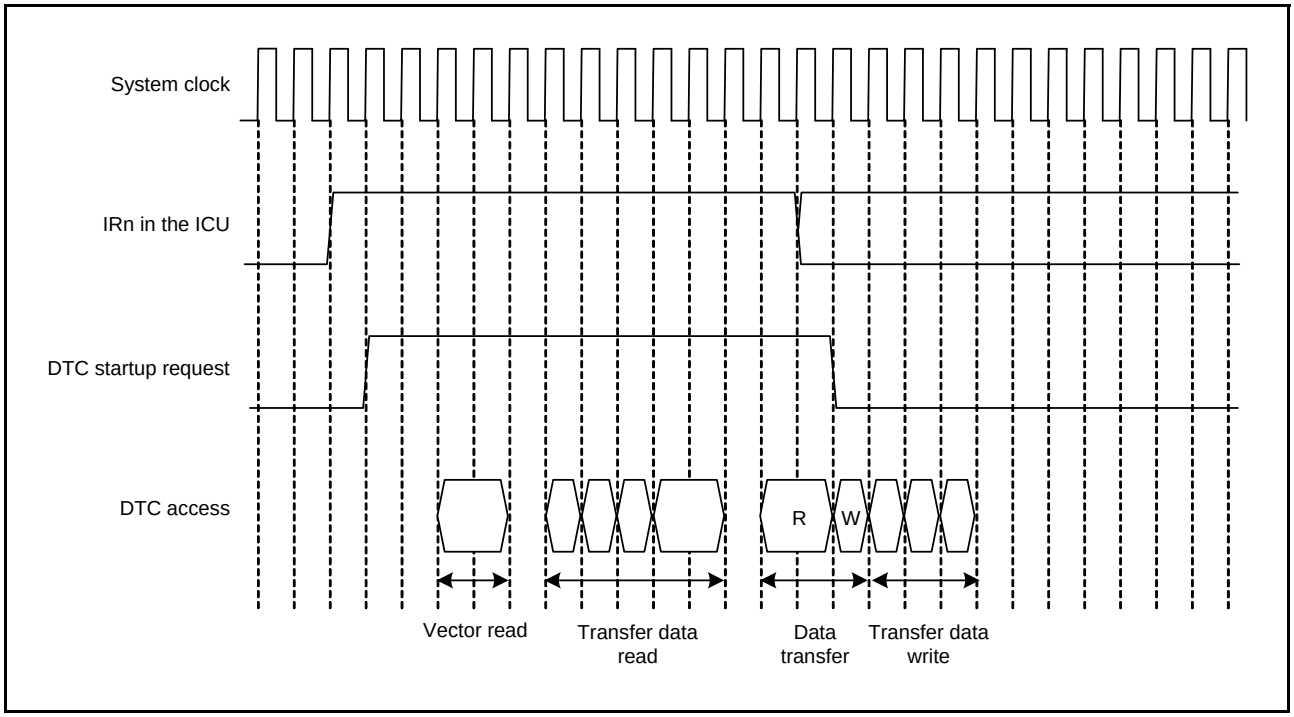
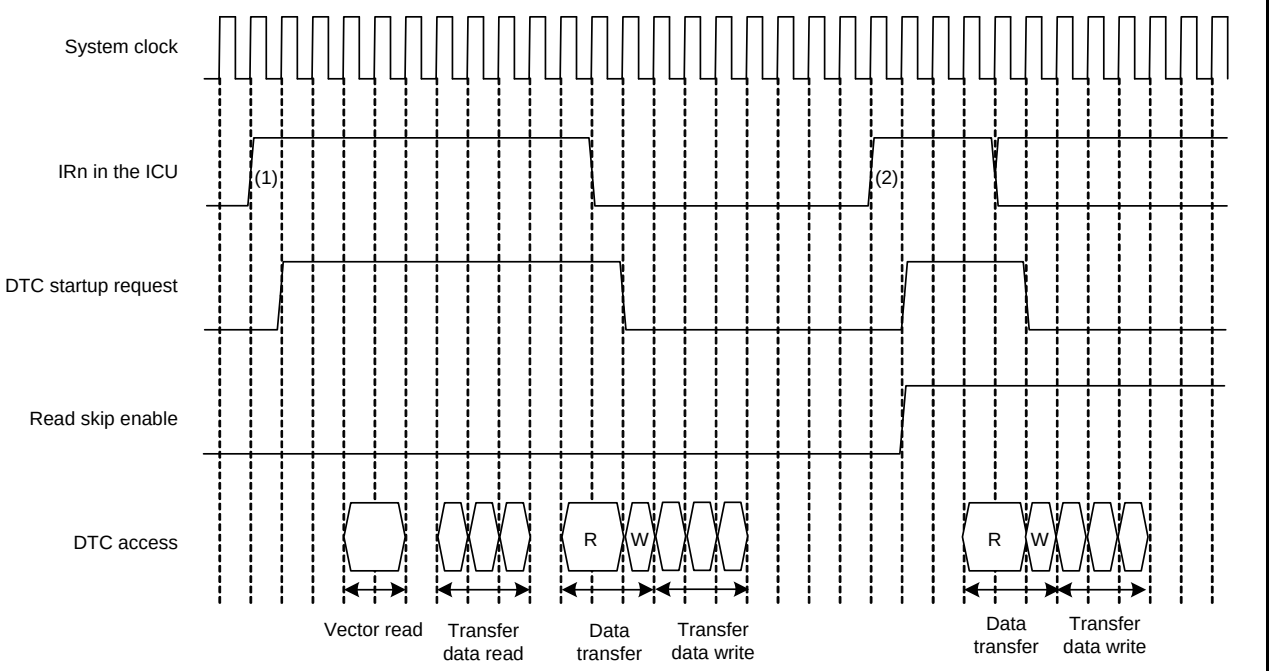


Figure 17.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: • When startup sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer data read for request (2) is skipped.

Figure 17.13 Example of Operation when Transfer Information Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

17.4.8 Execution Cycles of the DTC

Table 17.8 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 17.4.7, Operation Timing.

Table 17.8 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Data Read			Transfer Data 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 data 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 data storage destination
Ci: Cycles for access to transfer data 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 Data 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, see section 35, RAM, section 36, ROM (Flash Memory for Code Storage), and section 5, I/O Registers.

17.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer data read and transfer data write. While transfer data is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, see section 15, Buses.

17.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 17.14 shows the procedure to set the DTC.

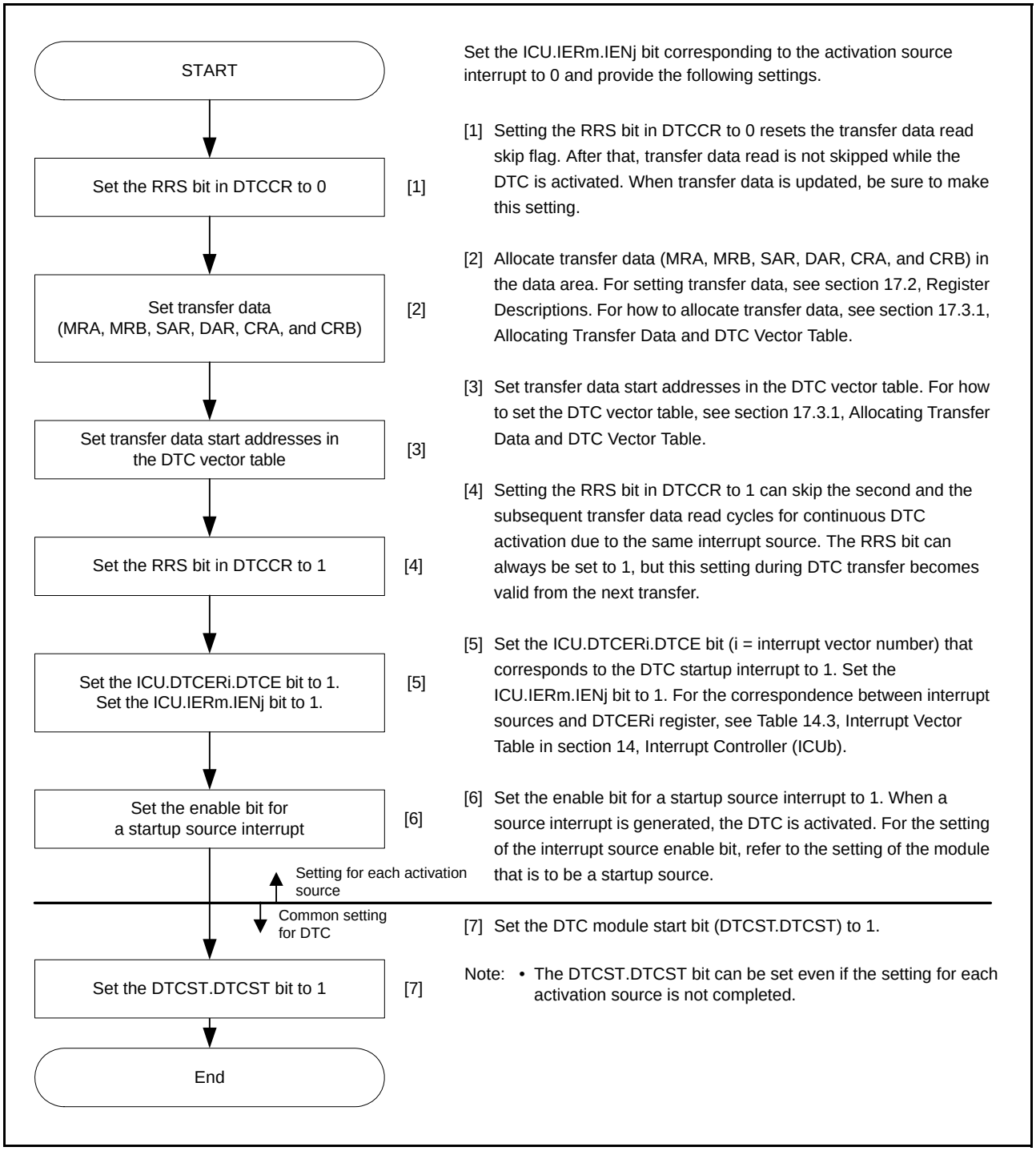


Figure 17.14 Procedure to Set the DTC

17.6 Examples of DTC Usage

17.6.1 Normal Transfer

As an example of DTC usage, its employment in the transfer of 128 bytes of data by an SCI is described below.

(1) Transfer Data Set

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 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 address of the RDR register in the SCIm (m = 0 to 12) 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

The address where the transfer-control information for use with the RXI starts is set in the vector table for the DTC.

(3) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERi.DTCE bit to 1 and the ICU.IERi.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Set

Enable the receive end interrupt (RXI) by setting the SCR.RIE bit in the SCIm 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 SCIm 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. Processing for completion is performed in the processing routine for this interrupt.

17.6.2 Chain Transfer when Counter = 0

The second data transfer is performed only when the counter = 0. Repeat transfer of a transfer count of 256 or more is enabled by the re-setting for the first data transfer.
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 17.15 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, CRA = 0000h (65,536 times), CHNE bit = 1 (chain transfer enabled) in MRB, CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0) in MRB, and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB.
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 200000h to 21FFFFh, 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 DAR in the first transfer data area for the transfer destination. At this time, set CHNE bit = 0 (chain transfer disabled) in MRB and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB. When setting the input buffer mentioned above to 200000h to 21FFFFh, 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.

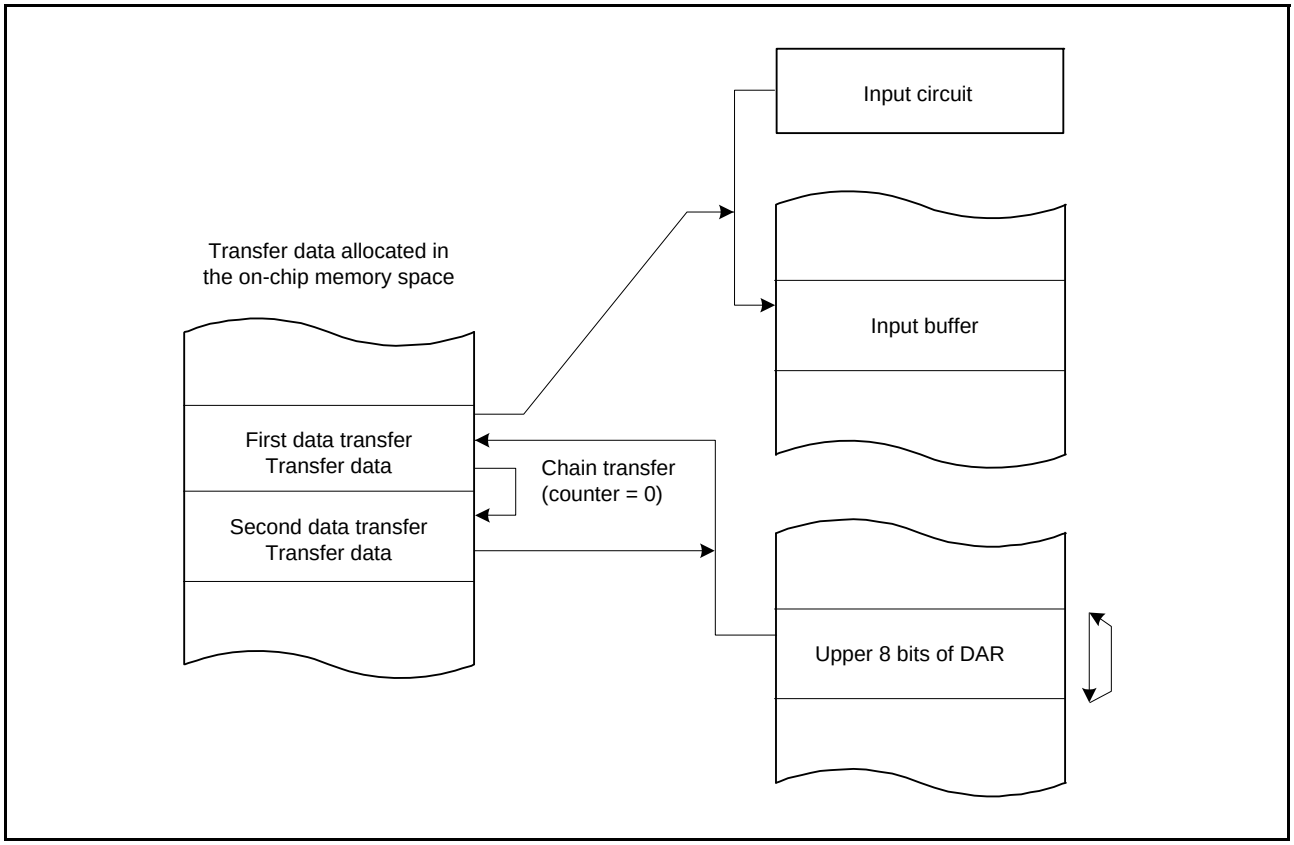


Figure 17.15 Chain Transfer when Counter = 0

17.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the DISEL bit in MRB 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 startup 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.

17.8 Event Link Function

The DTC outputs the event link request after transfer for that single activating source is completed.

17.9 Low Power Consumption Function

Before transition to the module stop state, all-module clock stop mode, or software standby mode, clear the DTCST bit in DTCST to 0 (the DTC 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 DTC. If DTC transfer is in progress at the time 1 is written to the MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPA28 bit is 1, accessing the DTC registers are prohibited.
Writing 0 to the MSTPA28 bit releases the DTC 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 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 Mode

Make settings in accord with the procedure under section 11.6.3.1, Transition to 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 follows the completion of DTC transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DTC transfer after returning from low power consumption mode, set the DTCST bit in DTCST 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 DTC 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.

17.10 Usage Notes

17.10.1 Transfer Data Start Address

Be sure to set multiples of 4 for the transfer data start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

17.10.2 Allocating Transfer Data

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 17.16.
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.

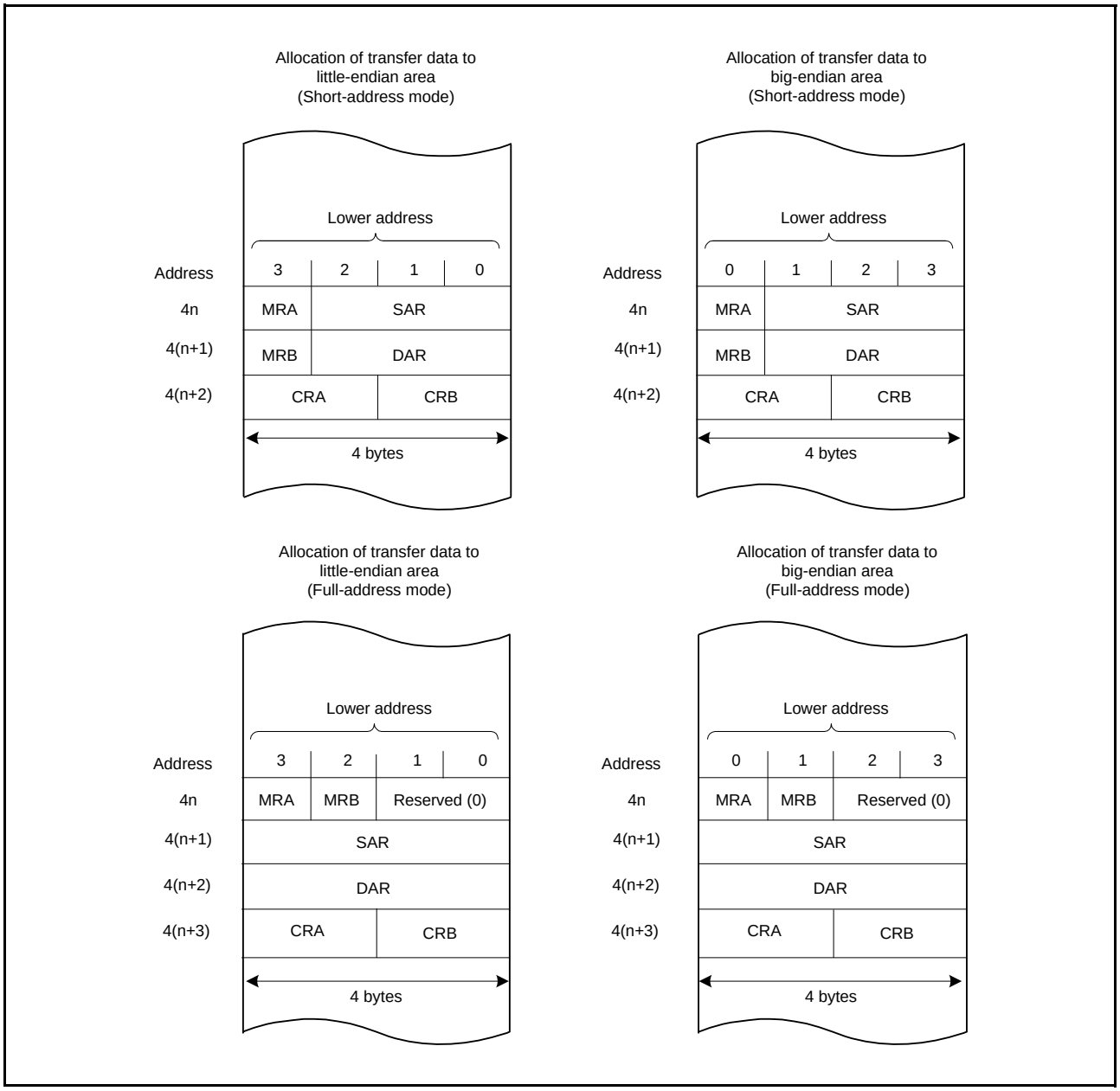


Figure 17.16 Allocation of Transfer Data

17.10.3 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 1 (DTC transfer enable). For details on the ICU.DTCERn and ICU.DMRSRn registers (n = number of DMAC channel), refer to section 14, Interrupt Controller (ICUb).

18. Event Link Controller (ELC)

18.1 Overview

The event link controller (ELC) connects (links) the events generated by various peripheral modules to different modules. Event linking allows direct cooperation between the modules without CPU intervention.

Table 18.1 lists the specifications of the ELC, and Figure 18.1 shows a block diagram of the ELC.

Table 18.1 ELC Specifications

Item	Description
Event link	• 46 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 ports B. Single-port*1: An event link can be set for a specified 1-bit in a port. Port group*1: An event link can be set for a group of specified bits within an 8-bit port.
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. In products with 64-pin packages, when ports PC0 and PC1 are selected in port switching register A (PSRA), input to ports PB6 and PB7 and the output port event function of the ELC cannot be used.
In products with 48-pin packages, when ports PC0 to PC3 are selected in port switching register B (PSRB), input to ports PB0, PB1, PB3, and PB5 and the output port event function of the ELC cannot be used.

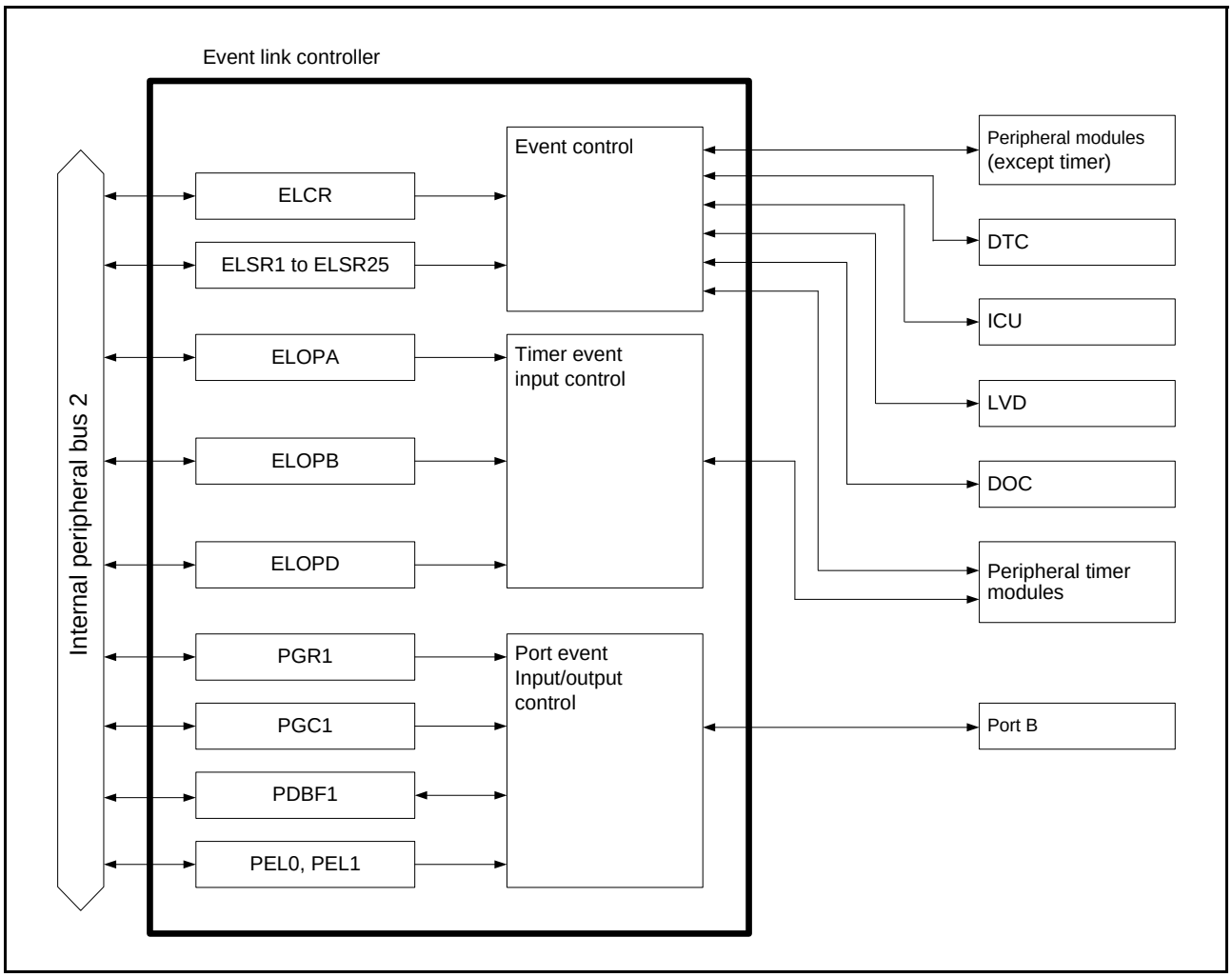


Figure 18.1 Block Diagram of Event Link Controller

18.2 Register Descriptions

18.2.1 Event Link Control Register (ELCR)

Address(es): 0008 B100h

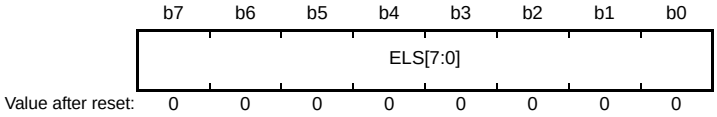
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

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: Linkage of all the event is disabled. 1: Linkage of all the event is enabled.	R/W

ELCR controls the operation of the event link controller (ELC).

18.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 10, 12, 15, 18, 20, 22, 24, 25)

Address(es): ELSR1: 0008 B102h, ELSR2: 0008 B103h, ELSR3: 0008 B104h, ELSR4: 0008 B105h, ELSR10: 0008 B10Bh, ELSR12: 0008 B10Dh, ELSR15: 0008 B110h, ELSR18: 0008 B113h, ELSR20: 0008 B115h, ELSR22: 0008 B117h, ELSR24: 0008 B119h, ELSR25: 0008 B11Ah



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	b7 b0 00000000: Event link function is disabled. 00000001 to 01101010: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

ELSRn specifies an event signal to be linked for each peripheral module. Table 18.2 shows the correspondence between ELSRn registers and the peripheral modules. Table 18.3 shows the correspondence between the event signal names set in ELSRn and the signal numbers.

Table 18.2 Correspondence between ELSRn Registers and Peripheral Functions

Register Name	Peripheral Function (Module)
ELSR1	MTU1
ELSR2	MTU2
ELSR3	MTU3
ELSR4	MTU4
ELSR10	TMR0
ELSR12	TMR2
ELSR15	12-bit A/D converter
ELSR18	Interrupt 1
ELSR20	Output port group 1
ELSR22	Input port group 1
ELSR24	Single-port 0*1
ELSR25	Single-port 1*1

Note 1. Do not set the DOC data operation condition met signal (ELS[7:0] bits = 6Ah) in the ELSR24 or ELSR25 register.

Table 18.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (1 / 2)

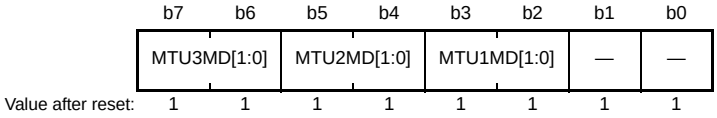
ELS[7:0] Bit Value	Name of Event Signal Set in ELSR
00001000 (08h)	MTU1 compare match 1A signal
00001001 (09h)	MTU1 compare match 1B signal
00001010 (0Ah)	MTU1 overflow signal
00001011 (0Bh)	MTU1 underflow signal
00001100 (0Ch)	MTU2 compare match 2A signal
00001101 (0Dh)	MTU2 compare match 2B signal
00001110 (0Eh)	MTU2 overflow signal
00001111 (0Fh)	MTU2 underflow signal
00010000 (10h)	MTU3 compare match 3A signal
00010001 (11h)	MTU3 compare match 3B signal
00010010 (12h)	MTU3 compare match 3C signal
00010011 (13h)	MTU3 compare match 3D signal
00010100 (14h)	MTU3 overflow signal
00010101 (15h)	MTU4 compare match 4A signal
00010110 (16h)	MTU4 compare match 4B signal
00010111 (17h)	MTU4 compare match 4C signal
00011000 (18h)	MTU4 compare match 4D signal
00011001 (19h)	MTU4 overflow signal
00011010 (1Ah)	MTU4 underflow signal
00100010 (22h)	TMR0 compare match A0 signal
00100011 (23h)	TMR0 compare match B0 signal
00100100 (24h)	TMR0 overflow signal
00101000 (28h)	TMR2 compare match A2 signal
00101001 (29h)	TMR2 compare match B2 signal
00101010 (2Ah)	TMR2 overflow signal
00111010 (3Ah)	SCI5 error (receive error or error signal detection) signal
00111011 (3Bh)	SCI5 receive data full signal
00111100 (3Ch)	SCI5 transmit data empty signal
00111101 (3Dh)	SCI5 transmit end signal
01001110 (4Eh)	RIIC0 communication error or event generation signal
01001111 (4Fh)	RIIC0 receive data full signal
01010000 (50h)	RIIC0 transmit data empty signal
01010001 (51h)	RIIC0 transmit end signal
01010010 (52h)	RSPI0 error (mode fault, overrun, or parity error) signal
01010011 (53h)	RSPI0 idle signal
01010100 (54h)	RSPI0 receive data full signal
01010101 (55h)	RSPI0 transmit data empty signal
01010110 (56h)	RSPI0 transmit end signal (except during clock synchronous operation in slave mode)
01011000 (58h)	A/D conversion end signal of 12-bit A/D converter
01011011 (5Bh)	LVD1 voltage detection signal
01100001 (61h)	DTC transfer end signal
01100011 (63h)	Input edge detection signal of input port group 1
01100101 (65h)	Input edge detection signal of single input port 0
01100110 (66h)	Input edge detection signal of single input port 1
01101001 (69h)	Software event signal

Table 18.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (2 / 2)

ELS[7:0] Bit Value	Name of Event Signal Set in ELSR
01101010 (6Ah)	DOC data operation condition met signal
Settings other than above are prohibited.	

18.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): 0008 B11Fh



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 in the MTU when an event is input. All events must be disabled when the ELC function is not to be used.

18.2.4 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 in the MTU when an event is input. All events must be disabled when the ELC function is not to be used.

18.2.5 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 in the TMR when an event is input. All events must be disabled when the ELC function is not to be used.

18.2.6 Port Group Setting Register 1 (PGR1)

Address(es): PGR1: 0008 B123h

	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

PGR1 specifies a group for I/O port bits. PGR1 specifies each port bit in the same 8-bit I/O port as the member of a group. One to eight port bits can be specified as the members of the same group as required. The correspondence between PGR1 and ports is shown in Table 18.4.

18.2.7 Port Group Control Register 1 (PGC1)

Address(es): PGC1: 0008 B125h

	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 PDBF register is disabled. 1: Overwriting PDBF 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, PGC1 specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, PGC1 enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).
The correspondence between PGC1 and ports is shown in Table 18.4.

18.2.8 Port Buffer Register 1 (PDBF1)

Address(es): PDBF1: 0008 B127h

	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 by the CPU is invalid. For details, see section 18.3, 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

PDBF1 is an 8-bit readable/writable register used in combination with PGR1. For PDBF1 operations, see section 18.3, Operation. Table 18.4 shows registers related to port groups and corresponding port numbers.

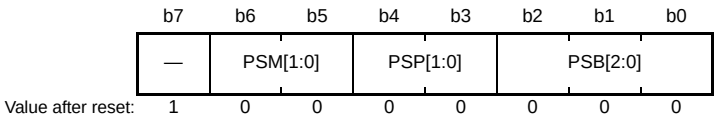
Table 18.4 Registers Related to Port Groups and Corresponding Port Numbers

Port Group Setting Register (PGR)	Port Group Control Register (PGC)	Port Buffer Register (PDBF)	Port Number
PGR1 register	PGC1 register	PDBF1 register	Port B

Note: • Since pins PE0 to PE5 are not available, bits b0 to b5 of the PGR2 and PDBF2 registers cannot be set.

18.2.9 Event Link Port Setting Register n (PELn) (n = 0, 1)

Address(es): PEL0: 0008 B129h, PEL1: 0008 B12Ah



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	A bit number in an 8-bit port is specified.	R/W
b4, b3	PSP[1:0]	Port Number Specification	b4 b3 0 0: Setting is invalid. 0 1: Port B (corresponding to PGR1) 1 X: Do not set this value.	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 1-bit port (hereinafter referred to as a 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 the RX220 Group, a total of two bits in either port B (8-bit port) can be specified as single-ports.

18.2.10 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 Write Disable	0: Write to ELSEGR is enabled. 1: Write to ELSEGR is disabled.	W

The MOV instruction must always 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 Write Disable)

ELSEGR can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

18.3 Operation

18.3.1 Relation between Interrupt Processing and Event Linking

The modules incorporated in the RX220 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 ELC uses interrupt requests (hereinafter referred to as events) generated in modules as event signals that directly activate other modules. This means that the event signal can be used whether or not the interrupt signal is enabled. Figure 18.2 shows the relation between the interrupt processing and ELC.

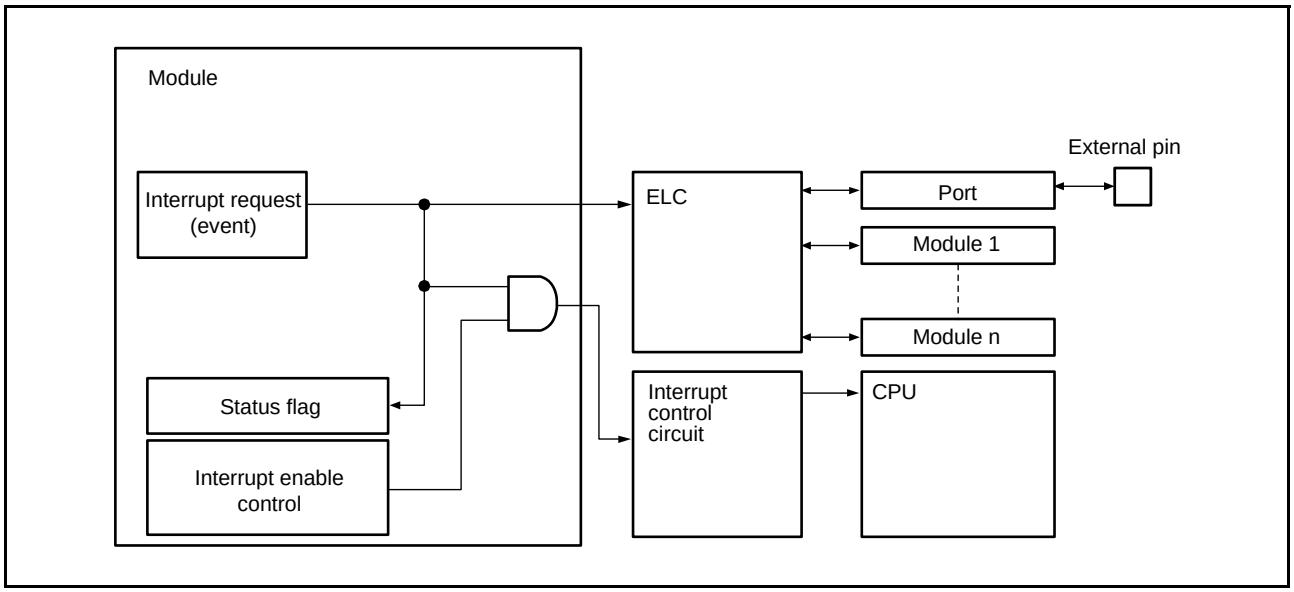


Figure 18.2 Relation between Interrupt Processing and ELC

18.3.2 Event Linkage

When an event has been set as a trigger in the event link setting registers (ELSRn) and then occurs, that event is linked with the corresponding module (the module is activated). Only one type of event can be connected with one module. When a module is to be activated by the event link controller, the operation of the module must be set up in advance. Table 18.5 lists the operations of modules when an event is input.

Table 18.5 Operations of Modules when Event is Input

Module	Operations when Event is Input		
MTU TMR	Each timer operates differently depending on the ELOPA, ELOPB, and ELOPD registers as below. • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (TMR). • Performs input-capture operation when an event is input (MTU).		
A/D converter	Starts A/D conversion when an event signal is input.		
Output ports	The value of PODR (output port register) changes when an event signal is input. (The value of the signal to be output from the relevant external pin changes.)	Port group	The port group operates differently depending on the settings as below. • Changes the PODR value to the specified value. • Transfers the PDBF1 value to PODR. • Rotates out the bit value.
		Single-port	Changes the PODR value to the specified value.
Input ports	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 PDBF1.
		Single-port	Event connection is not possible.
interrupt controller	Issues an interrupt request to the CPU, starts DMAC data transfer, and starts DTC data transfer.		

18.3.3 Operation of Peripheral Timer Modules When Event is Input

The operations are performed depending on the ELOPA, ELOPB, and ELOPD registers when an event is input.

(1) Counting 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) Counting 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. See the descriptions on the bit in the relevant timer section.

18.3.4 Operation of A/D and D/A Converters when Event is Input

The A/D converter starts A/D when the ADCSR.ADST bit*1 is set to 1.

Note 1. See the description on the bit in the A/D converter section.

18.3.5 Port Operation upon Event Input and Event Generation

The port operation to be performed upon event input to the port can be set and the operation causing the port 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 any bit in an 8-bit port. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same 8-bit port.
A single-port can be set by specifying any bit in the port*1 to which an event can be connected using the PEL0 and PEL1 registers. A port group can be set by specifying any one or more bits in the port*1 to which an event can be connected using the PGC1 register. One input port group and one output port group can be set in the same port.
If the port bit is specified as both a single-port and a member of a port group, both functions are effective when the relevant port is input, whereas only the port group function is effective when the relevant port is output.
The input or output direction of ports can be selected using the PDR register.

Note 1. Port B

(2) Event Generation by Input Single-Ports

An input single-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 and PEL1 registers. An example of operation is shown in Figure 18.3.

(3) Output Single-Port Operation upon Event Input

When an event is input to an output single-port, the PODR value of the relevant port changes. The specific change of the PODR value is specified using the PEL0 and PEL1 registers. Thus, the change of the PODR value changes the signal value of the external pin connected to the relevant port. An example of operation is shown in Figure 18.3.

(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 PGC1 registers. When an event is input to an input port group, the signal value of the external pin upon event input is transferred to PDBF1. 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 18.4.

(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 PGC1 settings. An example of operation is shown in Figure 18.5.

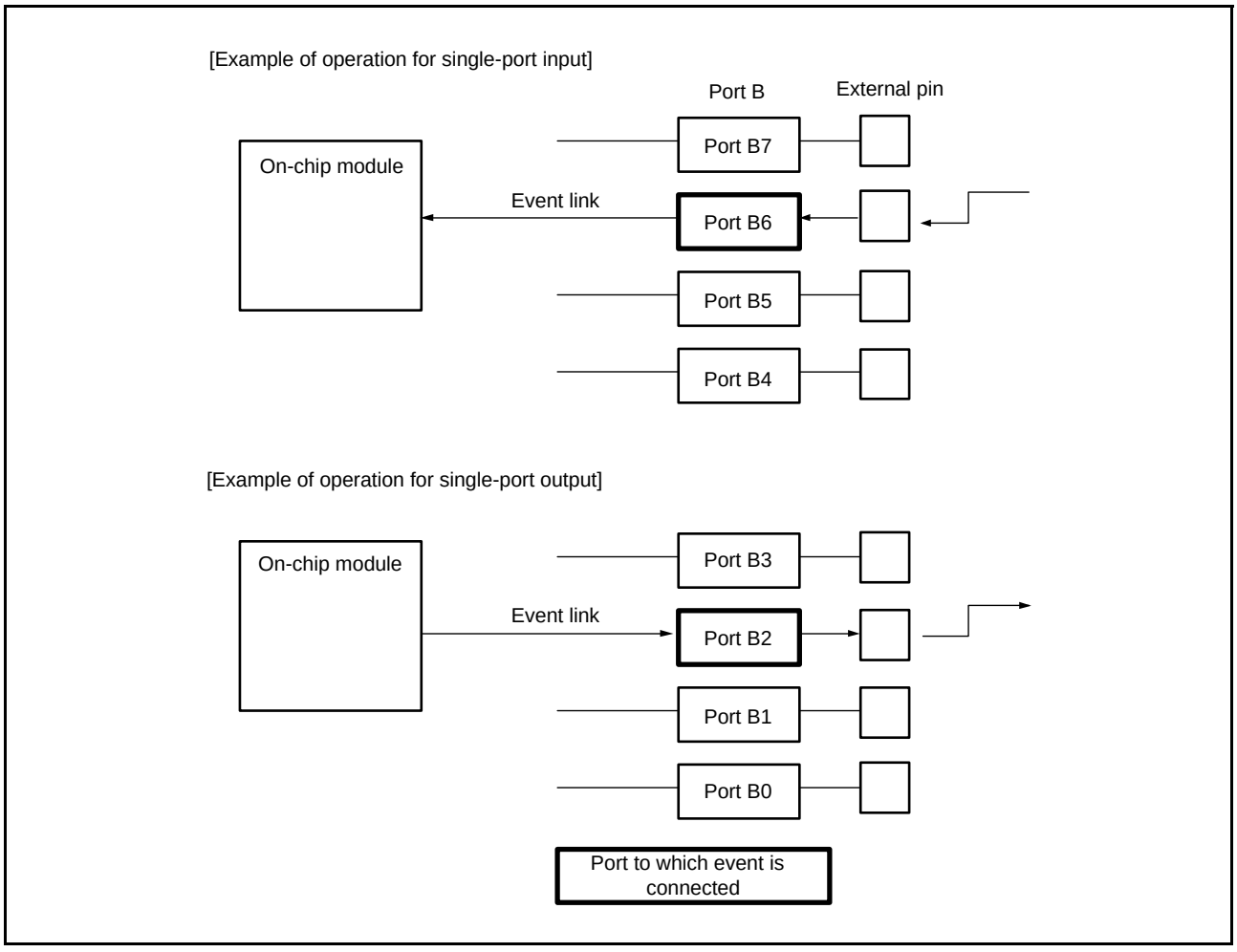


Figure 18.3 Event Linkage Related to Single-Ports

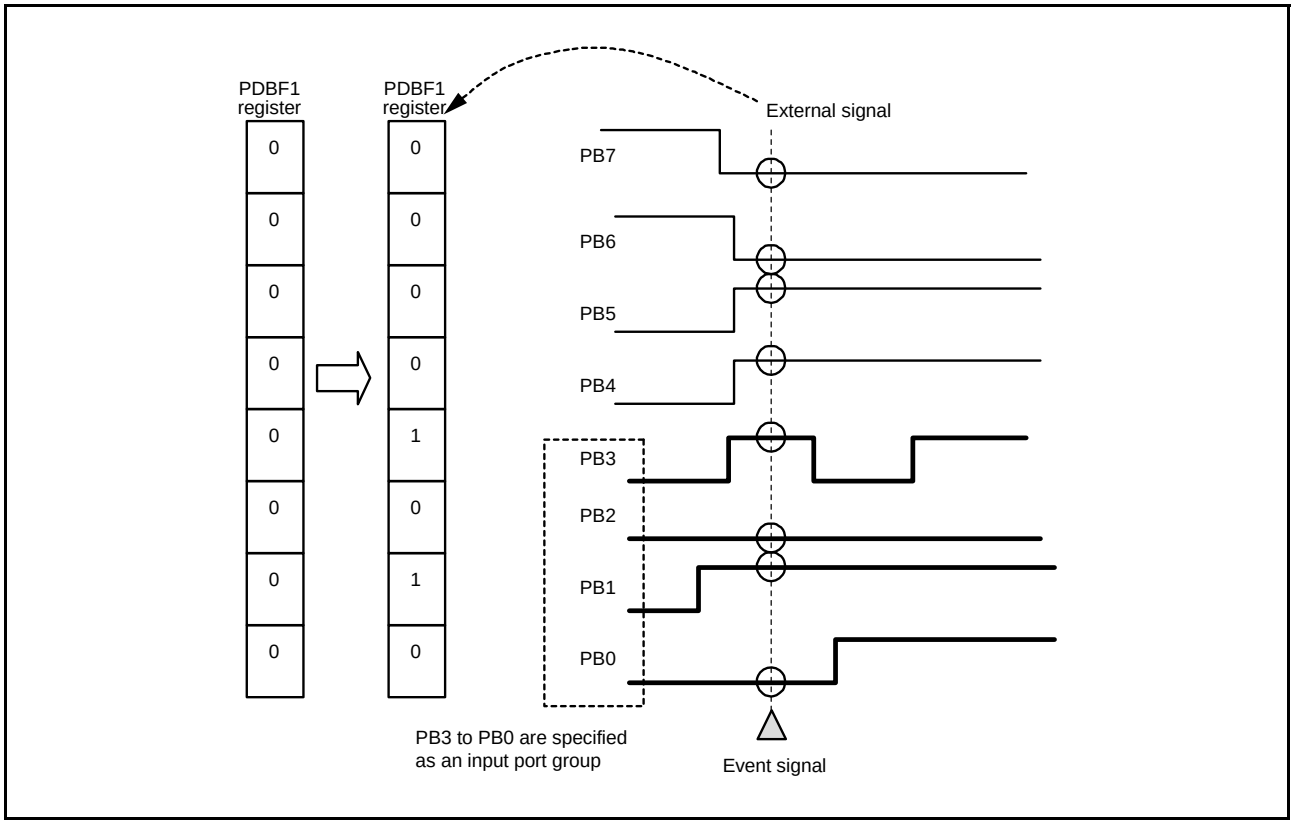


Figure 18.4 Event Linkage Related to Input Port Groups

(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 PDBF1. If another event is input to the input port group in this state, operations are performed depending on the PGC1.PGCOVE bit setting as described below.

- PGC1.PGCOVE = 0 (overwriting PDBF1 is disabled)
  If the PDBF1 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 PDBF1. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGC1.PGCOVE = 1 (overwriting PDBF1 is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to PDBF1.

(b) Output Port Groups

If an output port group is specified so that it should output the PDBF1 value, the PDBF1 value is transferred to PODR 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 (PGC1.PGCO[2:0] bits = 1xx), the PDBF1 data is transferred to PODR, 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 PDBF1.

Examples of operation are shown in Figure 18.5 and Figure 18.6.

(7) Restrictions on Writing to PODR or PDBF by CPU

When the ELCR.ELCON bit is set to 1, write access to the following registers is invalid.

- 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 PDBF1 by the CPU is invalid. However, when the DOC is selected for event input, write access by the CPU is not invalid.
- If port bits are specified as members of the output port group, write access to the relevant bits in PODR by the CPU is invalid.
- If a port bit is specified as an output single-port and the event linkage is set (by ELSRn) for the port, write access to the relevant bit in PODR by the CPU is invalid. However, when the DOC is selected for event input, write access by the CPU is not invalid.

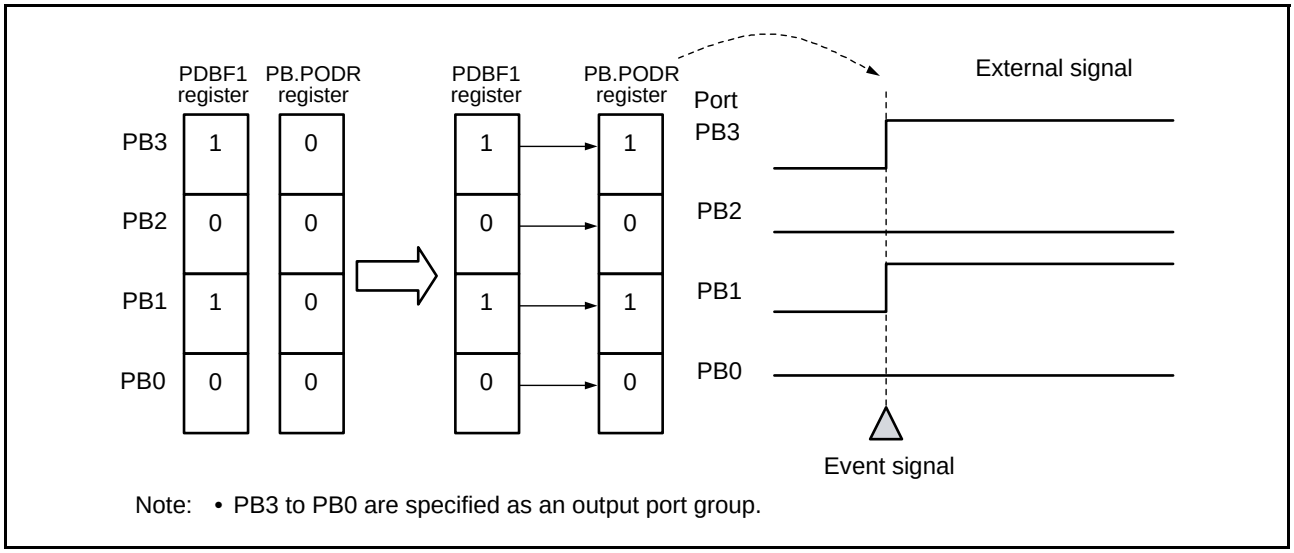
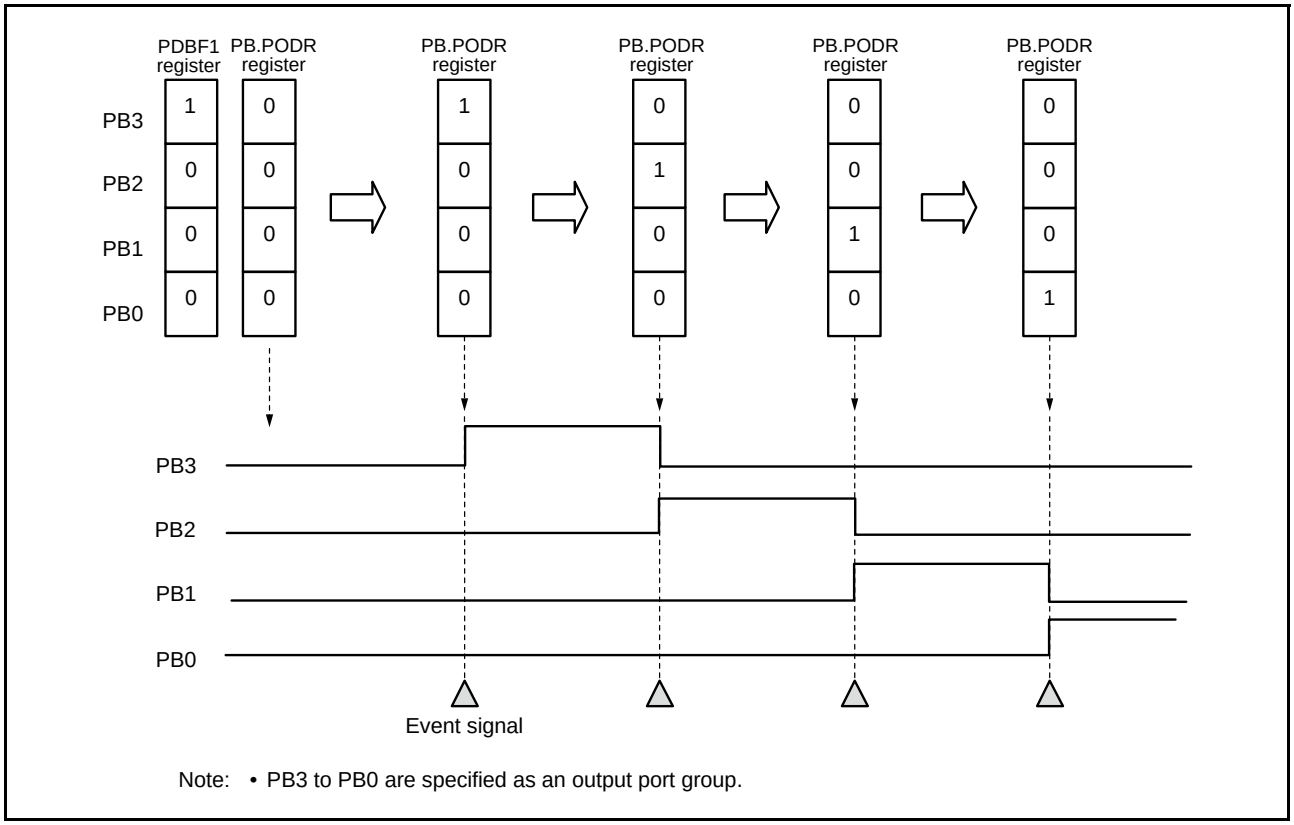


Figure 18.5 Event Linkage Related to Output Port Groups



Note: • PB3 to PB0 are specified as an output port group.

Figure 18.6 Bit-Rotating Operation of Output Port Groups

18.3.6 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.
   PGR1: If ports are used as a port group, set the ports (in bit units) to be grouped.
   PGC1: 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 timers, set the ELOPA, ELOPB and 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 start the module. This allows the event output from the module to start the module to which an event is linked as specified.
7. To stop event linkage of some independent modules, set 00000000b to the ELSRn.ELS[7:0] bits corresponding to the modules. To stop linkage of all the events, clear the ELCR.ELCON bit to 0.

18.4 Usage Notes

18.4.1 Setting the ELSRn Register

(1) Setting the ELSR18 register

Specify an event number from among 01100011 (63h) to 01101001 (6Ah) that are defined as event numbers; the other settings are prohibited.

(2) Setting the ELSR24, or ELSR25 register

Setting the DOC data operation condition met signal (01101010 (6Ah) is prohibited.

18.4.2 Setting the Bit-Rotating Operation of the Output Port Groups

When the values of PDBF1 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 PCLK cycle. If not, the normal operation cannot be provided.

18.4.3 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is completed.

18.4.4 Setting Clocks

To connect events output by the peripheral modules with each other and operate the modules, 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 or software standby mode).

18.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the ELC to be halted. Register access is enabled by canceling the module stop state. For details, see section 11, Low Power Consumption.

19. I/O Ports

19.1 Overview

The I/O ports function as a general I/O port, an I/O pin of a peripheral module, or an input pin for an interrupt.
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 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 20, Multi-Function Pin Controller (MPC).
Products with 48-pin and 64-pin packages include port switching register A (PSRA) and port switching register B (PSRB). However, when ports PC0 and PC1 are selected in port switching register A (PSRA), input to ports PB6 and PB7 and the output port event function of the ELC cannot be used. When ports PC0 to PC3 are selected in port switching register B (PSRB), input to ports PB0, PB1, PB3, and PB5 and the output port event function of the ELC cannot be used.
The configuration of the I/O ports differs depending on the package. Table 19.1 lists the specifications of I/O ports, and Table 19.2 lists the port functions.

Table 19.1 Specifications of I/O Ports

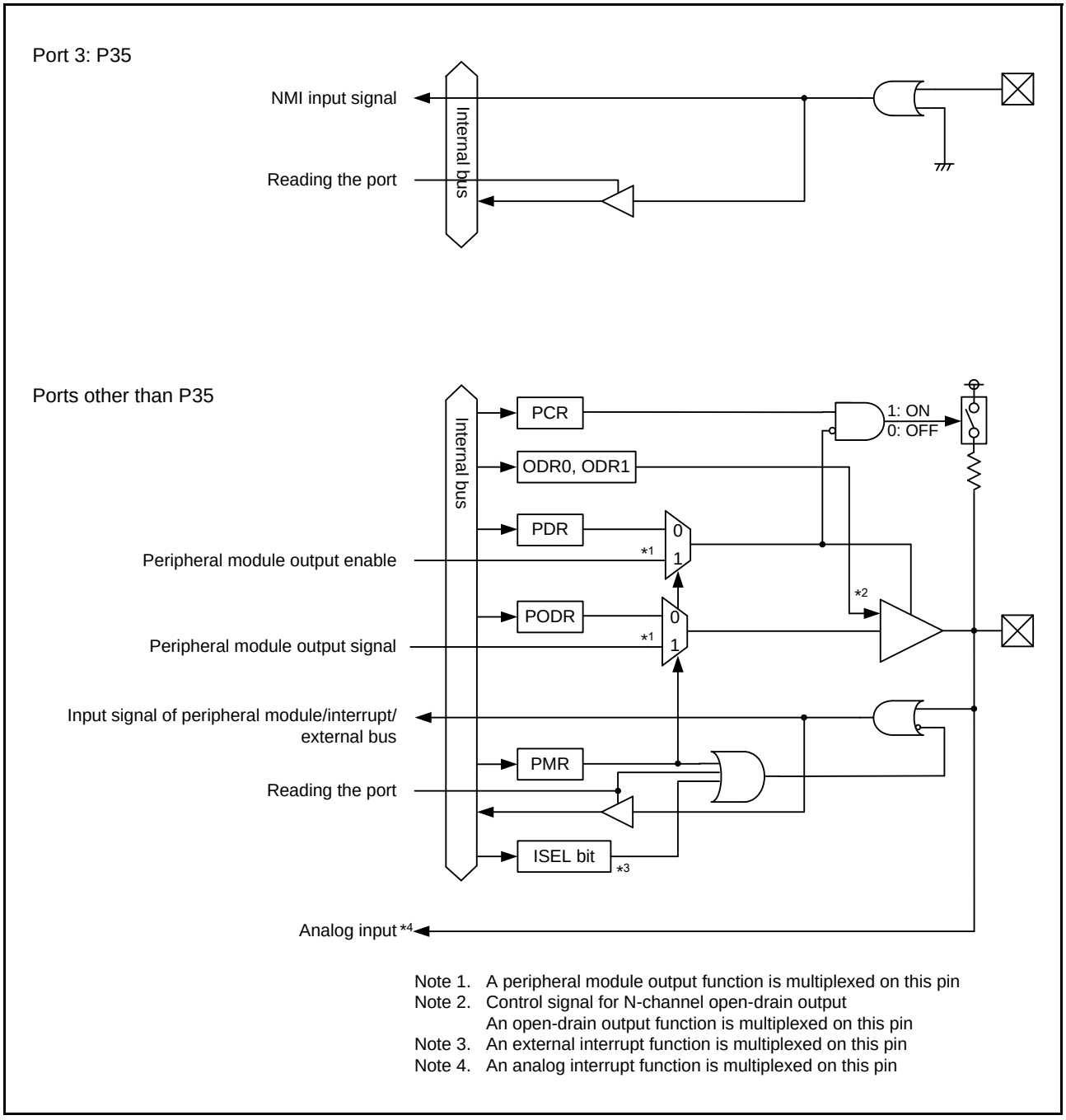
Port	Package		Package		Package	
	100 Pins	Number of Pin	64 Pins	Number of Pin	48 Pins	Number of Pin
PORT0	P03, P05, P07	3	P03, P05	2	Not provided	0
PORT1	P12 to P17	6	P14 to P17	4	P14 to P17	4
PORT2	P20 to P27	8	P26, P27	2	P26, P27	2
PORT3	P30 to P37	8	P30 to P32, P35 to P37	6	P30, P31, P35 to P37	5
PORT4	P40 to P47	8	P40 to P44, P46	6	P40 to P42, P46	4
PORT5	P50 to P55	6	P54, P55	2	Not provided	0
PORTA	PA0 to PA7	8	PA0, PA1, PA3, PA4, PA6	5	PA1, PA3, PA4, PA6	4
PORTB	PB0 to PB7	8	PB0, PB1, PB3, PB5 to PB7	6	PB0, PB1, PB3, PB5	4
PORTC	PC0 to PC7	8	PC2 to PC7	6	PC4 to PC7	4
PORTD	PD0 to PD7	8	Not provided	0	Not provided	0
PORTE	PE0 to PE7	8	PE0 to PE5	6	PE1 to PE4	4
PORTH	PH0 to PH3	4	PH0 to PH3	4	PH0 to PH3	4
PORTJ	PJ1, PJ3	2	Not provided	0	Not provided	0
	Total of Pins	85	Total of Pins	49	Total of Pins	35

Table 19.2 Port Functions

Port	Pin	Input Pull-up	Open Drain Output	Driving Abilify Switching	5-V Tolerant
PORT0	P03, P05, P07	○	—	Fixed to normal output	—
PORT1	P12, P13, P16, P17	○	○	○	○
	P14	○	—	○	—
	P15	○	○	○	—
PORT2	P20 to P25	○	—	Fixed to normal output	—
	P26, P27	○	○	Fixed to normal output	—
PORT3	P30, P32 to P34	○	○	Fixed to normal output	—
	P31, P36, P37	○	—	Fixed to normal output	—
	P35	—	—	—	—
PORT4	P40 to P47	○	—	Fixed to normal output	—
PORT5	P50 to P55	○	—	Fixed to normal output	—
PORTA	PA0	○	—	Fixed to normal output	—
	PA1 to PA7	○	○	Fixed to normal output	—
PORTB	PB0, PB1, PB3, PB5 to PB7	○	○	○	—
	PB2, PB4	○	—	○	—
PORTC	PC0 to PC7	○	○	○	—
PORTD	PD0 to PD7	○	—	Fixed to normal output	—
PORTE	PE0 to PE2	○	○	Fixed to normal output	—
	PE3 to PE7	○	—	Fixed to normal output	—
PORTH	PH0 to PH3	○	—	Fixed to normal output	—
PORTJ	PJ1, PJ3	○	—	Fixed to normal output	—

Specifying input pull-up, open-drain output, switching of driving ability, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

19.2 I/O Port Configuration



Note 1. A peripheral module output function is multiplexed on this pin
Note 2. Control signal for N-channel open-drain output
An open-drain output function is multiplexed on this pin
Note 3. An external interrupt function is multiplexed on this pin
Note 4. An analog interrupt function is multiplexed on this pin

Figure 19.1 I/O Port Configuration

19.3 Register Descriptions

19.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTH.PDR 0008 C011h, PORTJ.PDR 0008 C012h

	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 5, A to E, H, J

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. In products with less than 100 pins, the bit corresponding to the port m pin that does not exist is reserved. Write 1 (output) to the bit.

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.

19.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTH.PODR 0008 C031h, PORTJ.PODR 0008 C032h

	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 5, A to E, H, J

PODR holds the data to be output from the pins used for general output ports.
In products with less than 100 pins, the bit corresponding to the port m pin that does not exist is reserved. Write 0 (low output) to the bit.
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.

19.3.3 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, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h

	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 5, A to E, H, J

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.

19.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h

	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 5, A to E, H, J

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units. In products with less than 100 pins, the bit corresponding to the port m pin that does not exist is reserved. Write 0 (general I/O port) to the bit.

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.

19.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT1.ODR0 0008 C082h, PORT3.ODR0 0008 C086h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTE.ODR0 0008 C09Ch

	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 output	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	• PA1, PB1, PC1 b2 0: CMOS output 1: N-channel open-drain output 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 output 1 0: P-channel open-drain output 1 1: Hi-Z	R/W
b3	B3			R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain output	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 output	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1, 3, A to C, E

In products with less than 100 pins, the bit corresponding to the port m pin that does not exist is reserved. Write 0 (CMOS output) to the bit.

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.

19.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h

	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 output	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 output	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 output	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 output	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1 to 3, A to C

In products with less than 100 pins, the bit corresponding to the port m pin that does not exist is reserved. Write 0 (CMOS output) to the bit. 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.

19.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTH.PCR 0008 C0D1h, PORTJ.PCR 0008 C0D2h

	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 5, A to E, H, J

While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.

When a pin is set as 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 PCR.

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 reserved. A reserved bit is read as 0. The write value should be 0.

19.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT1.DSCR 0008 C0E1h, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh

	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 = 1, B, C

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.

19.3.9 Port Switching Register A (PSRA)

Address(es): PORT.PSRA 0008 C121h

	b7	b6	b5	b4	b3	b2	b1	b0
	PSEL7	PSEL6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	PSEL6	PB6/PC0 Switching	0: PB6 general I/O port function is selected 1: PC0 general I/O port function is selected	R/W
b7	PSEL7	PB7/PC1 Switching	0: PB7 general I/O port function is selected 1: PC1 general I/O port function is selected	R/W

The PSRA register is for 64-pin packages. The PSRA register is used to select either the general I/O functions of PB6 and PB7 or those of PC0 and PC1. When 1 is written to the PSEL6 and PSEL7 bits, port C can be used as an 8-bit port. As for the I/O functions of the peripheral functions, functions multiplexed with PB6 and PB7 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.
Rewriting to this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

19.3.10 Port Switching Register B (PSRB)

Address(es): PORT.PSRB 0008 C120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSEL5	—	PSEL3	—	PSEL1	PSEL0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSEL0	PB0/PC0 Switching	0: PB0 general I/O port function is selected 1: PC0 general I/O port function is selected	R/W
b1	PSEL1	PB1/PC1Switching	0: PB1 general I/O port function is selected 1: PC1 general I/O port function is selected	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PSEL3	PB3/PC2 Switching	0: PB3 general I/O port function is selected 1: PC2 general I/O port function is selected	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	PSEL5	PB5/PC3 Switching	0: PB5 general I/O port function is selected 1: PC3 general I/O port function is selected	R/W
b7, b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The PSRB register is for 48-pin packages. The PSRB register is used to select either the general I/O functions of PB0, PB1, PB3, and PB5 and those of PC0 to PC3. When 1 is written to the PSEL0, PSEL1, PSEL3, and PSEL5 bits, port C can be used as an 8-bit port.

As for the I/O functions of the peripheral functions, functions multiplexed with PB0, PB1, PB3, and PB5 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.

Rewriting to this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

19.4 Treatment of Unused Pins

The treatment of unused pins is listed in Table 19.3.

Table 19.3 Treatment of Unused Pins

Pin Name	Description
MD	(Always 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.
P36/EXTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P36). When this pin is not used as port P36 either, it is configured in the same way as port 0 to 5.
P37/XTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P37). When this pin is not used as port P37 either, it is configured in the same way as port 0 to 5. When the external clock is input to the EXTAL pin, leave this pin open.
XCIN	Connect this pin to VSS via a pull-down resistor.
XCOUT	Leave this pin open.
Port 0 to 5 Port A to E, H, J	• 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.

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. Multi-Function Pin Controller (MPC)

20.1 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.

Table 20.1 shows the allocation of pin functions to multiple pins. The symbols ○ and × in the table indicate whether the pins are or are not present on the given package. Allocating the same function to more than one pin is prohibited.

Table 20.1 Allocation of Pin Functions to Multiple Pins (1 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
Interrupt		NMI (input)	P35	○	○	○
Interrupt	IRQ0	IRQ0 (input)	P30	○	○	○
			PD0	○	×	×
			PH1	○	○	○
	IRQ1	IRQ1 (input)	P31	○	○	○
			PD1	○	×	×
			PH2	○	○	○
	IRQ2	IRQ2 (input)	P32	○	○	×
			P12	○	×	×
			PD2	○	×	×
	IRQ3	IRQ3 (input)	P33	○	×	×
			P13	○	×	×
			PD3	○	×	×
	IRQ4	IRQ4 (input)	PB1	○	○	○
			P14	○	○	○
			P34	○	×	×
			PD4	○	×	×
	IRQ5	IRQ5 (input)	PA4	○	○	○
			P15	○	○	○
			PD5	○	×	×
			PE5	○	○	×
	IRQ6	IRQ6 (input)	PA3	○	○	○
			P16	○	○	○
			PD6	○	×	×
			PE6	○	×	×
	IRQ7	IRQ7 (input)	PE2	○	○	○
			P17	○	○	○
			PD7	○	×	×
			PE7	○	×	×

Table 20.1 Allocation of Pin Functions to Multiple Pins (2 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
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	○	×	×
			PB7	○	○	×
			PC5	○	○	○
		MTIOC3C (input/output)	P16	○	○	○
			PC0	○	×	×
			PC6	○	○	○
			PJ3	○	×	×
		MTIOC3D (input/output)	P16	○	○	○
			P23	○	×	×
			PB6	○	○	×
			PC4	○	○	○

Table 20.1 Allocation of Pin Functions to Multiple Pins (3 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
Multi-function timer unit 2	MTU4	MTIOC4A (input/output)	P24	○	×	×
			PA0	○	○	×
			PB3	○	○	○
			PE2	○	○	○
		MTIOC4B (input/output)	P30	○	○	○
			P54	○	○	×
			PC2	○	○	×
			PD1	○	×	×
			PE3	○	○	○
		MTIOC4C (input/output)	P25	○	×	×
			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	○	○	○

Table 20.1 Allocation of Pin Functions to Multiple Pins (4 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
Port output enable 2	POE0	POE0# (input)	PC4	○	○	○
			PD7	○	×	×
	POE1	POE1# (input)	PB5	○	○	○
			PD6	○	×	×
	POE2	POE2# (input)	P34	○	×	×
			PA6	○	○	○
			PD5	○	×	×
	POE3	POE3# (input)	P33	○	×	×
			PB3	○	○	○
			PD4	○	×	×
	POE8	POE8# (input)	P17	○	○	○
			P30	○	○	○
			PD3	○	×	×
			PE3	○	○	○
8-bit timer	TMR0	TMO0 (output)	P22	○	×	×
			PB3	○	○	○
			PH1	○	○	○
		TMCI0 (input)	P21	○	×	×
			PB1	○	○	○
			PH3	○	○	○
		TMRI0 (input)	P20	○	×	×
			PA4	○	○	○
			PH2	○	○	○
	TMR1	TMO1 (output)	P17	○	○	○
			P26	○	○	○
		TMCI1 (input)	P12	○	×	×
			P54	○	○	×
			PC4	○	○	○
		TMRI1 (input)	P24	○	×	×
			PB5	○	○	○
	TMR2	TMO2 (output)	P16	○	○	○
			PC7	○	○	○
		TMCI2 (input)	P15	○	○	○
			P31	○	○	○
			PC6	○	○	○
		TMRI2 (input)	P14	○	○	○
			PC5	○	○	○

Table 20.1 Allocation of Pin Functions to Multiple Pins (5 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
8-bit timer	TMR3	TMO3 (output)	P13	○	×	×
			P32	○	○	×
			P55	○	○	×
		TMCI3 (input)	P27	○	○	○
			P34	○	×	×
			PA6	○	○	○
		TMRI3 (input)	P30	○	○	○
			P33	○	×	×
Serial communications interface	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	○	○	○
	SCI5/IrDA	RXD5 (input)/ SMISO5 (input/output)/ SSCL5 (input/output)/ IRRXD5 (input)	PA2	○	×	×
			PA3	○	○	○
			PC2	○	○	×
		TXD5 (output)/ SMOSI5 (input/output)/ SSDA5 (input/output)/ IRTXD5 (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)	P33	○	×	×
			PB0	○	○	○
		TXD6 (output)/ SMOSI6 (input/output)/ SSDA6 (input/output)	P32	○	○	×
			PB1	○	○	○
		SCK6 (input/output)	P34	○	×	×
			PB3	○	○	○
		CTS6# (input)/ RTS6# (output)/ SS6# (input)	PB2	○	×	×
			PJ3	○	×	×

Table 20.1 Allocation of Pin Functions to Multiple Pins (6 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
Serial communications interface	SCI9	RXD9 (input)/ SMISO9 (input/output)/ SSCL9 (input/output)	PB6	○	○	×
		TXD9 (output)/ SMOSI9 (input/output)/ SSDA9 (input/output)	PB7	○	○	×
		SCK9 (input/output)	PB5	○	○	×
		CTS9# (input)/ RTS9# (output)/ SS9# (input)	PB4	○	×	×
	SCI12	RXD12 (input)/ SMISO12 (input/output)/ SSCL12 (input/output)/ RXDX12 (input)	PE2	○	○	○ (SMISO12 function is not available)
		TXD12 (output)/ SMOSI12 (input/output)/ SSDA12 (input/output)/ TXDX12 (output)/ SIOX12 (input/output)	PE1	○	○	○ (SMOSI12 function is not available)
		SCK12 (input/output)	PE0	○	○	×
		CTS12# (input)/ RTS12# (output)/ SS12# (input)	PE3	○	○	○ (SS12# function is not available)
I^2C bus interface	RIIC0	SCL (input/output)	P16	○	○	○
			P12	○	×	×
		SDA (input/output)	P17	○	○	○
			P13	○	×	×
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5	○	×	×
			PB0	○	○	○
			PC5	○	○	○
		MOSIA (input/output)	P16	○	○	○
			PA6	○	○	○
			PC6	○	○	○
		MISOA (input/output)	P17	○	○	○
			PA7	○	×	×
			PC7	○	○	○
Serial peripheral interface	RSPI0	SSLA0 (input/output)	PA4	○	○	○
			PC4	○	○	○
		SSLA1 (output)	PA0	○	○	×
			PC0	○	×	×
		SSLA2 (output)	PA1	○	○	○
			PC1	○	×	×
		SSLA3 (output)	PA2	○	×	×
			PC2	○	○	×
Realtime clock		RTCOUT (output)	P16	○	○	×
			P32	○	○	×

Table 20.1 Allocation of Pin Functions to Multiple Pins (7 / 7)

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100 pin	64 pin	48 pin
12-bit A/D converter		AN000 (input)*1	P40	○	○	○
		AN001 (input)*1	P41	○	○	○
		AN002 (input)*1	P42	○	○	○
		AN003 (input)*1	P43	○	○	×
		AN004 (input)*1	P44	○	○	×
		AN005 (input)*1	P45	○	×	×
		AN006 (input)*1	P46	○	○	○
		AN007 (input)*1	P47	○	×	×
		AN008 (input)*1	PE0	○	○	×
		AN009 (input)*1	PE1	○	○	○
		AN010 (input)*1	PE2	○	○	○
		AN011 (input)*1	PE3	○	○	○
		AN012 (input)*1	PE4	○	○	○
		AN013 (input)*1	PE5	○	○	×
		AN014 (input)*1	PE6	○	×	×
		AN015 (input)*1	PE7	○	×	×
		ADTRG0# (input)	P07	○	×	×
			P16	○	○	○
			P25	○	×	×
Clock frequency accuracy measurement circuit		CACREF (input)	PA0	○	○	×
			PC7	○	○	○
			PH0	○	○	○
Comparator A		CMPA1 (input)*1	PE3	○	○	○
		CMPA2 (input)*1	PE4	○	○	○
		CVREFA (input)*1	PA1	○	○	○

Note 1. 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.

20.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

20.2.1 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.

20.2.2 P07 Pin Function Control Register (P07PFS)

Address(es): P07PFS 0008 C147h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The Pmn pin function control register (PmnPFS) selects the pin function. Bits PSEL[3:0] 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 20.2 Register Settings for Input/Output Pin Function in 100-Pin

	pin
PSEL[3:0] Settings	**P07**
0000b (initial value)	Hi-Z
1001b	ADTRG0#

20.2.3 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[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5, b4	—	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 input switch (100 pins) P13: IRQ3 input switch (100 pins) P14: IRQ4 input switch (100 pins, 64 pins, 48 pins) P15: IRQ5 input switch (100 pins, 64 pins, 48 pins) P16: IRQ6 input switch (100 pins, 64 pins, 48 pins) P17: IRQ7 input switch (100 pins, 64 pins, 48 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 20.3 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0]	Pin					
Settings	P12	P13	P14	P15	P16	P17
0000b (initial value)				Hi-Z		
0001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
0010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
0101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
0111b	—	—	—	—	RTCOUT	POE8#
1001b	—	—	—	—	ADTRG0#	—
1010b	—	—	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
1011b	—	—	CTS1# RTS1# SS1#	—	—	—
1101b	—	—	—	—	MOSIA	MISOA
1111b	SCL	SDA	—	—	SCL	SDA

—: Do not specify this value.

Table 20.4 Register Settings for Input/Output Pin Function in 64-Pin

PSEL[3:0] Settings	Pin P14	P15	P16	P17
0000b (initial value)			Hi-Z	
0001b	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
0010b	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
0101b	TMRI2	TMCI2	TMO2	TMO1
0111b	—	—	RTCOUT	POE8#
1001b	—	—	ADTRG0#	—
1010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
1011b	CTS1# RTS1# SS1#	—	—	—
1101b	—	—	MOSIA	MISOA
1111b	—	—	SCL	SDA

—: Do not specify this value.

Table 20.5 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin P14	P15	P16	P17
0000b (initial value)			Hi-Z	
0001b	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
0010b	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
0101b	TMRI2	TMCI2	TMO2	TMO1
0111b	—	—	—	POE8#
1001b	—	—	ADTRG0#	—
1010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
1011b	CTS1# RTS1# SS1#	—	—	—
1101b	—	—	MOSIA	MISOA
1111b	—	—	SCL	SDA

—: Do not specify this value.

20.2.4 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
	—	—	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 20.6 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin P20	P21	P22	P23	P24	P25	P26	P27
0000b (initial value)	Hi-Z							
0001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
0010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
0101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
1001b	—	—	—	—	—	ADTRG0#	—	—
1010b	—	—	—	—	—	—	TXD1 SMOSI1 SSDA1	SCK1

—: Do not specify this value.

Table 20.7 Register Settings for Input/Output Pin Function in 64-Pin and 48-Pin

PSEL[3:0] Settings	Pin P26	P27
0000b (initial value)	Hi-Z	
0001b	MTIOC2A	MTIOC2B
0101b	TMO1	TMCI3
1010b	TXD1 SMOSI1 SSDA1	SCK1

20.2.5 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[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5, b4	—	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 input switch (100 pins, 64 pins, 48 pins) P31: IRQ1 input switch (100 pins, 64 pins, 48 pins) P32: IRQ2 input switch (100 pins, 64 pins) P33: IRQ3 input switch (100 pins) P34: IRQ4 input switch (100 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 20.8 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin				
	P30	P31	P32	P33	P34
0000b (initial value)	Hi-Z				
0001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
0101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
0111b	POE8#	—	RTCOUT	POE3#	POE2#
1010b	RXD1 SMISO1 SSCL1	—	—	—	—
1011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6

—: Do not specify this value.

Table 20.9 Register Settings for Input/Output Pin Function in 64-Pin

PSEL[3:0] Settings	Pin		
	P30	P31	P32
0000b (initial value)	Hi-Z		
0001b	MTIOC4B	MTIOC4D	MTIOC0C
0101b	TMRI3	TMCI2	TMO3
0111b	POE8#	—	RTCOUT
1010b	RXD1 SMISO1 SSCL1	—	—
1011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6

—: Do not specify this value.

Table 20.10 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin	
	P30	P31
0000b (initial value)	Hi-Z	
0001b	MTIOC4B	MTIOC4D
0101b	TMRI3	TMCI2
0111b	POE8#	—
1010b	RXD1 SMISO1 SSCL1	—
1011b	—	CTS1# RTS1# SS1#

—: Do not specify this value.

20.2.6 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 (100 pins, 64 pins, 48 pins) P41: AN001 (100 pins, 64 pins, 48 pins) P42: AN002 (100 pins, 64 pins, 48 pins) P43: AN003 (100 pins, 64 pins) P44: AN004 (100 pins, 64 pins) P45: AN005 (100 pins) P46: AN006 (100 pins, 64 pins, 48 pins) P47: AN007 (100 pins)	R/W

20.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 4, 5)

Address(es): P54PFS 0008 C16Ch, P55PFS 0008 C16Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 20.11 Register Settings for Input/Output Pin Function in 100-Pin and 64-Pin

	Pin	
PSEL[3:0] Settings	**P54**	**P55**
0000b (initial value)	Hi-Z	
0001b	MTIOC4B	MTIOC4D
0101b	TMCI1	TMO3

20.2.8 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

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5, b4	—	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 PA3: IRQ6 input switch (100 pins, 64 pins, 48 pins) PA4: IRQ5 input switch (100 pins, 64 pins, 48 pins)	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PA1: CVREFA (100 pins, 64 pins, 48 pins)	R/W

Table 20.12 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin							
	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
0000b (initial value)	Hi-Z							
0001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
0010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
0101b	—	—	—	—	TMRI0	—	TMCI3	—
0111b	CACREF	—	—	—	—	—	POE2#	—
1010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	—	—	—
1011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
1101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA

—: Do not specify this value.

Table 20.13 Register Settings for Input/Output Pin Function in 64-Pin

PSEL[3:0] Settings	Pin				
	PA0	PA1	PA3	PA4	PA6
0000b (initial value)	Hi-Z				
0001b	MTIOC4A	MTIOC0B	MTIOC0D	MTIC5U	MTIC5V
0010b	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
0101b	—	—	—	TMRI0	TMCI3
0111b	CACREF	—	—	—	POE2#
1010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	—
1011b	—	—	—	—	CTS5# RTS5# SS5#
1101b	SSLA1	SSLA2	—	SSLA0	MOSIA

—: Do not specify this value.

Table 20.14 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin			
	PA1	PA3	PA4	PA6
0000b (initial value)	Hi-Z			
0001b	MTIOC0B	MTIOC0D	MTIC5U	MTIC5V
0010b	MTCLKC	MTCLKD	MTCLKA	MTCLKB
0101b	—	—	TMRI0	TMCI3
0111b	—	—	—	POE2#
1010b	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	—
1011b	—	—	—	CTS5# RTS5# SS5#
1101b	SSLA2	—	SSLA0	MOSIA

—: Do not specify this value.

20.2.9 PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB2PFS 0008 C19Ah, PB3PFS 0008 C19Bh, PB4PFS 0008 C19Ch, PB5PFS 0008 C19Dh, PB6PFS 0008 C19Eh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5, b4	—	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 PB1: IRQ4 (100 pins, 64 pins, 48 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 20.15 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
0000b (initial value)	Hi-Z							
0001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
0010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
0101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
0111b	—	—	—	POE3#	—	POE1#	—	—
1010b	—	—	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
1011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
1101b	RSPCKA	—	—	—	—	—	—	—

—: Do not specify this value.

Table 20.16 Register Settings for Input/Output Pin Function in 64-Pin

PSEL[3:0] Settings	Pin					
	PB0	PB1	PB3	PB5	PB6	PB7
0000b (initial value)	Hi-Z					
0001b	MTIC5W	MTIOC0C	MTIOC0A	MTIOC2A	MTIOC3D	MTIOC3B
0010b	—	MTIOC4C	MTIOC4A	MTIOC1B	—	—
0101b	—	TMCI0	TMO0	TMRI1	—	—
0111b	—	—	POE3#	POE1#	—	—
1010b	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
1011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—	—	—
1101b	RSPCKA	—	—	—	—	—

—: Do not specify this value.

Table 20.17 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin			
	PB0	**PB1**	**PB3**	**PB5**
0000b (initial value)			Hi-Z	
0001b	MTIC5W	MTIOC0C	MTIOC0A	MTIOC2A
0010b	—	MTIOC4C	MTIOC4A	MTIOC1B
0101b	—	TMCI0	TMO0	TMRI1
0111b	—	—	POE3#	POE1#
1011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—
1101b	RSPCKA	—	—	—

—: Do not specify this value.

20.2.10 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
	—	—	—	—	PSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 20.18 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin							
	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0000b (initial value)	Hi-Z							
0001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
0010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
0101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
0111b	—	—	—	—	POE0#	—	—	CACREF
1010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	SCK5	—	—	—
1011b	CTS5# RTS5# SS5#	—	—	—	—	—	—	—
1101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA

—: Do not specify this value.

Table 20.19 Register Settings for Input/Output Pin Function in 64-Pin

PSEL[3:0] Settings	Pin					
	PC2	PC3	PC4	PC5	PC6	PC7
0000b (initial value)	Hi-Z					
0001b	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
0010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
0101b	—	—	TMCI1	TMRI2	TMCI2	TMO2
0111b	—	—	POE0#	—	—	CACREF
1010b	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	SCK5	—	—	—
1101b	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA

—: Do not specify this value.

Table 20.20 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin			
	PC4	PC5	PC6	PC7
0000b (initial value)			Hi-Z	
0001b	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
0010b	MTCLKC	MTCLKD	MTCLKA	MTCLKB
0101b	TMCI1	TMRI2	TMCI2	TMO2
0111b	POE0#	—	—	CACREF
1010b	SCK5	—	—	—
1101b	SSLA0	RSPCKA	MOSIA	MISOA

—: Do not specify this value.

20.2.11 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[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b5, b4	—	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 input switch (100 pins) PD1: IRQ1 input switch (100 pins) PD2: IRQ2 input switch (100 pins) PD3: IRQ3 input switch (100 pins) PD4: IRQ4 input switch (100 pins) PD5: IRQ5 input switch (100 pins) PD6: IRQ6 input switch (100 pins) PD7: IRQ7 input switch (100 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 20.21 Register Settings for Input/Output Pin Function in 100-Pin

PSEL[3:0] Settings	Pin						
	PD1	PD2	PD3	PD4	PD5	PD6	PD7
0000b (initial value)	Hi-Z						
0001b	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
0111b	—	—	POE8#	POE3#	POE2#	POE1#	POE0#

—: Do not specify this value.

20.2.12 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[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the tables below.	R/W
b5, b4	—	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 input switch (100 pins, 64 pins, 48 pins) PE5: IRQ5 input switch (100 pins, 64 pins) PE6: IRQ6 input switch (100 pins) PE7: IRQ7 input switch (100 pins)	R/W
b7	ASEL	Analog Function Select	0: Not used as an analog pin 1: Used as an analog pin PE0:AN008 (100 pins, 64 pins) PE1:AN009 (100 pins, 64 pins, 48 pins) PE2:AN010 (100 pins, 64 pins, 48 pins) PE3:AN011 or CMPA1 (100 pins, 64 pins, 48 pins) PE4:AN012 or CMPA2 (100 pins, 64 pins, 48 pins) PE5:AN013 (100 pins, 64 pins) PE6:AN014 (100 pins) PE7:AN015 (100 pins)	R/W

Table 20.22 Register Settings for Input/Output Pin Function in 100-Pin and 64-Pin

PSEL[3:0] Settings	Pin					
	PE0	PE1	PE2	PE3	PE4	PE5
0000b (initial value)	Hi-Z					
0001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C
0010b	—	—	—	—	MTIOC1A	MTIOC2B
0111b	—	—	—	POE8#	—	—
1100b	SCK12	TXD12 TXDX12 SIOX12 SMOSI12 SSDA12	RXD12 RXDX12 SMISO12 SSCL12	CTS12# RTS12# SS12#	—	—

—: Do not specify this value.

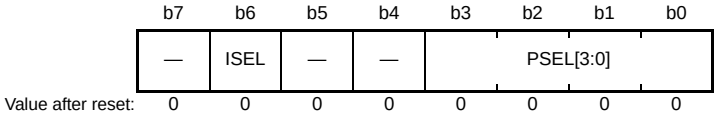
Table 20.23 Register Settings for Input/Output Pin Function in 48-Pin

PSEL[3:0] Settings	Pin			
	PE1	PE2	PE3	PE4
0000b (initial value)	Hi-Z			
0001b	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D
0010b	—	—	—	MTIOC1A
0111b	—	—	POE8#	—
1100b	TXD12 TXDX12 SIOX12 SSDA12	RXD12 RXDX12 SSCL12	CTS12# RTS12#	—

—: Do not specify this value.

20.2.13 PHn Pin Function Control Registers (PHnPFS) (n = 0 to 3)

Address(es): PH0PFS 0008 C1C8h, PH1PFS 0008 C1C9h, PH2PFS 0008 C1CAh, PH3PFS 0008 C1CBh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	This bit selects the peripheral function. For individual pin functions, see the table below.	R/W
b5, b4	—	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 (100 pins, 64 pins, 48 pins) PH2: IRQ1 (100 pins, 64 pins, 48 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 20.24 Register Settings for Input/Output Pin Function in 100-Pin, 64-Pin, and 48-Pin

	Pin			
PSEL[3:0] Settings	**PH0**	**PH1**	**PH2**	**PH3**
0000b (initial value)	Hi-Z			
0101b	—	TMO0	TMRI0	TMCI0
0111b	CACREF	—	—	—

—: Do not specify this value.

20.2.14 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[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PSEL[3:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 20.25 Register Settings for Input/Output Pin Function in 100-Pin

	Pin	
PSEL[3:0] Settings	**PJ1**	**PJ3**
0000b (initial value)	Hi-Z	
0001b	MTIOC3A	MTIOC3C
1011b	—	CTS6# RTS6# SS6#

—: Do not specify this value.

20.3 Usage Notes

20.3.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the port mode register (PMR) 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 port mn pin function select register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 5, A to E, H, and, J, n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[3:0] 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.

20.3.2 Notes on MPC Register Setting

(1) Settings of the port mn pin function select register (PmnPFS) should be made only while the PMR register for the target pin is cleared to 0. If a port mn pin function select 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 port mn pin function select 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 port mn pin function selection register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 20.26.

Table 20.26 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[3:0]	
After a reset	0	0	0	0	0000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	×	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	×	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 20.2 to Table 20.25)	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	×	
NMI	×	×	×	×*1	×	Register settings are not required.
Analog inputs and outputs	0	0	1	×*1	×	Set these as general input port pins so that the output buffers are turned off.
EXTAL/XTAL	0	0	×	×*1	×	Set these as general input port pins so that the output buffers are turned off.
XCIN/XCOUT	0	0	×	×*1	×	Set these as general input port pins so that the output buffers are turned off.

×: 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[3:0] 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.

20.3.3 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 port mn pin function select register (PmnPFS) to 1.

21. Multi-Function Timer Pulse Unit 2 (MTU2a)

21.1 Overview

The RX220 Group has a on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with six channels (MTU0 to MTU5).

Table 21.1 lists the specifications of the MTU, Table 21.2 lists the function list. Figure 21.1 shows a block diagram of the MTU.

Table 21.1 Specifications of MTU

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	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 six-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
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
Lower power consumption function	Module stop state can be set.

Table 21.2 MTU Functions (1/3)

Item		MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clock		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
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	—
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 21.2 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	—
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 21.2 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, see section 11, Low Power Consumption.

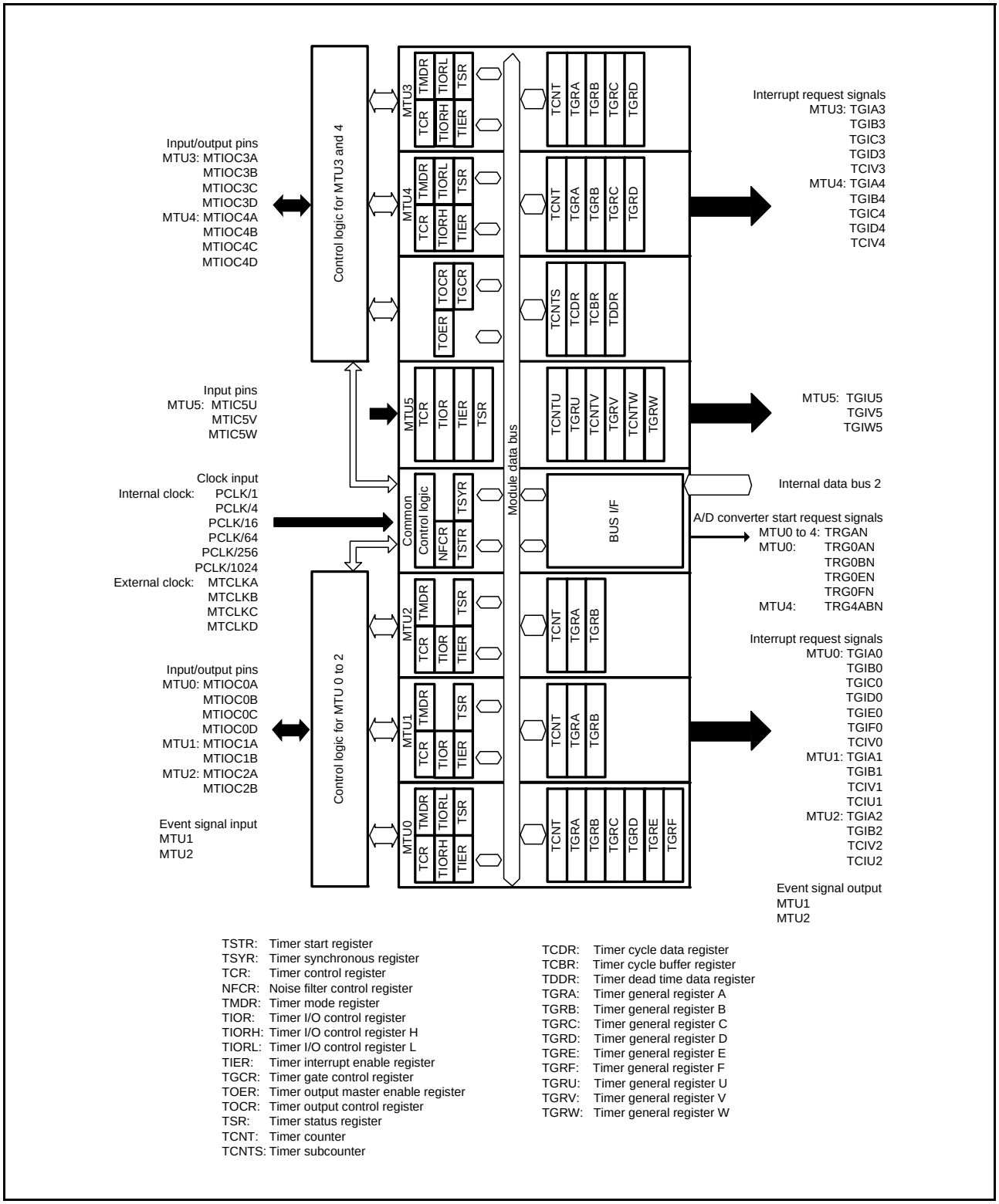


Figure 21.1 Block Diagram of MTU

Table 21.3 lists the pin configuration of the MTU.

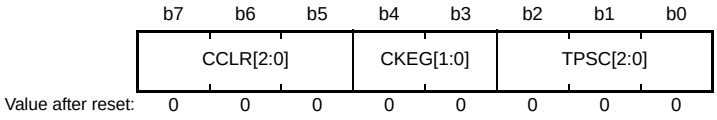
Table 21.3 Pin Configuration of MTU

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

21.2 Register Descriptions

21.2.1 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 21.6 to Table 21.10.	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 21.4 and Table 21.5.	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.
TCR is an 8-bit readable/writable register that 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 counter clock. The clock source can be selected independently for each channel. See Table 21.6 to Table 21.10 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 at both edges = PCLK/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLK/4 or slower. When PCLK/1 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 21.4 and Table 21.5 for details.

Table 21.4 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.SYNC bit to 1.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 21.5 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.SYNC bit 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 21.6 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
	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: 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 21.7 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
	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: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: • This setting is ignored when MTU1 is in phase counting mode.

Table 21.8 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
	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: 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

Note: • This setting is ignored when MTU2 is in phase counting mode.

Table 21.9 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
	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	Internal clock: counts on PCLK/256
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

Table 21.10 TPSC[1:0] (MTU5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
MTU5	0	0	Internal clock: counts on PCLK/1
	0	1	Internal clock: counts on PCLK/4
	1	0	Internal clock: counts on PCLK/16
	1	1	Internal clock: counts on PCLK/64

Note: • Bits 7 to 2 are reserved in MTU5. These bits are read as 0. The write value should be 0.

21.2.2 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 21.11 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

TMDR is an 8-bit readable/writable register that specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

Table 21.11 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. The initial values 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 TGIEC bit in timer interrupt enable register (MTU4.TIER) 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. Refer to Figure 21.40 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 TGIED bit in timer interrupt enable register 3 or 4 (MTU3.TIER or MTU4.TIER) 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. Refer to Figure 21.40 for an illustration of the Tb interval in complementary PWM mode.

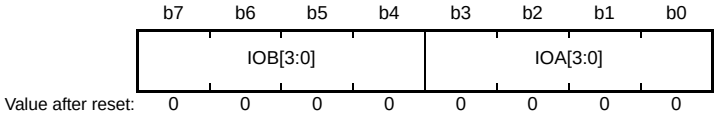
BFE Bit (Buffer Operation E)

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.

21.2.3 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 21.20 MTU1.TIOR: Table 21.22 MTU2.TIOR: Table 21.23 MTU3.TIORH: Table 21.24 MTU4.TIORH: Table 21.26	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables.*1 MTU0.TIORH: Table 21.12 MTU1.TIOR: Table 21.14 MTU2.TIOR: Table 21.15 MTU3.TIORH: Table 21.16 MTU4.TIORH: Table 21.18	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 21.21 MTU3.TIORL: Table 21.25 MTU4.TIORL: Table 21.27	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables.*1 MTU0.TIORL: Table 21.13 MTU3.TIORL: Table 21.17 MTU4.TIORL: Table 21.19	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 21.28	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/V/W) each 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 CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared 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 21.12 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 21.13 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 21.14 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 21.15 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 21.16 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 21.17 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 BFB bit in MTU3.TMDR 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 21.18 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 21.19 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 BFB bit in MTU4.TMDR 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 21.20 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 21.21 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 BFA bit in MTU0.TMDR 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 21.22 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 21.23 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 21.24 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 21.25 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 BFA bit in MTU3.TMDR 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 21.26 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 21.27 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 BFA bit in MTU4.TMDR 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 21.28 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	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Setting prohibited
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

x: Don't care

21.2.4 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

TCNTCMPCLR is an 8-bit readable/writable register that specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

21.2.5 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.
TIER is an 8-bit readable/writable register that 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 or 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 or 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.TGRm (m = E or 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 5m)

Each bit enables or disables interrupt requests (TGIm5) (m = W, V, or U).

21.2.6 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.
TSR is an 8-bit readable/writable register that 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.

21.2.7 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.
TBTM is an 8-bit readable/writable register that 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 reserved. It is read as 0 and the write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

21.2.8 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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTU has one TICCR for MTU1.
TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.

21.2.9 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	0: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W*1
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W*1
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable	0: TCI4V interrupt skipping is not linked 1: TCI4V interrupt skipping is linked	R/W*1
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W*1
b4	DT4BE	Down-Count TRG4BN Enable	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*1
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	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*1
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/B Transfer Timing Select	See Table 21.29 for details.	R/W

Note: • TADCR must not be accessed in 8-bit units; it should be accessed in 16-bit units.

Note: • When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) 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 ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

Note: • If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.

Note 1. Do not set any bit from among b6 and b4 to b0 to 1 unless complementary PWM mode is not selected.

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

Table 21.29 Setting of Transfer Timing by BF[1:0] Bits

Bit 15	Bit 14	
BF1	BF0	Description
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the MTU4.TCNT count.*1
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count.*2
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count.*2

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the MTU4.TCNT count is reached in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal operation mode.
Note 2. These settings are prohibited when complementary PWM mode is not selected.

21.2.10 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 always be accessed in 16-bit units.

TADCORA and TADCORB are 16-bit readable/writable registers. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.
The TADCORA and TADCORB values after reset are FFFFh.

21.2.11 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 always be accessed in 16-bit units.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.
The TADCOBRA and TADCOBRB values after reset are FFFFh.

21.2.12 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 always 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 16-bit readable/writable counter.
TCNT is initialized to 0000h by a reset.

21.2.13 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 always 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. TGR is a 16-bit readable/writable register.
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.

21.2.14 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

TSTR starts or stops 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

21.2.15 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 presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. TCNT synchronous presetting/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 presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W

TSYR selects 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 presetting 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 by means of TCR.CCLR[2:0] bits.

21.2.16 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

TRWER enables or disables 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 conditions]

- When 0 is written to the RWE bit after reading RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUm.TCR, MTUm.TMDR, MTUm.TIORH, MTUm.TIORL, MTUm.TIER, MTUm.TGRA, MTUm.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUm.TCNT (m = 3, 4)

21.2.17 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 the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports.

TOER enables or disables 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 CST3 and CST4 bits in TSTR to 0 (see Figure 21.35 and Figure 21.38).

21.2.18 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 21.30.	R/W
b1	OLSN	Output Level Select N*2, *3	See Table 21.31.	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.

TOCR1 is 8-bit readable/writable registers that 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 21.30 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 21.31 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 21.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

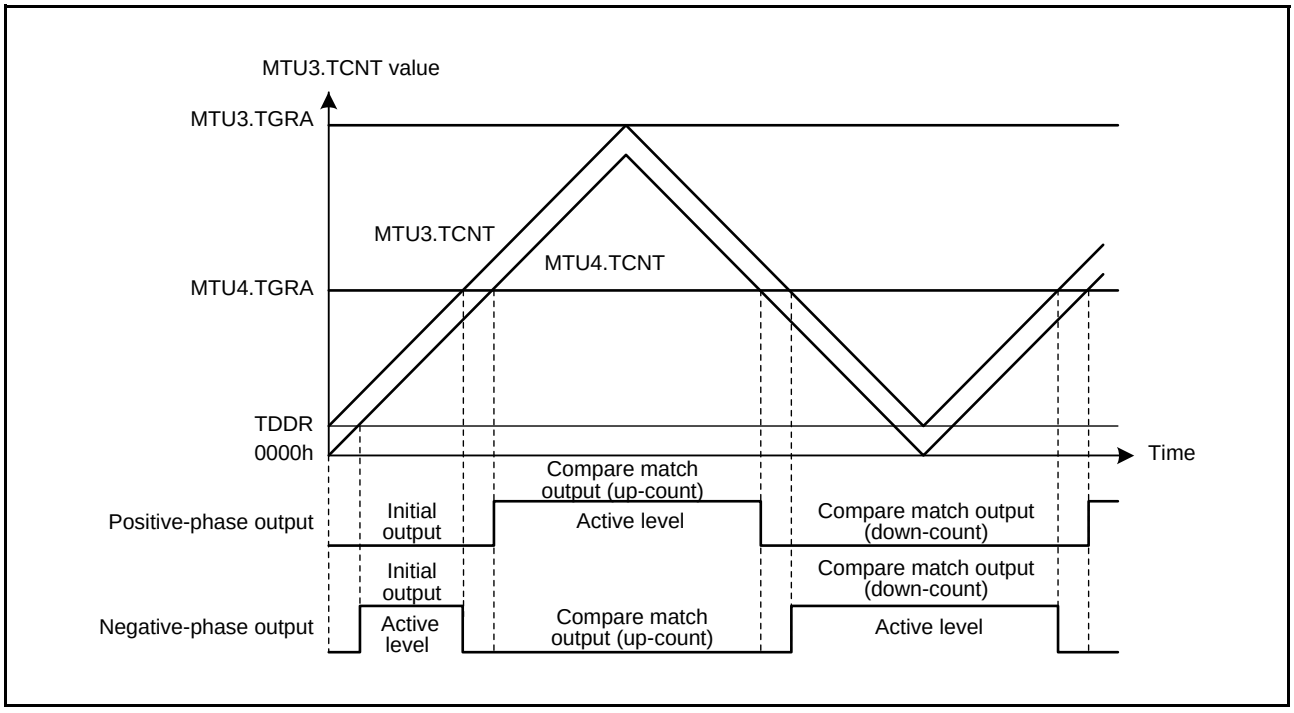


Figure 21.2 Example of Output in Complementary PWM Mode

21.2.19 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 21.32.	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 21.33.	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 21.34.	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 21.35.	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 21.36.	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 21.37.	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 21.38 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).

TOCR2 control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Table 21.32 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 21.33 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 21.34 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 21.35 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 21.36 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 21.37 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 21.38 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 the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from the buffer register (TOLBR) to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

21.2.20 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

TOLBR is 8-bit readable/writable registers that function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 21.3 shows an example of the PWM output level setting procedure in buffer operation.

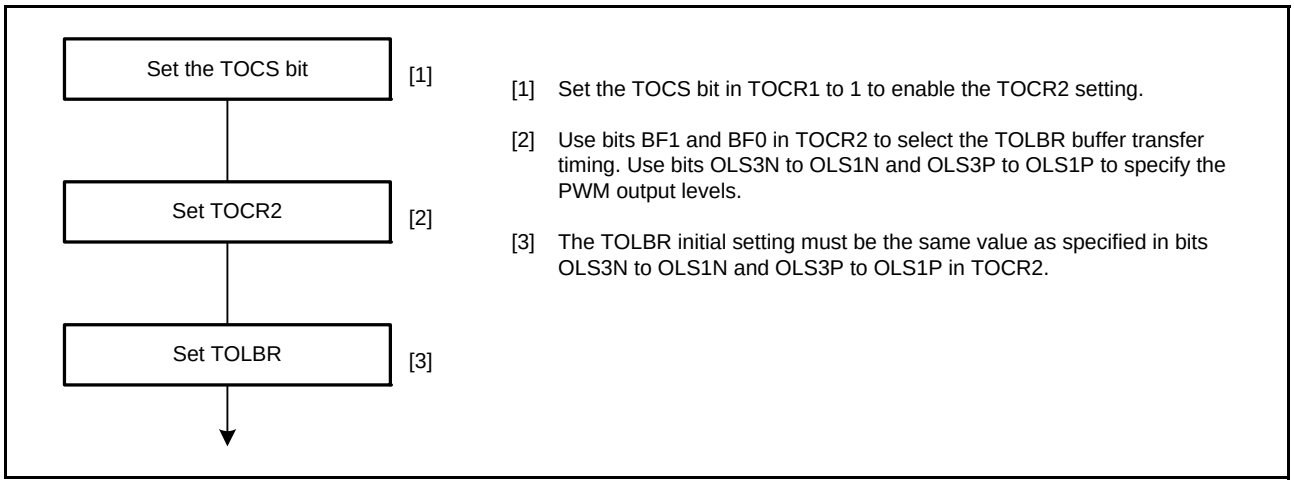


Figure 21.3 Example of PWM Output Level Setting Procedure in Buffer Operation

21.2.21 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 s set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 21.39.	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

TGCR 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 21.39.

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 21.39 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

21.2.22 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: • Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

TCNTS is 16-bit read-only counters that are used only in complementary PWM mode.
The TCNTS value after reset is 0000h.

21.2.23 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: • Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

TDDR is 16-bit registers, used only in complementary PWM mode, that specify the MTU3.TCNT and MTU4.TCNT counter offset value. 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. The TDDR value after reset is FFFFh.

21.2.24 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: • Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

TCDR is 16-bit registers used only in complementary PWM mode. Set half the PWM carrier 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). The TCDR value after reset is FFFFh.

21.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

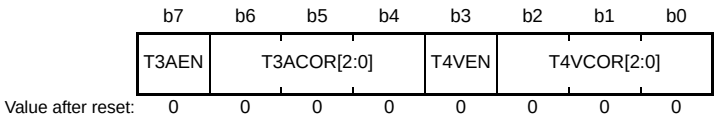
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: • Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

TCBR is 16-bit registers, used only in complementary PWM mode, that function as buffer registers for TCDR. The TCBR value is transferred to TCDR with the transfer timing set in TMDR. The TCBR value after reset is FFFFh.

21.2.26 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 21.40.	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 21.41.	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 timer interrupt skipping counter (TITCNT).

T4VCOR[2:0] Bits (TCIV4 Interrupt Skipping Count Setting)
T3ACOR[2:0] Bits (TGIA3 Interrupt Skipping Count Setting)

These bits specify the TCIV3 and TGIA4 interrupt skipping count within the range from 0 to 7. For details, see Table 21.40 and Table 21.41.

Table 21.40 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 21.41 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.

21.2.27 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 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 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.

TITCNT is 8-bit readable counters. 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

21.2.28 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 21.42 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTER is 8-bit readable/writable registers that 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 21.42 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 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 21.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) 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 BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

21.2.29 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.

TDER is 8-bit readable/writable registers that control 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 TDER bit = 1

21.2.30 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.

TWCR is 8-bit readable/writable registers. TWCR controls 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 CCE bit and WRE bit in TWCR 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 21.40.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE bit = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to the CCE bit after reading CCE bit = 0

21.2.31 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 NFCRs for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

MTUn.NFCR is 8-bit readable and writable register (n = 0 to 4). These registers control enabling and disabling of 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.

21.2.32 Bus Master Interface

The timer counters (TCNT), 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/B), and timer A/D converter start request cycle set buffer registers (TADCOBRA/B) 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. Always 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.

21.3 Operation

21.3.1 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 21.4 shows an example of the count operation setting procedure.

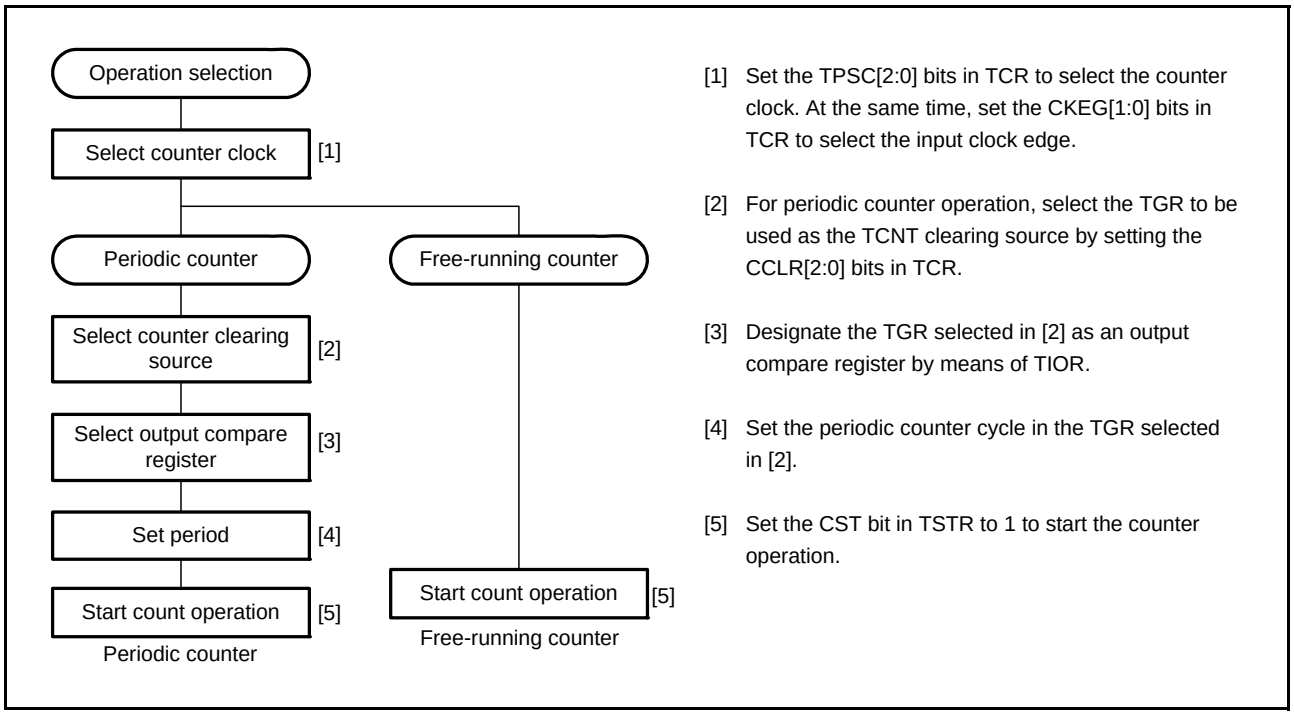


Figure 21.4 Example of Counter Operation Setting Procedure

(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 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 21.5 illustrates free-running counter operation.

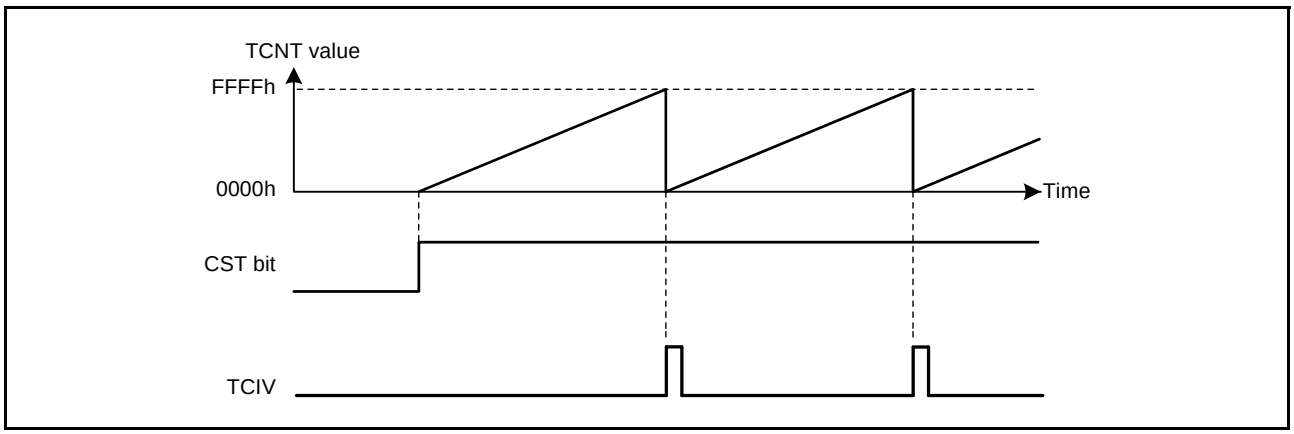


Figure 21.5 Free-Running Counter Operation

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 bits CCLR[2:0] in TCR. 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 cleared to 0000h.
If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.
Figure 21.6 illustrates periodic counter operation.

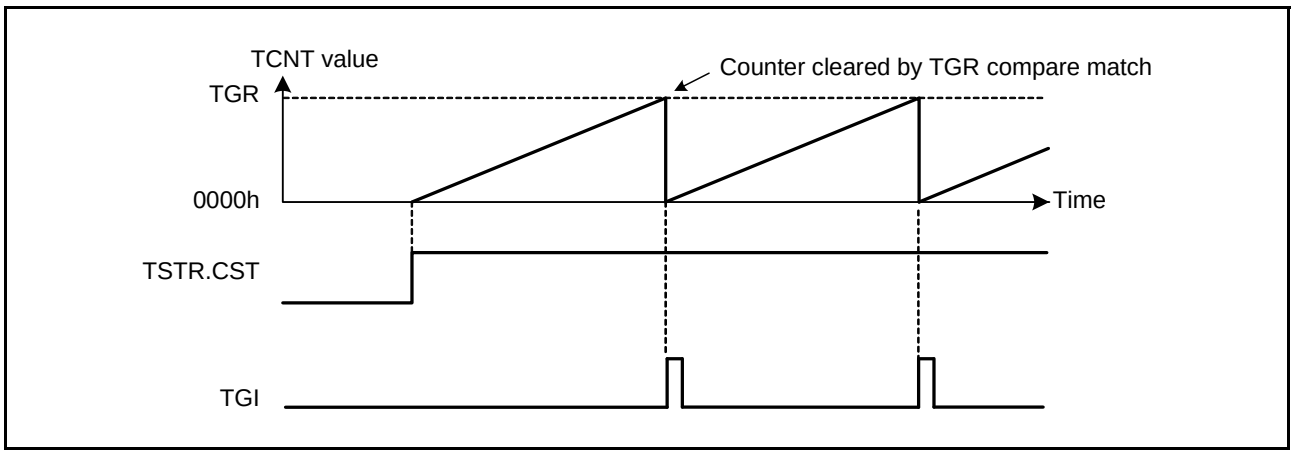


Figure 21.6 Periodic Counter Operation

(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 21.7 shows an example of the procedure for setting waveform output by compare match.

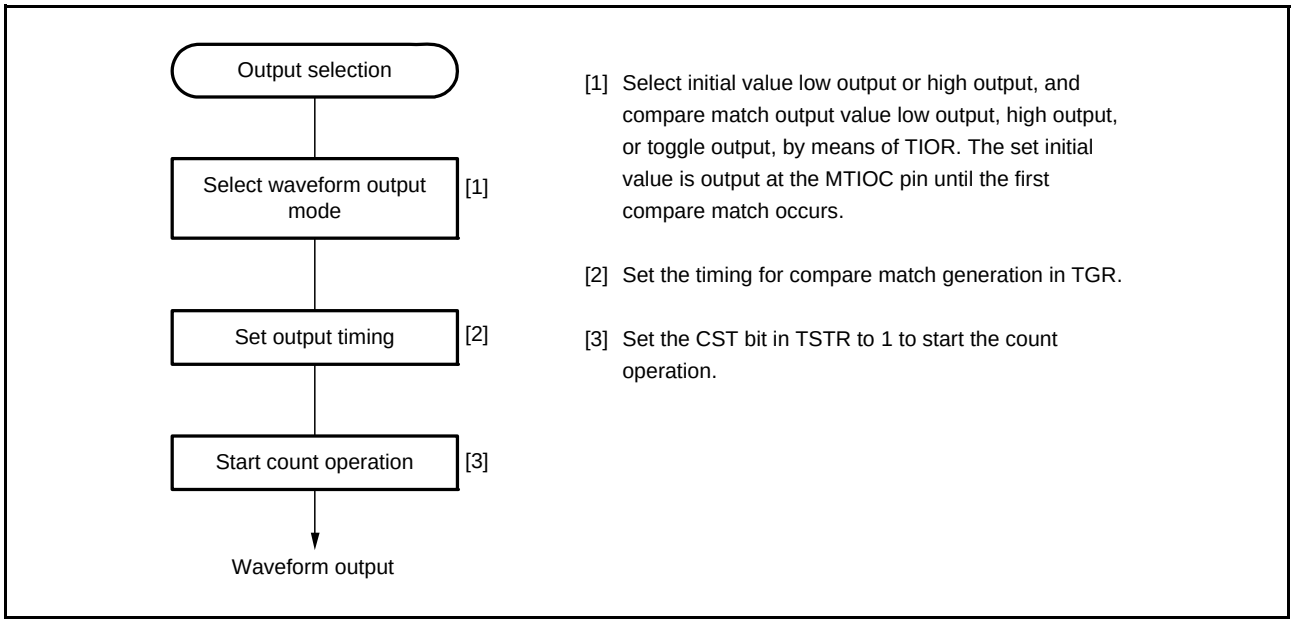


Figure 21.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 21.8 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.

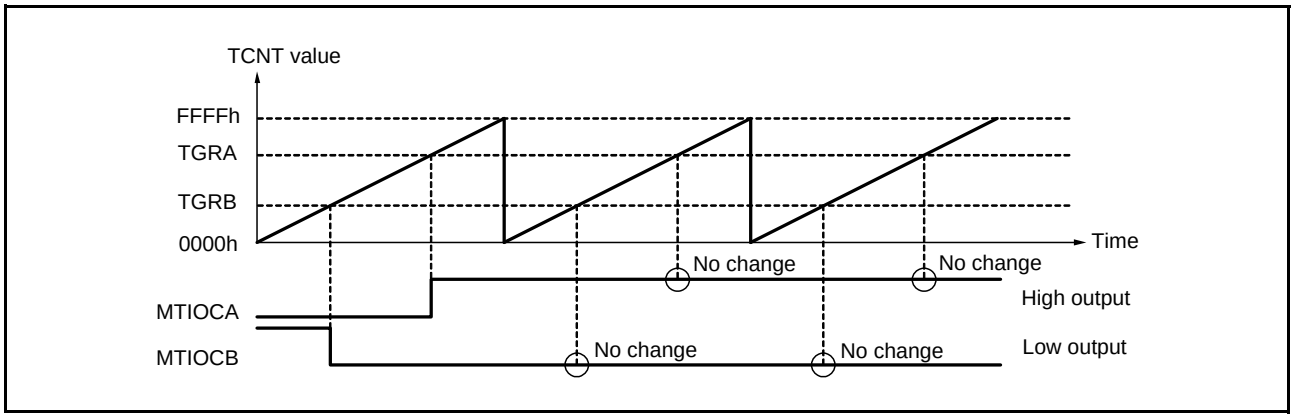


Figure 21.8 Example of Low Output and High Output Operation

Figure 21.9 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.

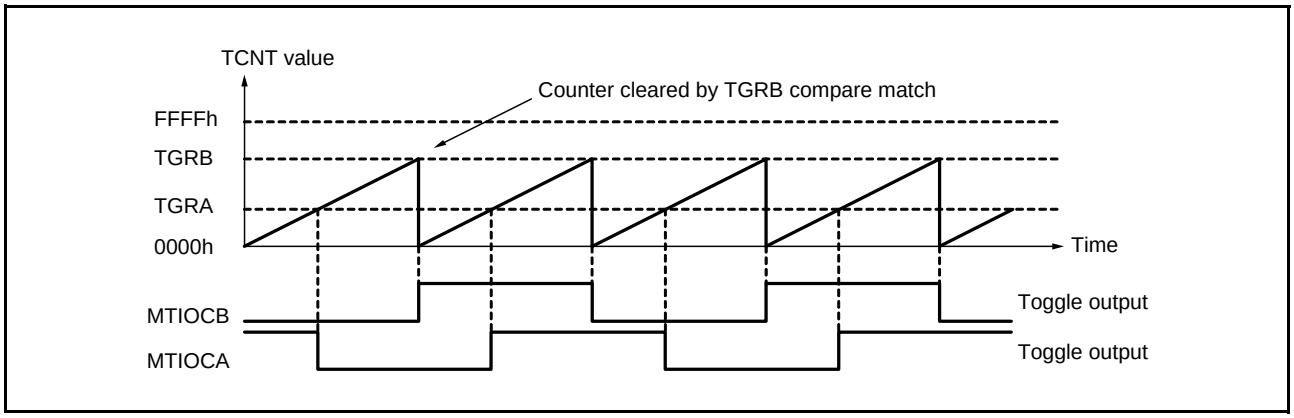


Figure 21.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOC pin input edge.

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 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 21.10 shows an example of the input capture operation setting procedure.

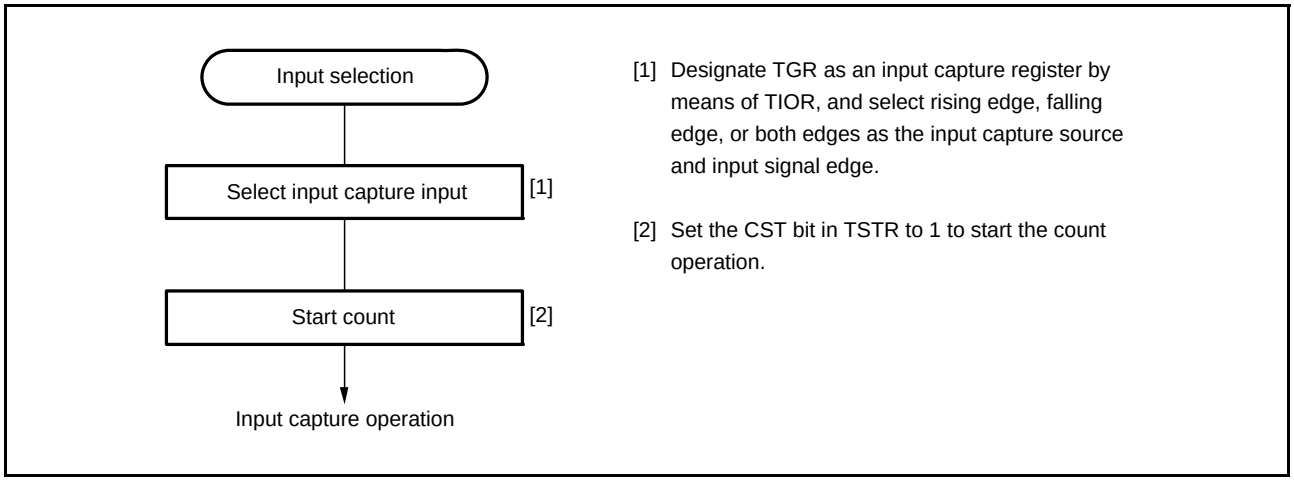


Figure 21.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 21.11 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.

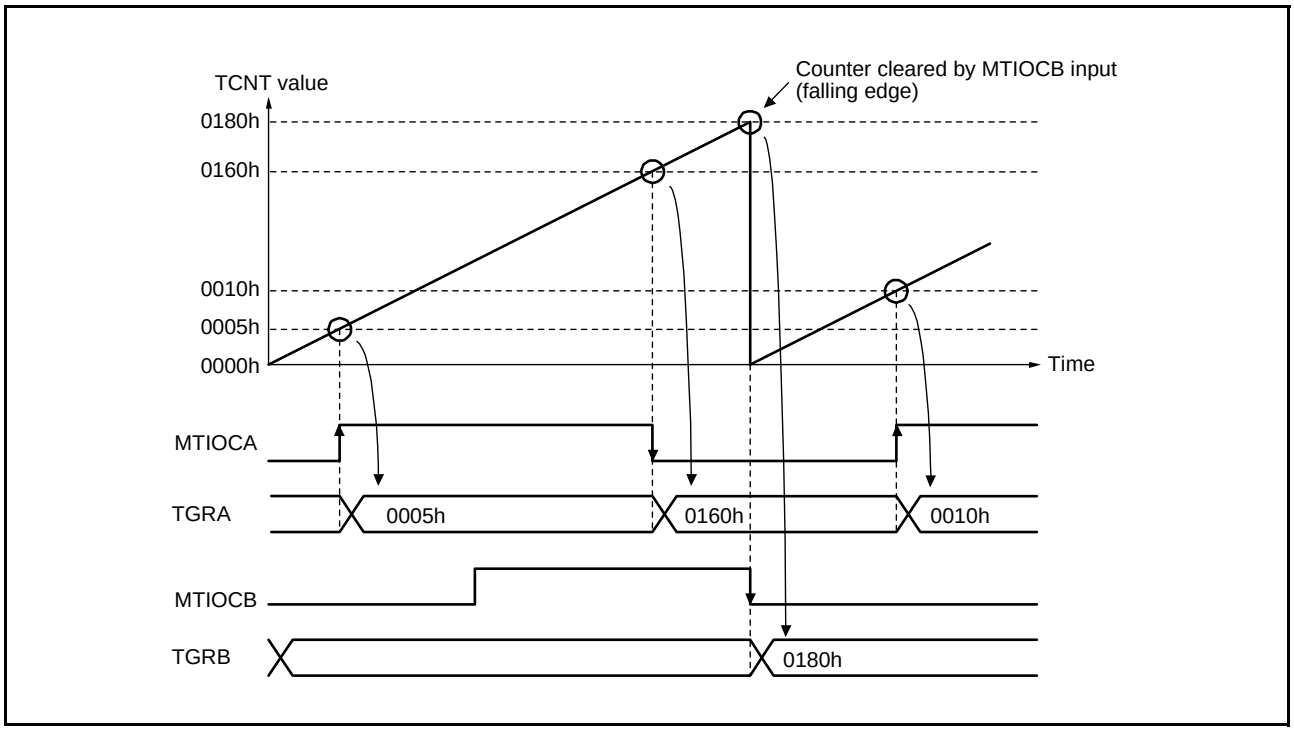


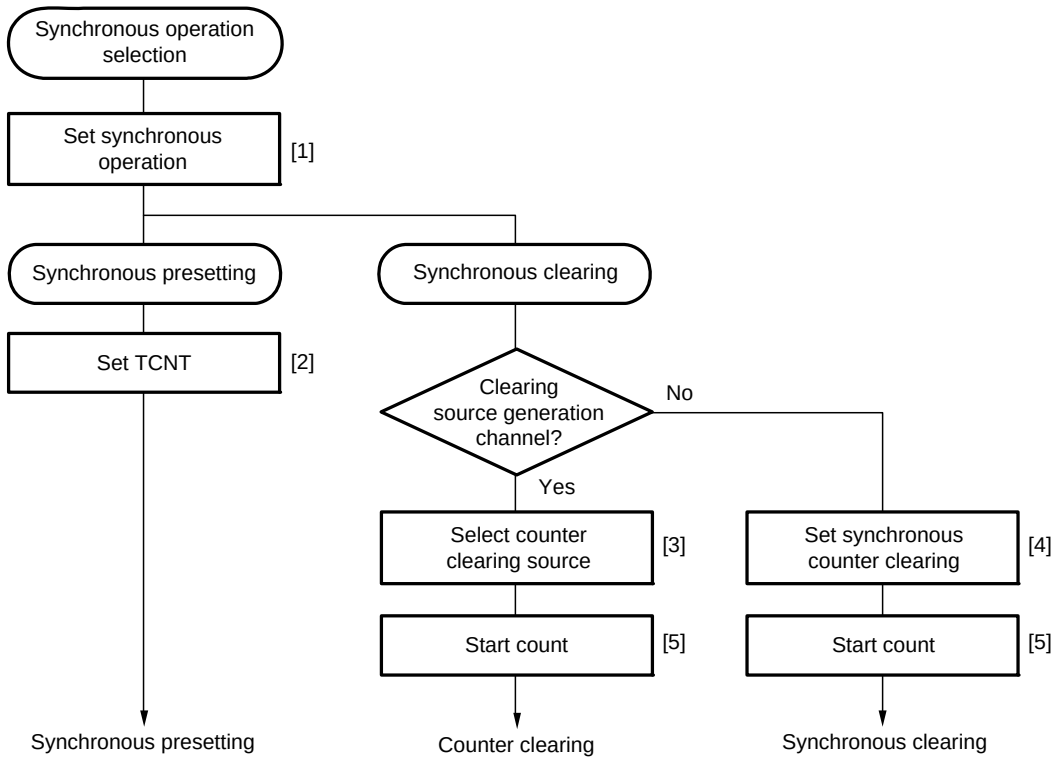
Figure 21.11 Example of Input Capture Operation

21.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). 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 21.12 shows an example of the synchronous operation setting procedure.



[1] Set 1 in the SYNC bit 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 21.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 21.13 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 presetting 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, see section 21.3.5, PWM Modes.

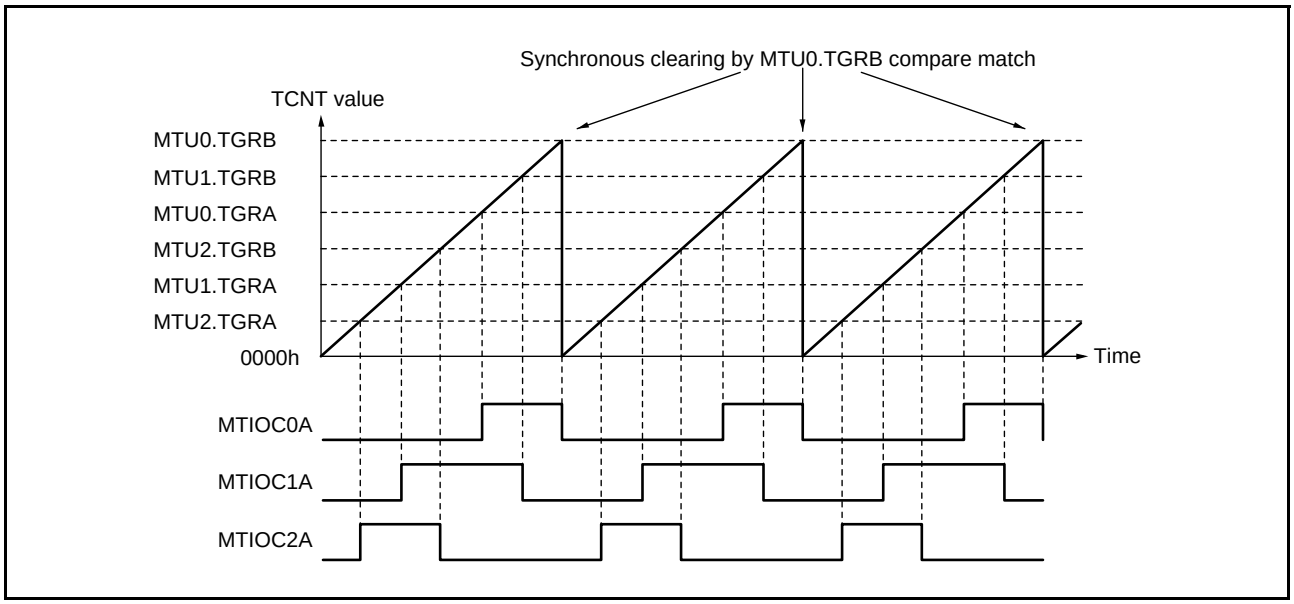


Figure 21.13 Example of Synchronous Operation

21.3.3 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 21.43 shows the register combinations used in buffer operation.

Table 21.43 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 21.14.

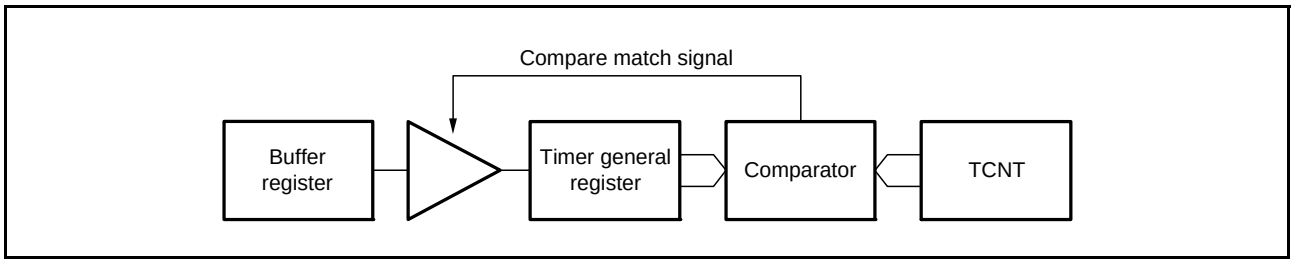


Figure 21.14 Compare Match Buffer Operation

- 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 21.15.

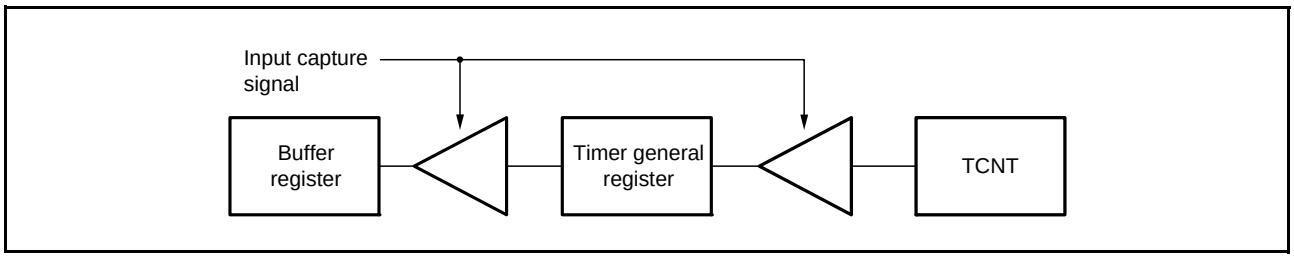


Figure 21.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 21.16 shows an example of the buffer operation setting procedure.

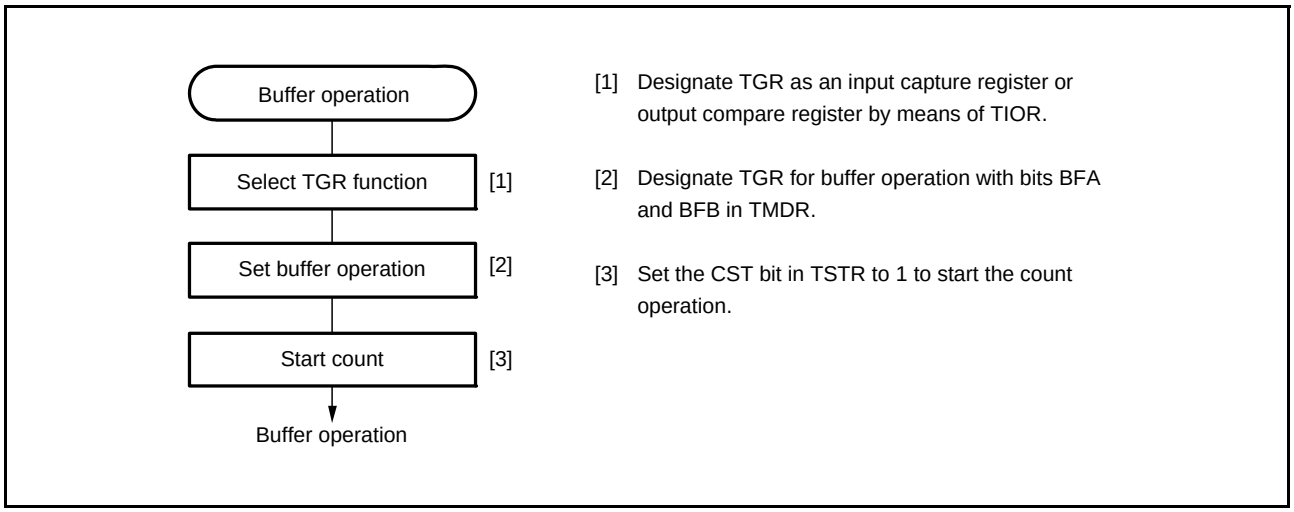


Figure 21.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 21.17 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 TTSA bit in TBTM is cleared 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, see section 21.3.5, PWM Modes.

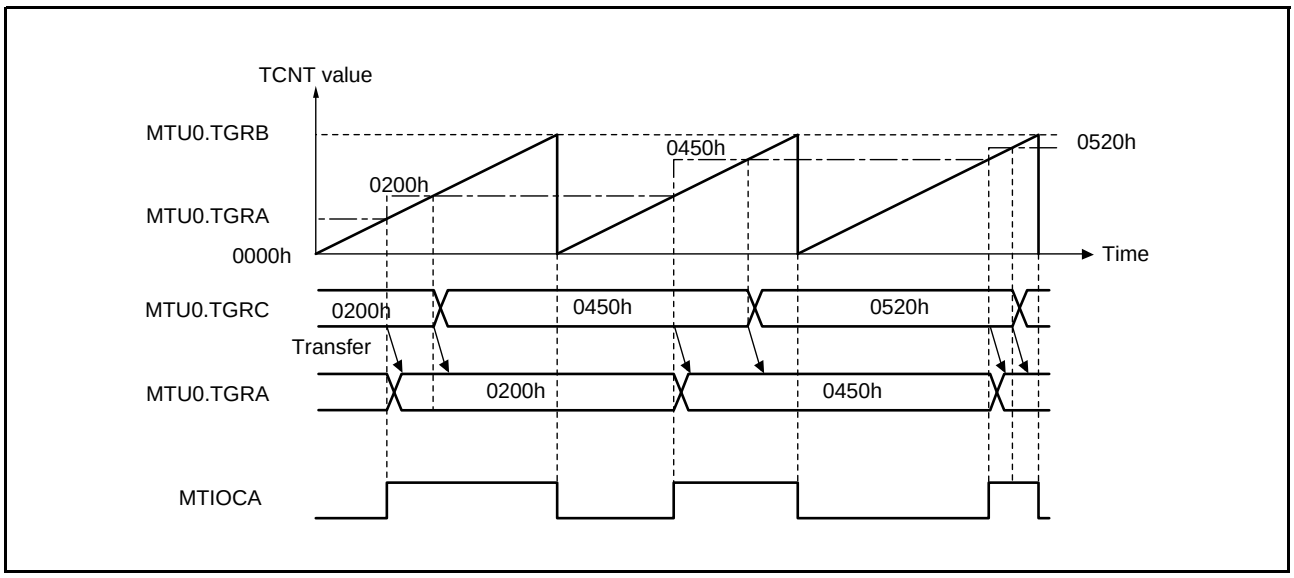


Figure 21.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 21.18 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 stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

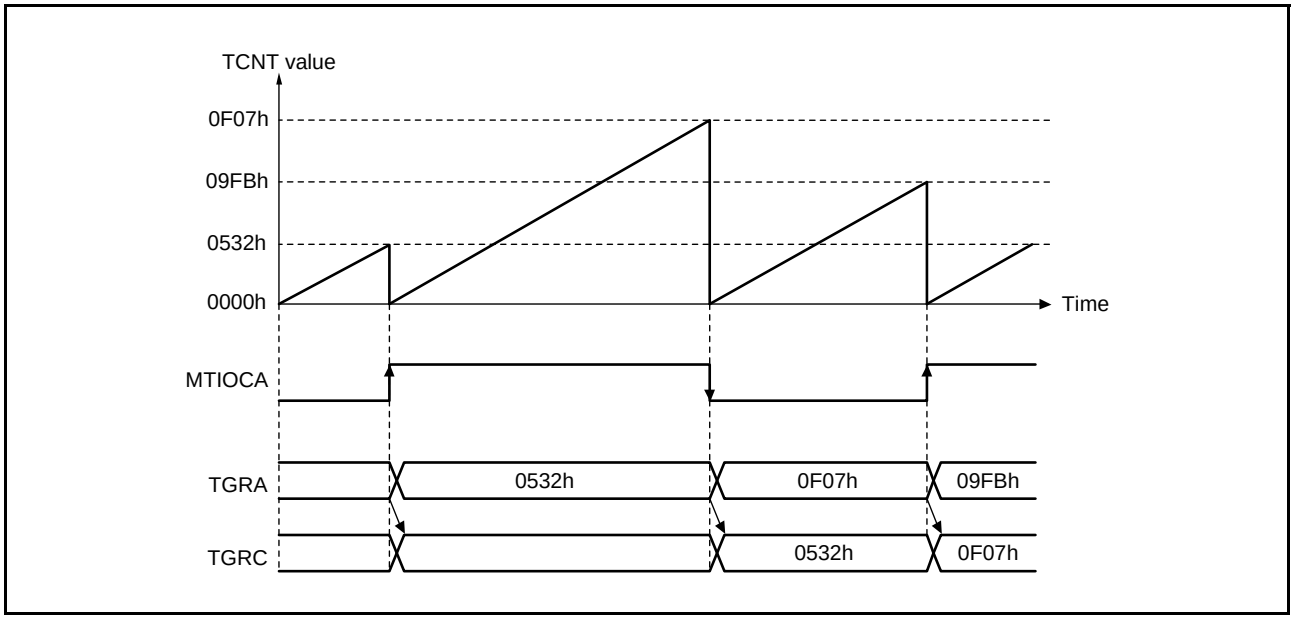


Figure 21.18 Example of Buffer Operation (2)

(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 cleared to 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: • TBTM must be modified only while TCNT stops.

Figure 21.19 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 TTSA bit in MTU0.TBTM is set to 1.

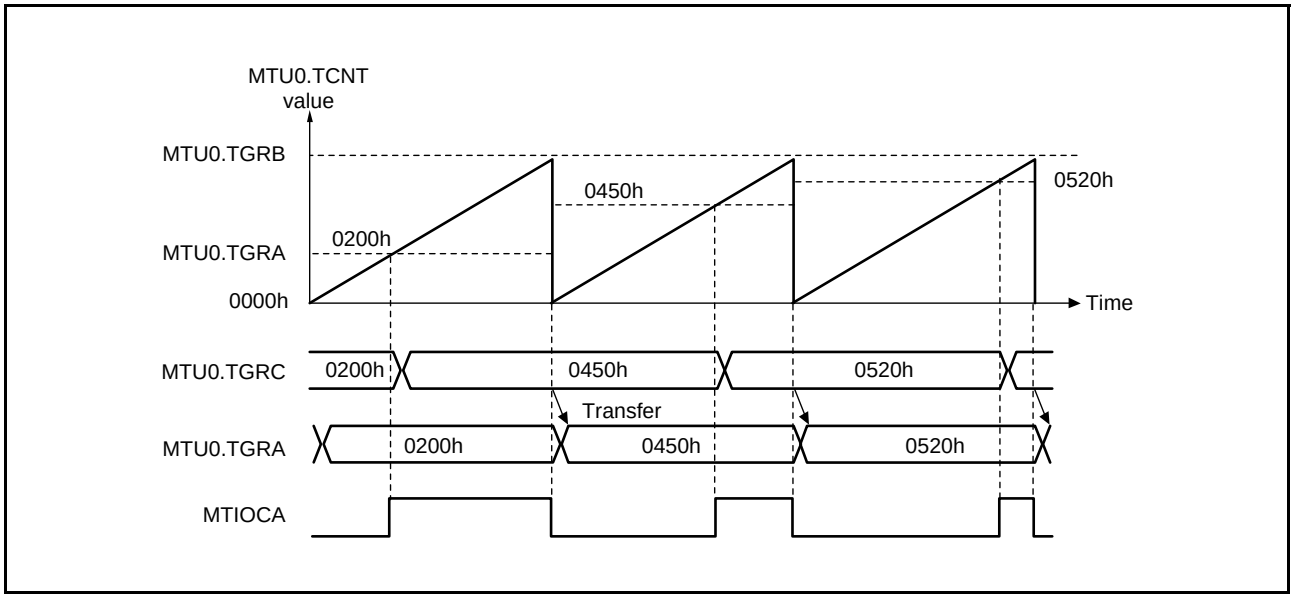


Figure 21.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing

21.3.4 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 counter clock for MTU1 through the TPSC[2:0] bits in TCR.
Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.
Table 21.44 lists the register combinations used in cascaded operation.

Note: • When phase counting mode is set for MTU1 or MTU2, the counter clock setting is invalid and the counters operate independently in phase counting mode.

Table 21.44 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 timer input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high, 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 21.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 21.45 lists the TICCR setting and input capture input pins.

Table 21.45 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 21.20 shows an example of the cascaded operation setting procedure.

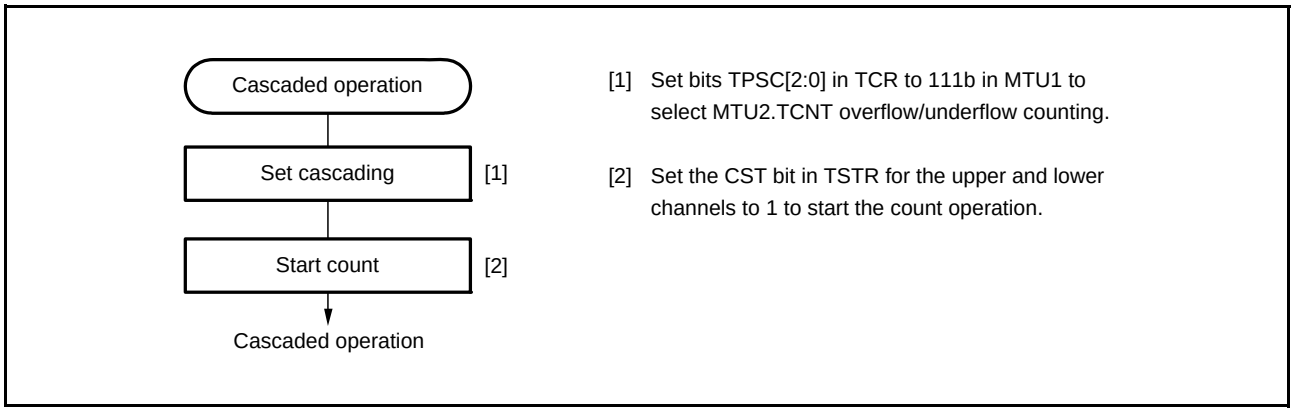


Figure 21.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 21.21 illustrates the operation when MTU2.TCNT overflow/underflow counting has been set for MTU1.TCNT and phase counting mode has been designated for MTU2.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

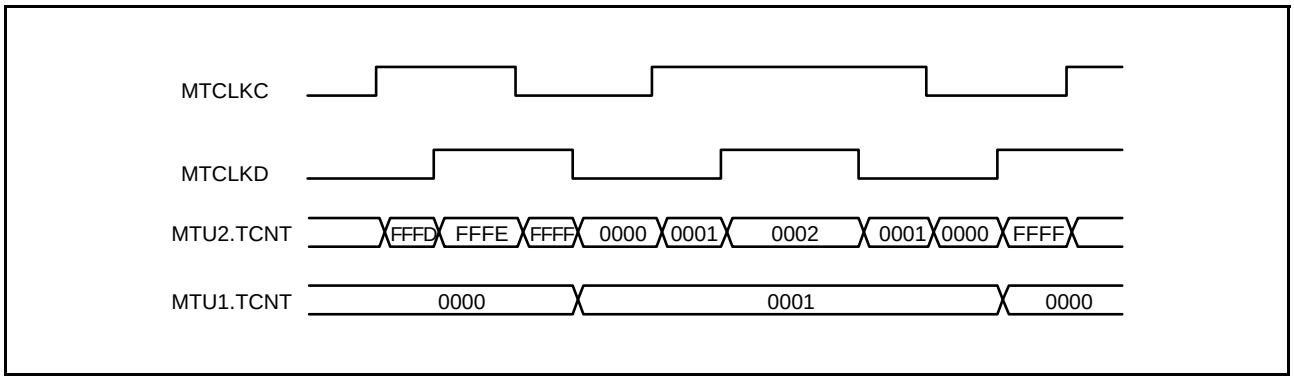


Figure 21.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 21.22 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA3 to IOA0 bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA3 to IOA0 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.

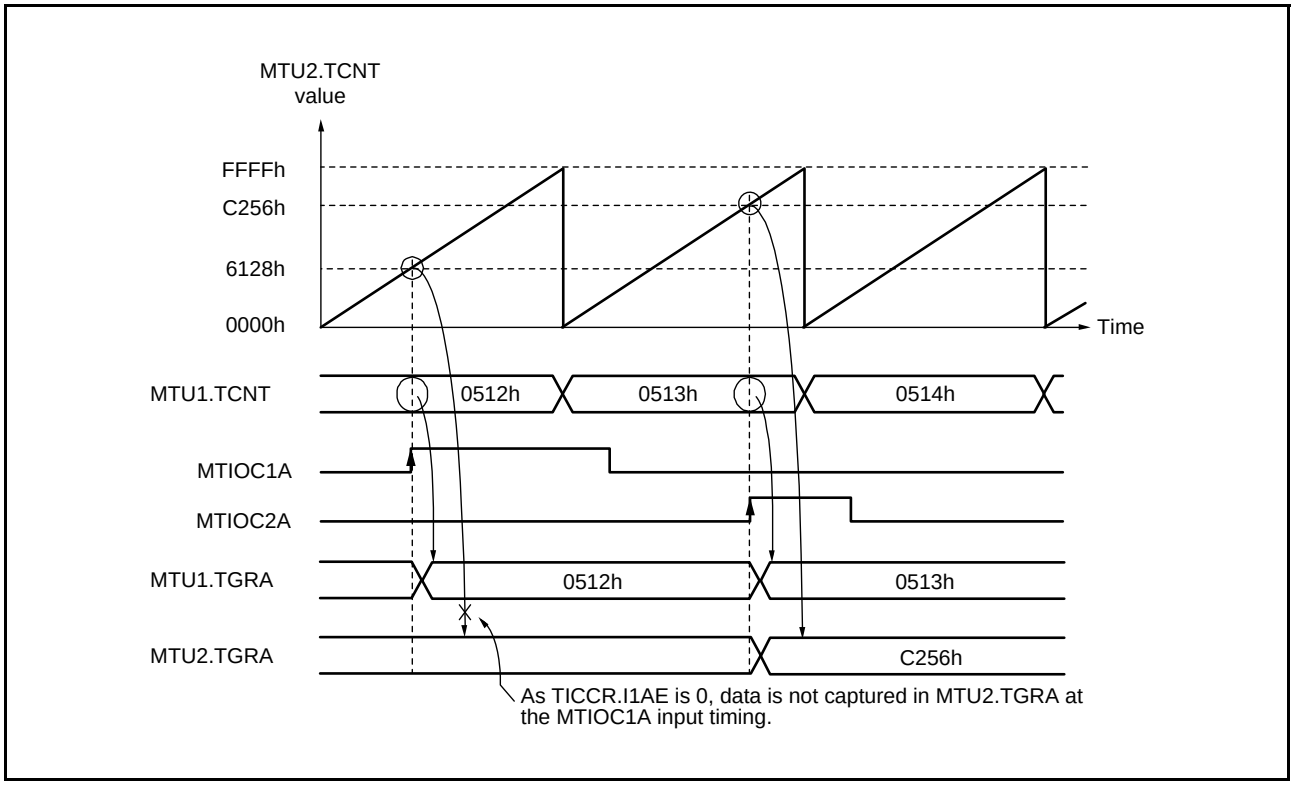
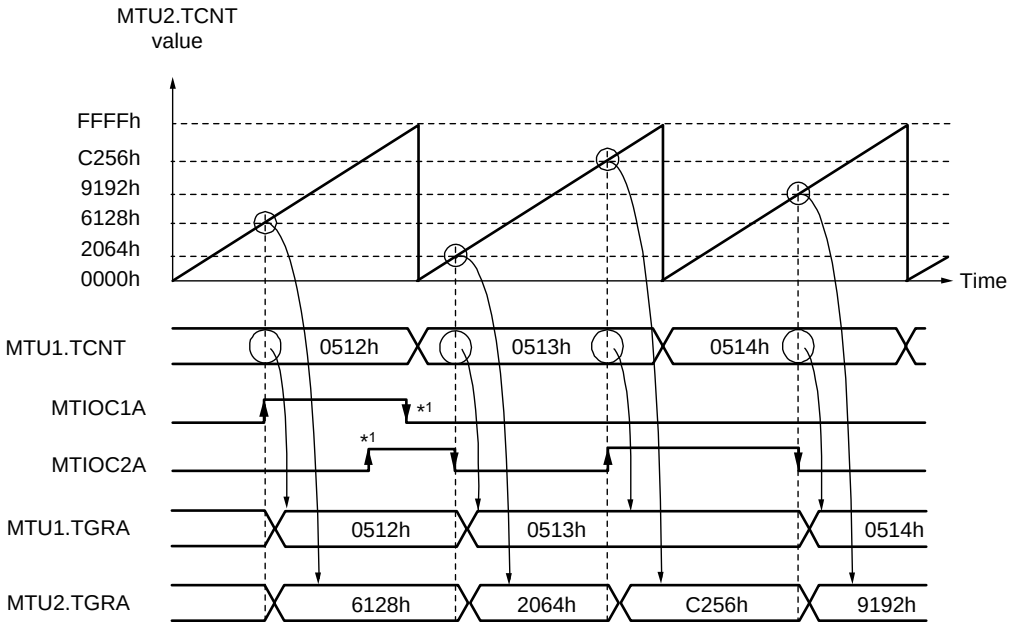


Figure 21.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 21.23 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 IOA3 to IOA0 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 21.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 21.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA3 to IOA0 bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA3 to IOA0 bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

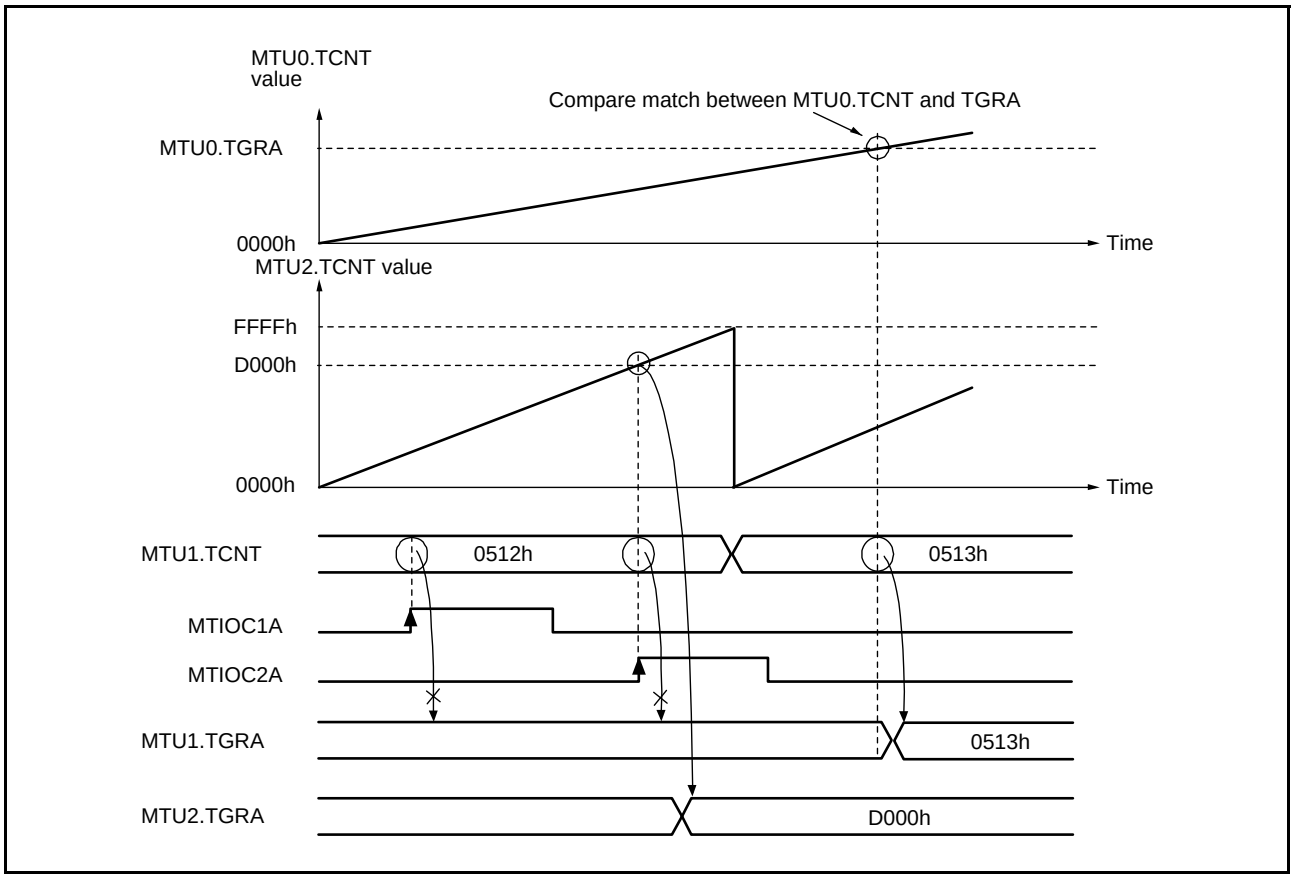


Figure 21.24 Cascaded Operation Example (d)

21.3.5 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. Synchronous operation is also possible.
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 bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the levels specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR 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 synchronized 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 21.46.

Table 21.46 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 21.25 shows an example of the PWM mode setting procedure.

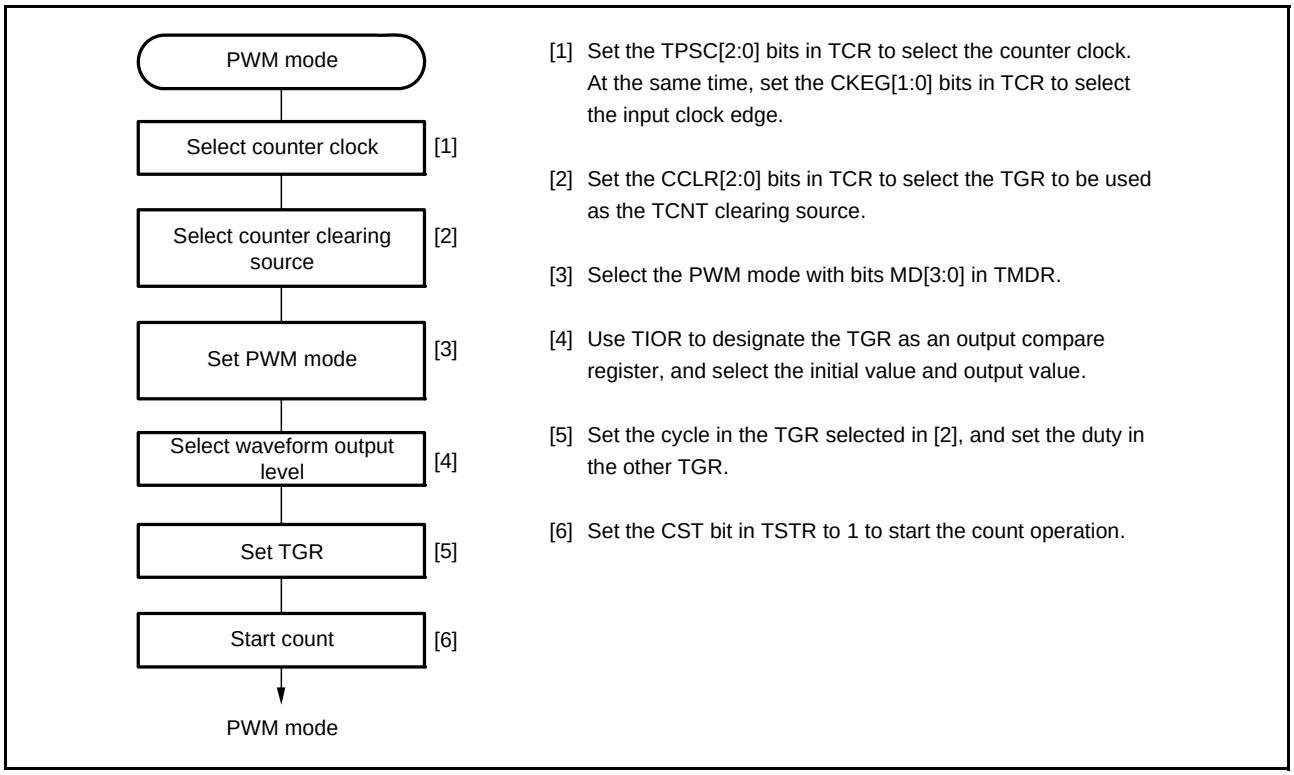


Figure 21.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 21.26 shows an example of operation in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, low is set as the initial output value and output value for TGRA, and high 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.

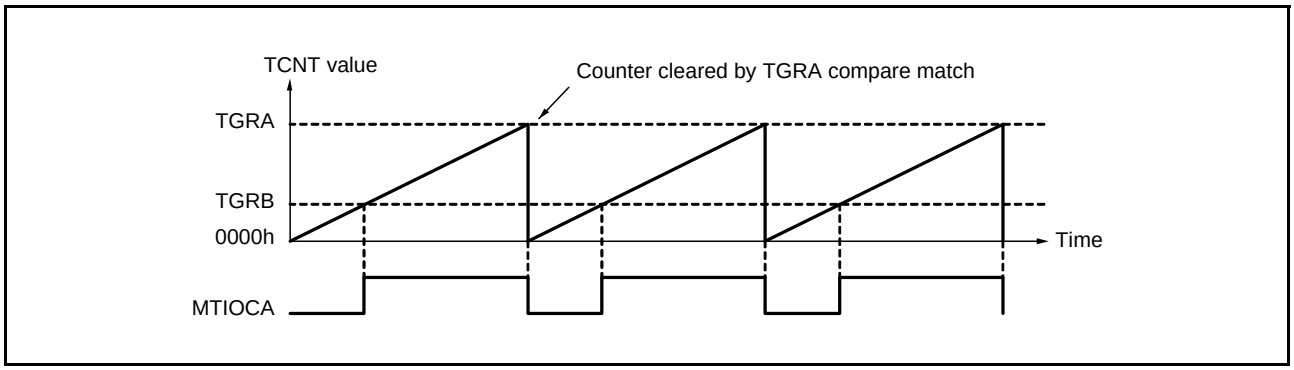


Figure 21.26 Example of PWM Mode Operation

Figure 21.27 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 Low is set as the initial output value and High as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.
In this case, the value set in MTU1.TGRB is used as the cycle, and the values set in the other TGRs are used as the duty.

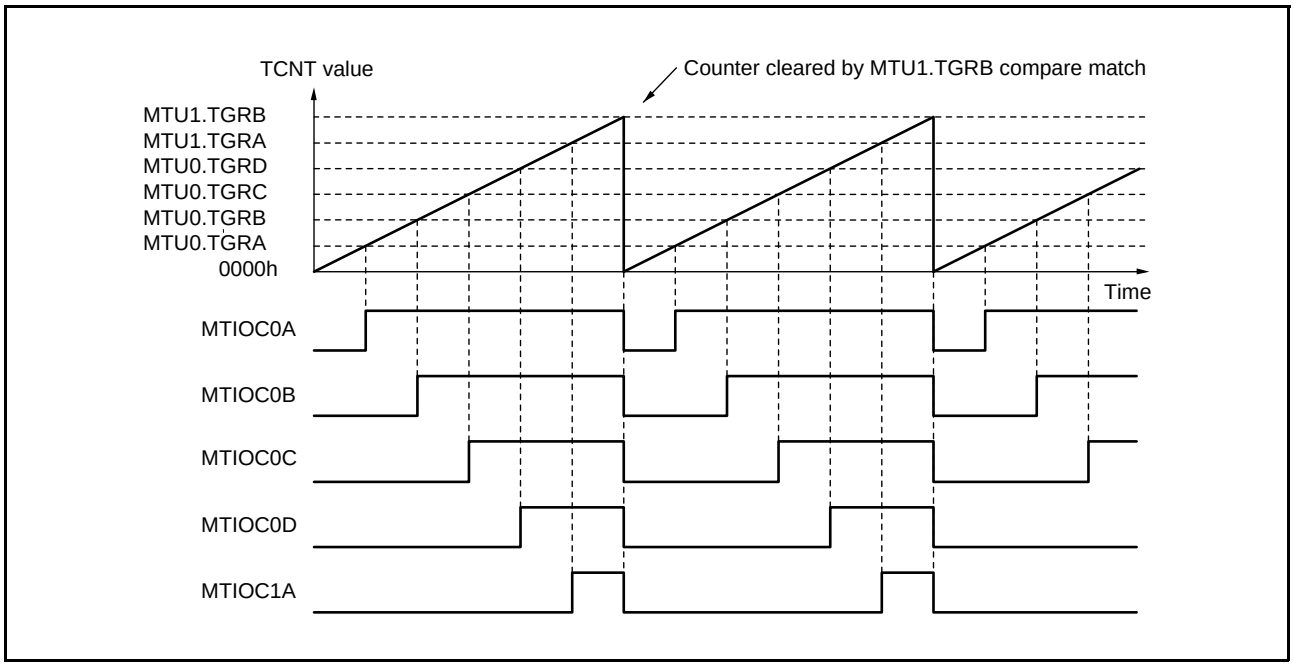


Figure 21.27 Example of PWM Mode Operation

Figure 21.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

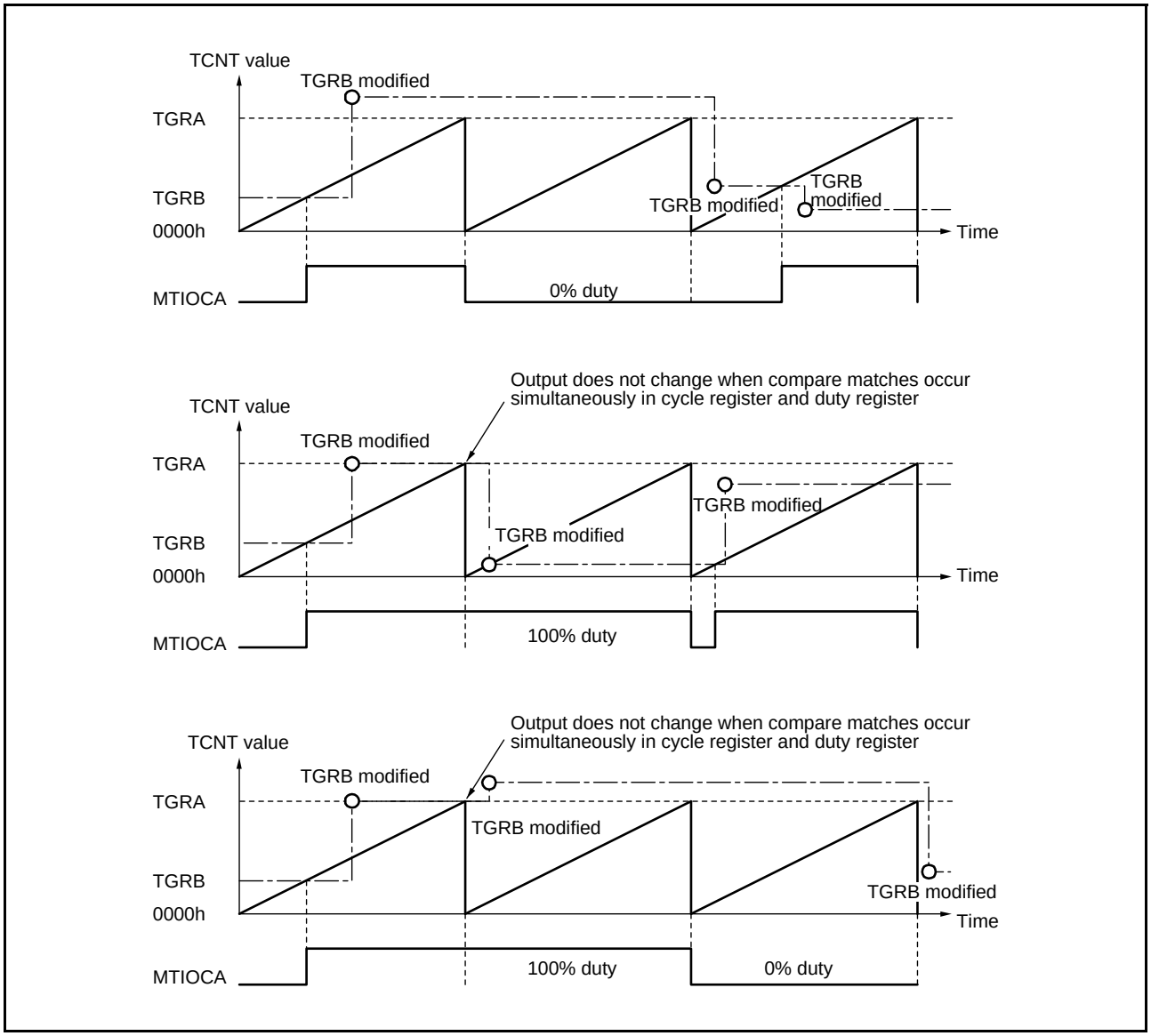


Figure 21.28 Examples of PWM Mode Operation

21.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external input clocks is detected and TCNT is incremented or decremented accordingly. This mode can be set for MTU1 and MTU2.

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC[2:0] and bits CKEG[1:0] in TCR. However, the functions of bits CCLR[1:0] in TCR and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the TCIEV bit in the corresponding TIER is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the TCIEU bit in the corresponding TIER is 1.

The TCFD bit in TSR is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.

Table 21.47 lists the correspondence between external clock pins and channels.

Table 21.47 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 21.29 shows an example of the phase counting mode setting procedure.

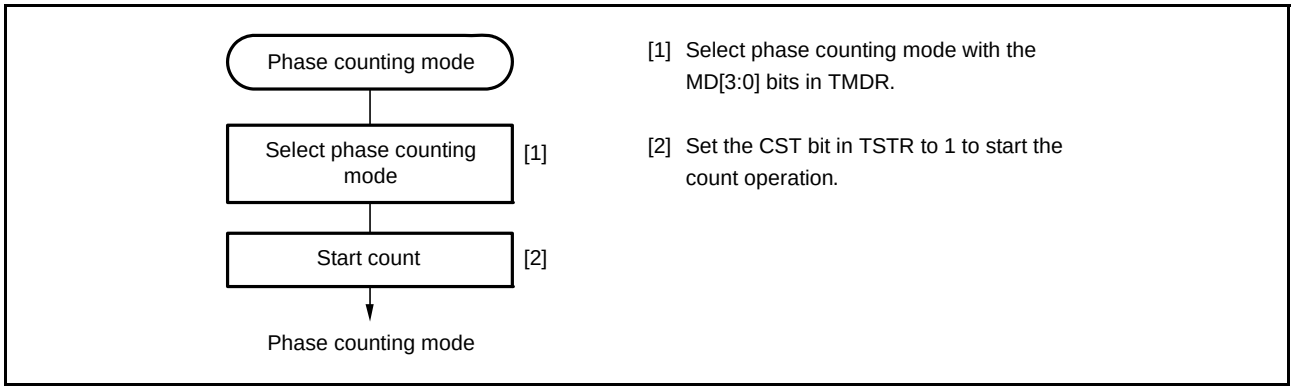


Figure 21.29 Example of Phase Counting Mode Setting Procedure

(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 21.30 shows an example of operation in phase counting mode 1, and Table 21.48 summarizes the TCNT up/down-count conditions.

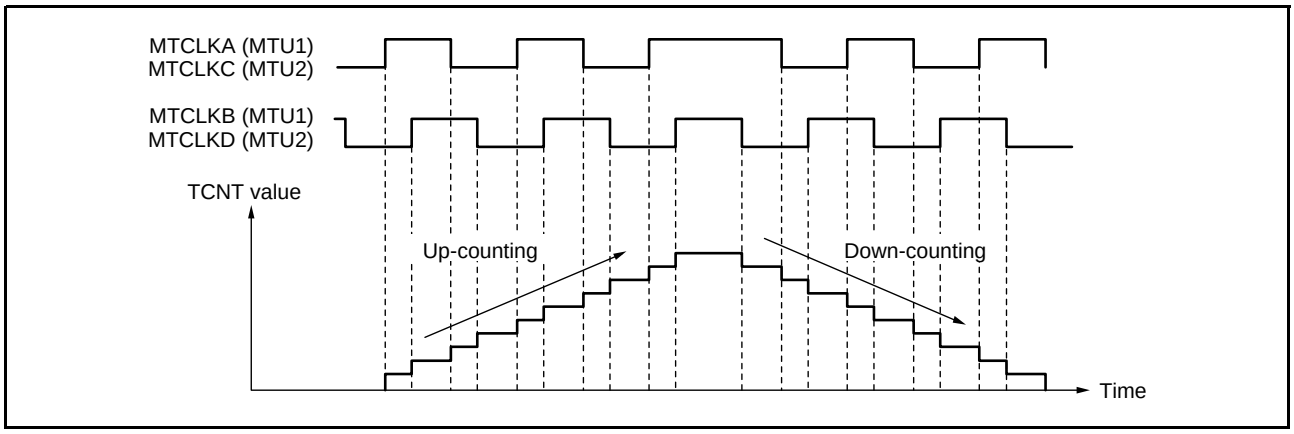


Figure 21.30 Example of Operation in Phase Counting Mode 1

Table 21.48 Up/Down-Count 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 21.31 shows an example of operation in phase counting mode 2, and Table 21.49 summarizes the TCNT up/down-count conditions.

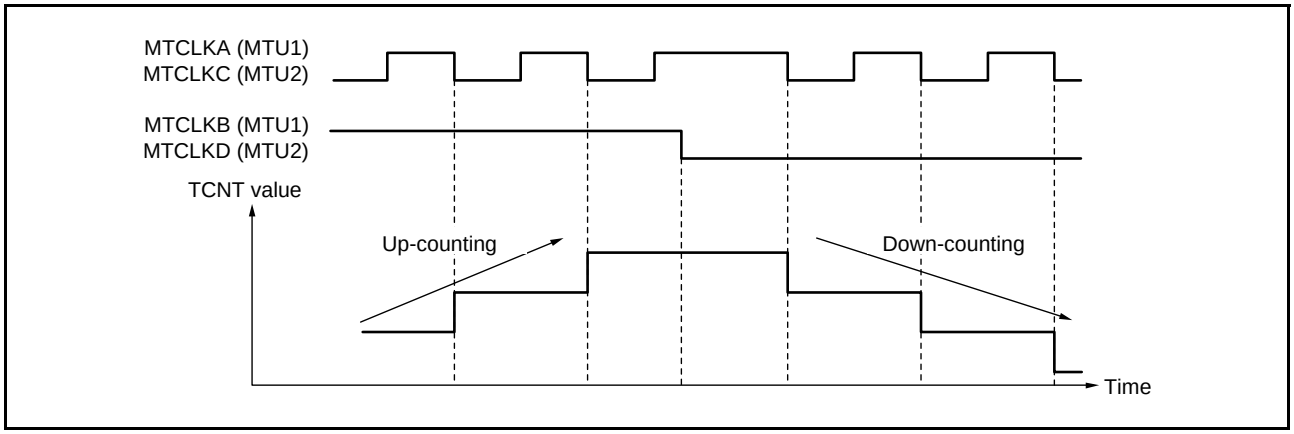


Figure 21.31 Example of Operation in Phase Counting Mode 2

Table 21.49 Up/Down-Count 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 21.32 shows an example of operation in phase counting mode 3, and Table 21.50 summarizes the TCNT up/down-count conditions.

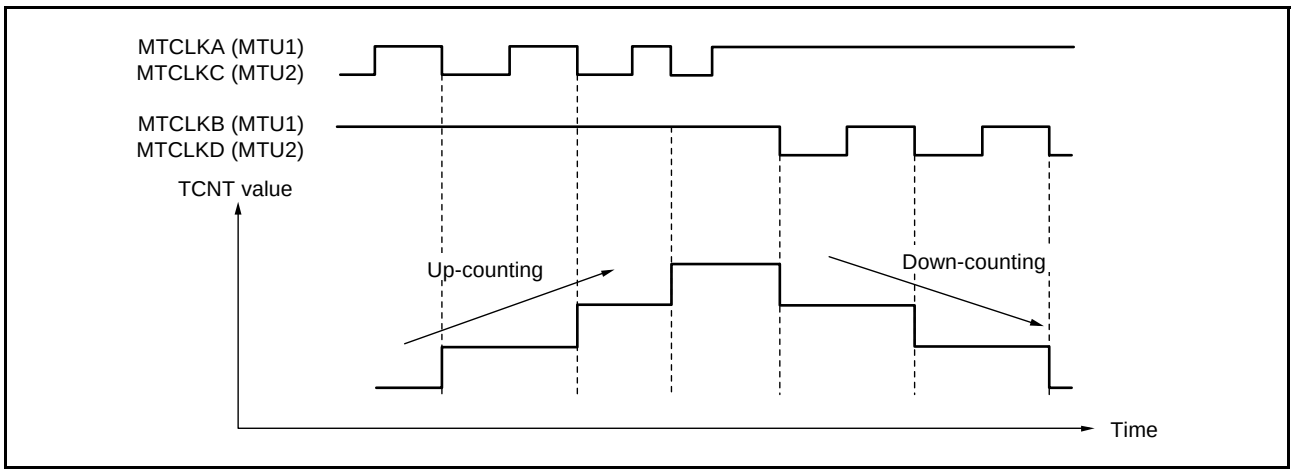


Figure 21.32 Example of Operation in Phase Counting Mode 3

Table 21.50 Up/Down-Count 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 21.33 shows an example of operation in phase counting mode 4, and Table 21.51 summarizes the TCNT up/down-count conditions.

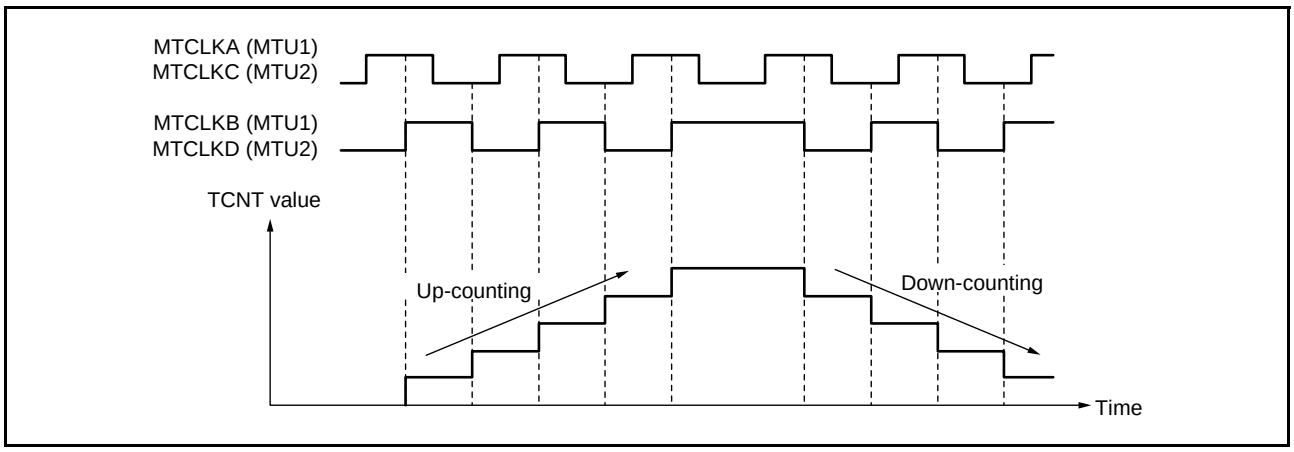


Figure 21.33 Example of Operation in Phase Counting Mode 4

Table 21.51 Up/Down-Count 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 21.34 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.

In MTU0, MTU0.TGRC compare match is specified as the 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 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 TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control cycles. This procedure enables the accurate detection of position and speed.

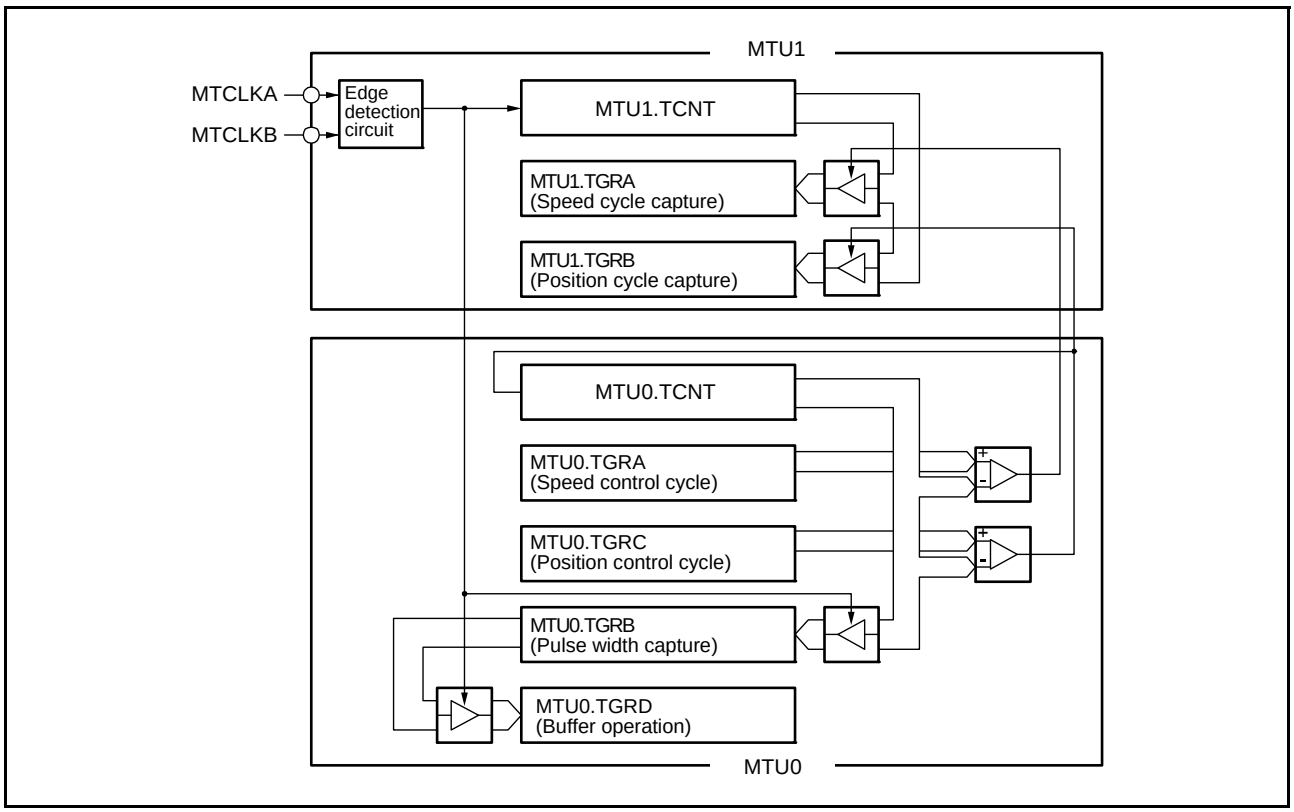


Figure 21.34 Phase Counting Mode Application Example

21.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three 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 timer counter 3 (MTU3.TCNT) functions as an up-counter.
Table 21.52 lists the PWM output pins. Table 21.53 lists the settings of the registers.

Table 21.52 Output Pins for Reset-Synchronized PWM Mode

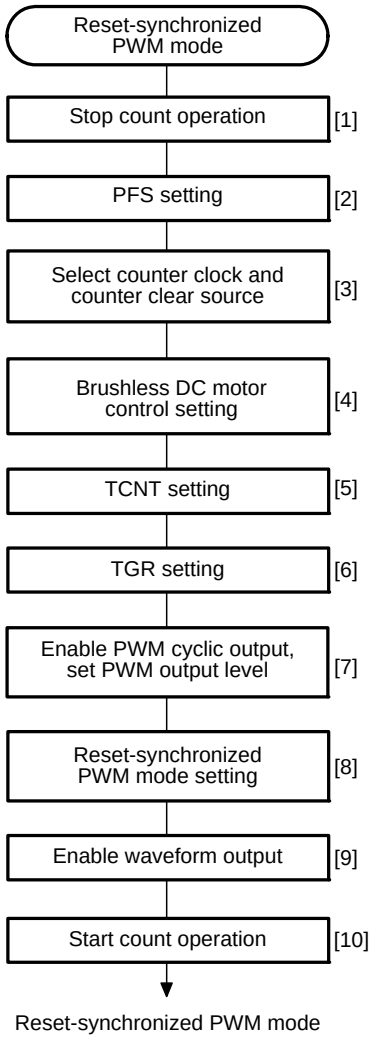
Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 21.53 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 21.35 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 counter 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 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 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.

$X \leq$ MTU3.TGRA (X: Setting value)

[7] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE 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 21.3.

[8] Set bits MD[3:0] 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 21.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 21.36 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.

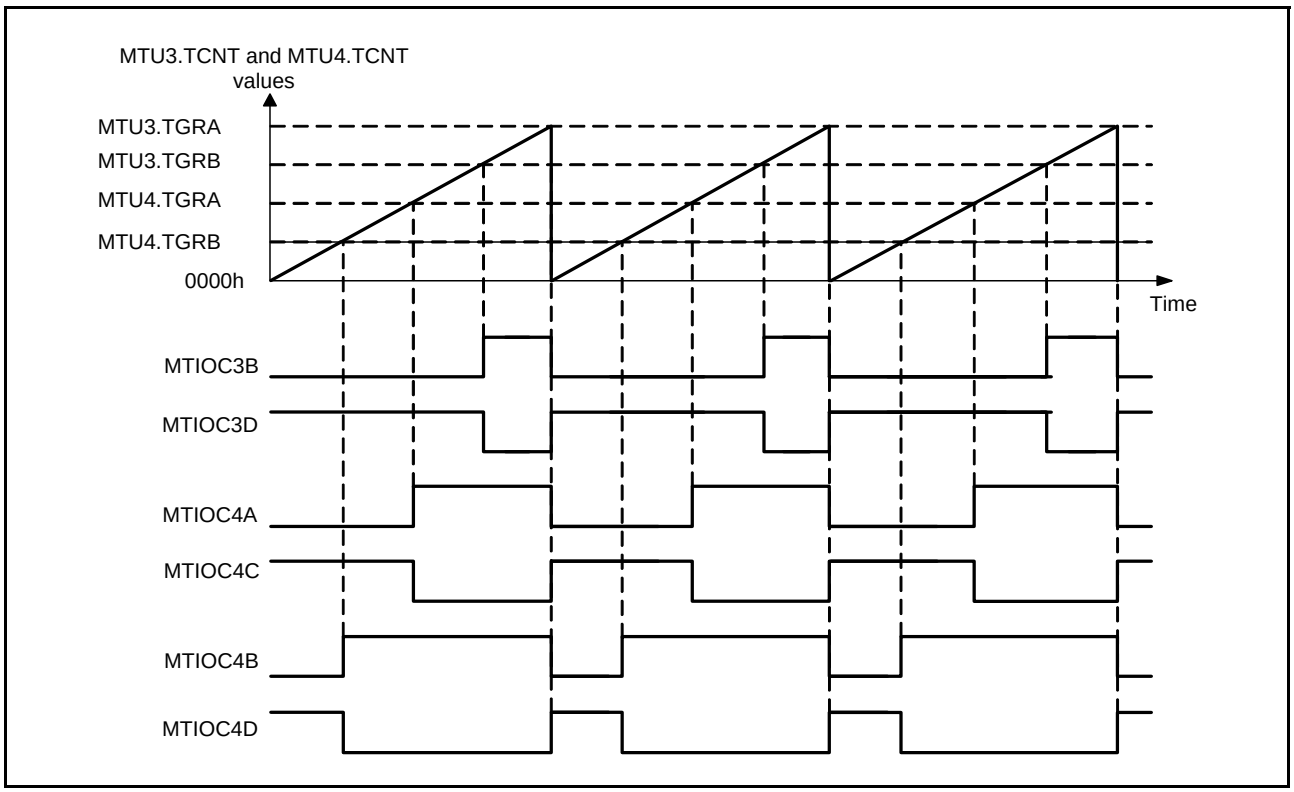


Figure 21.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1's OLSN = 1 and OLSP = 1)

21.3.8 Complementary PWM Mode

In complementary PWM mode, three phases of non-overlapping positive and negative PWM waveforms can be output by combining MTU3 and MTU4. PWM waveforms without non-overlapping interval are also available.
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 21.54 lists the PWM output pins used. Table 21.55 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 21.54 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' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 21.55 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	Always readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	Always 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	Always readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	Always 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	Always 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).

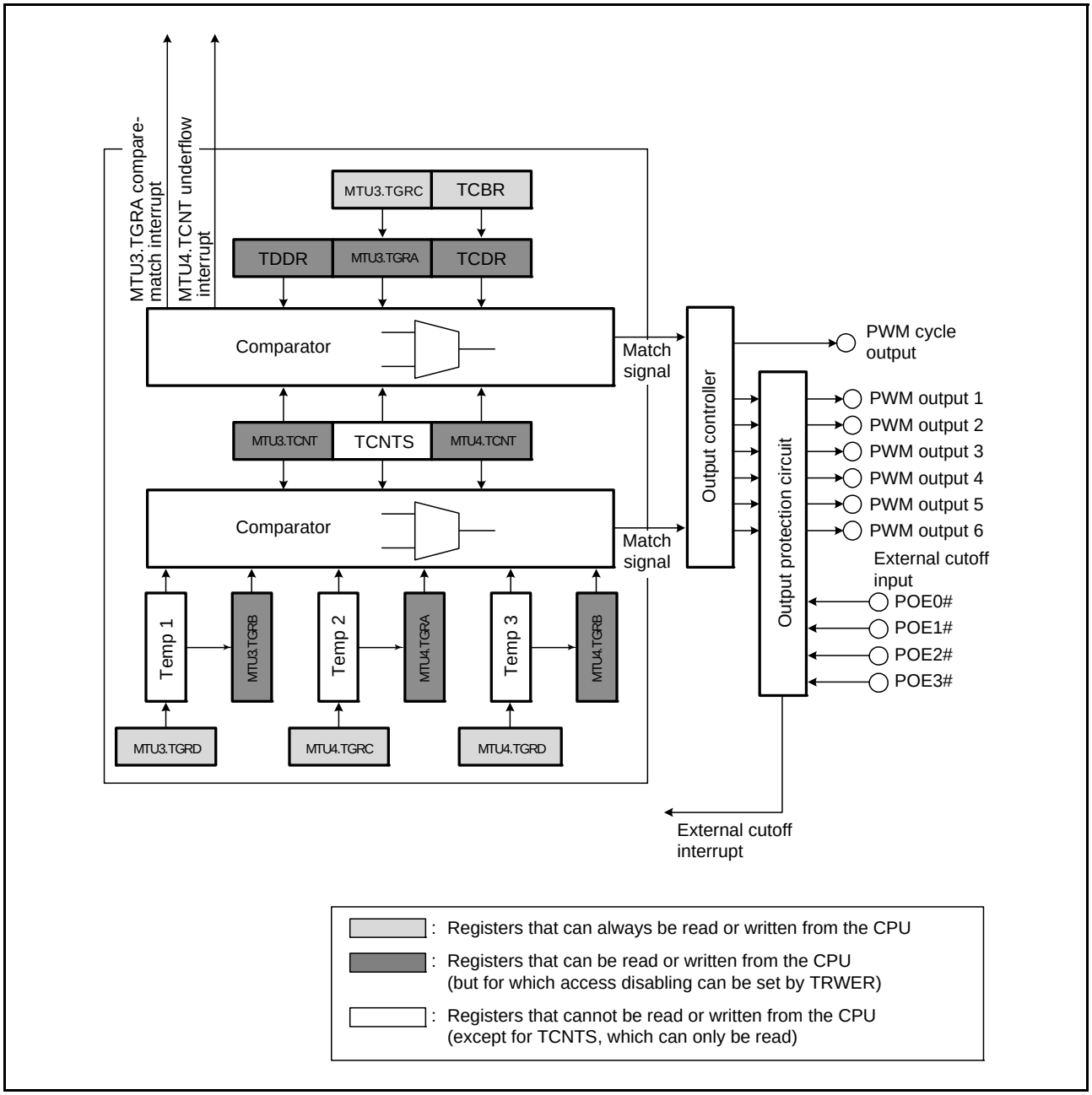
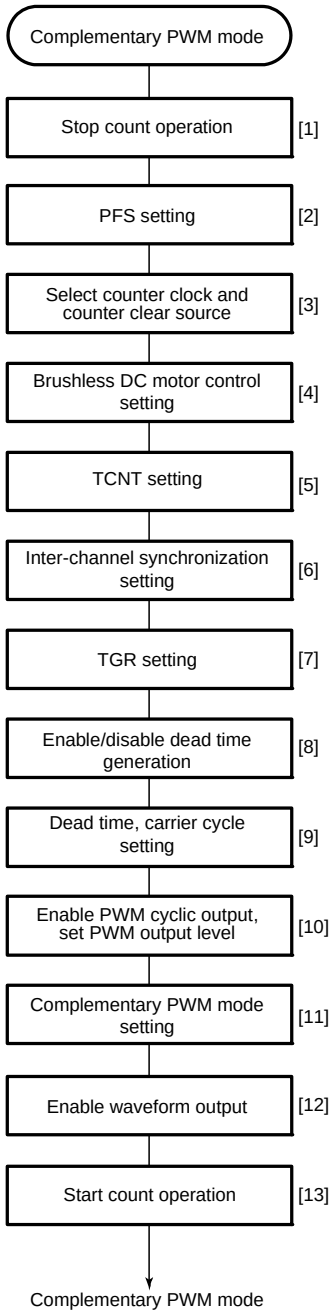


Figure 21.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 21.38 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 counter 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 bits CCLR[2:0] 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 MTU2.TOCR2, see Figure 21.3, 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 of 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 21.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases of PWM waveforms can be output. Figure 21.39 illustrates counter operation in complementary PWM mode, and Figure 21.40 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 CST bit in TSTR 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.

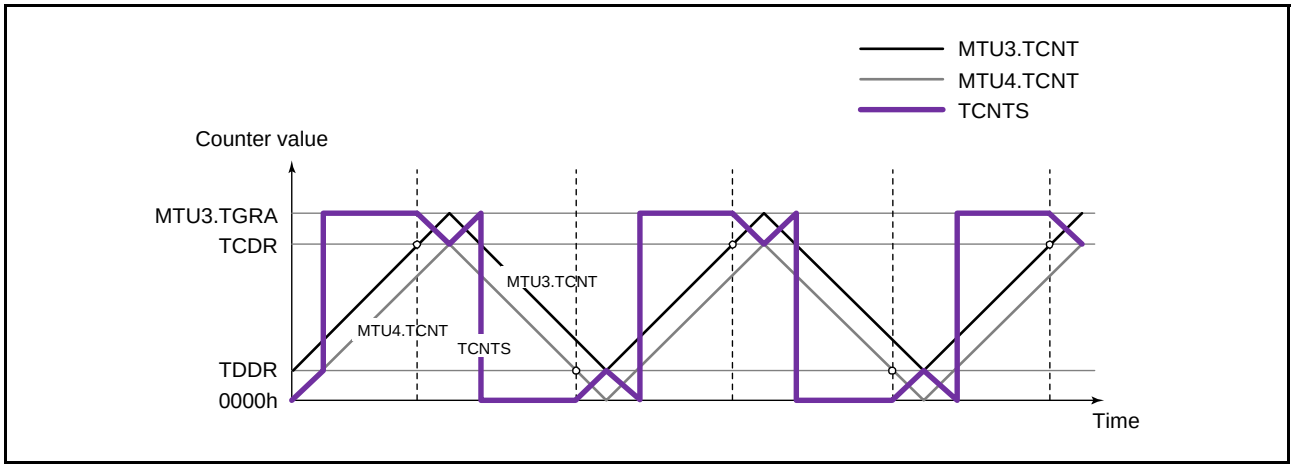


Figure 21.39 Counter Operation in Complementary PWM Mode

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used. Figure 21.40 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 bits OLSN and OLSP in the timer output control register 1 (TOCR1) is output.

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.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches MTU3.TGRA while TCNTS is counting up, or 0000h while counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register (TMDR). Figure 21.40 shows an example in which the trough is selected for the transfer timing.

In the Tb (Tb2 in Figure 21.40) 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.

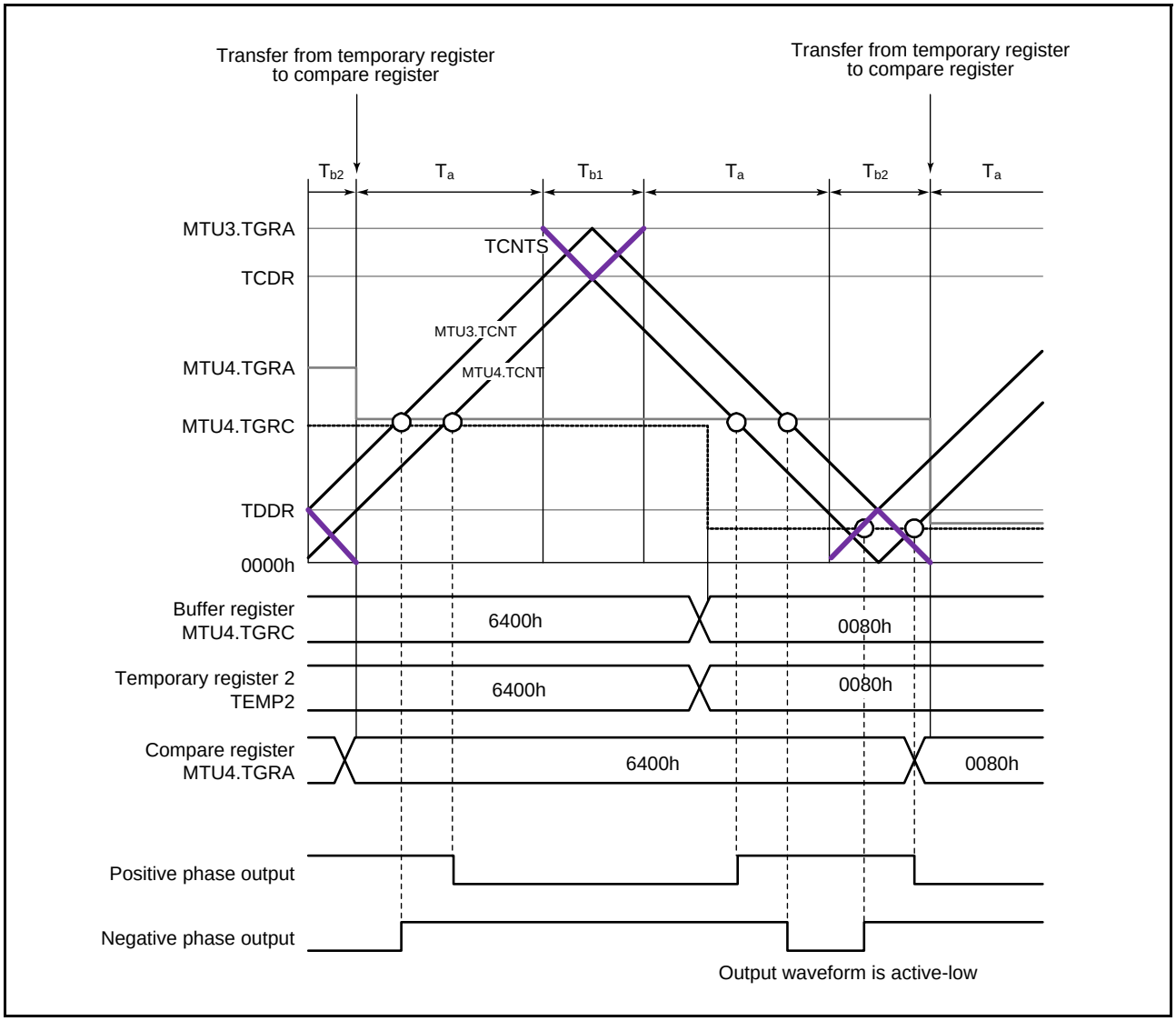


Figure 21.40 Example of Operation in Complementary PWM Mode

(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 bits MD[3:0] in the timer mode register (TMDR), 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 carrier 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 carrier 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 carrier 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 21.56 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
MTU3.TGRC	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier 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 carrier 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 carrier 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 carrier cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse 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, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.
The non-overlap 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 non-overlapping interval 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 bit in the timer dead time enable register (TDER) to 0. TDER bit can be cleared to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 21.41 shows an example of operation without dead time.

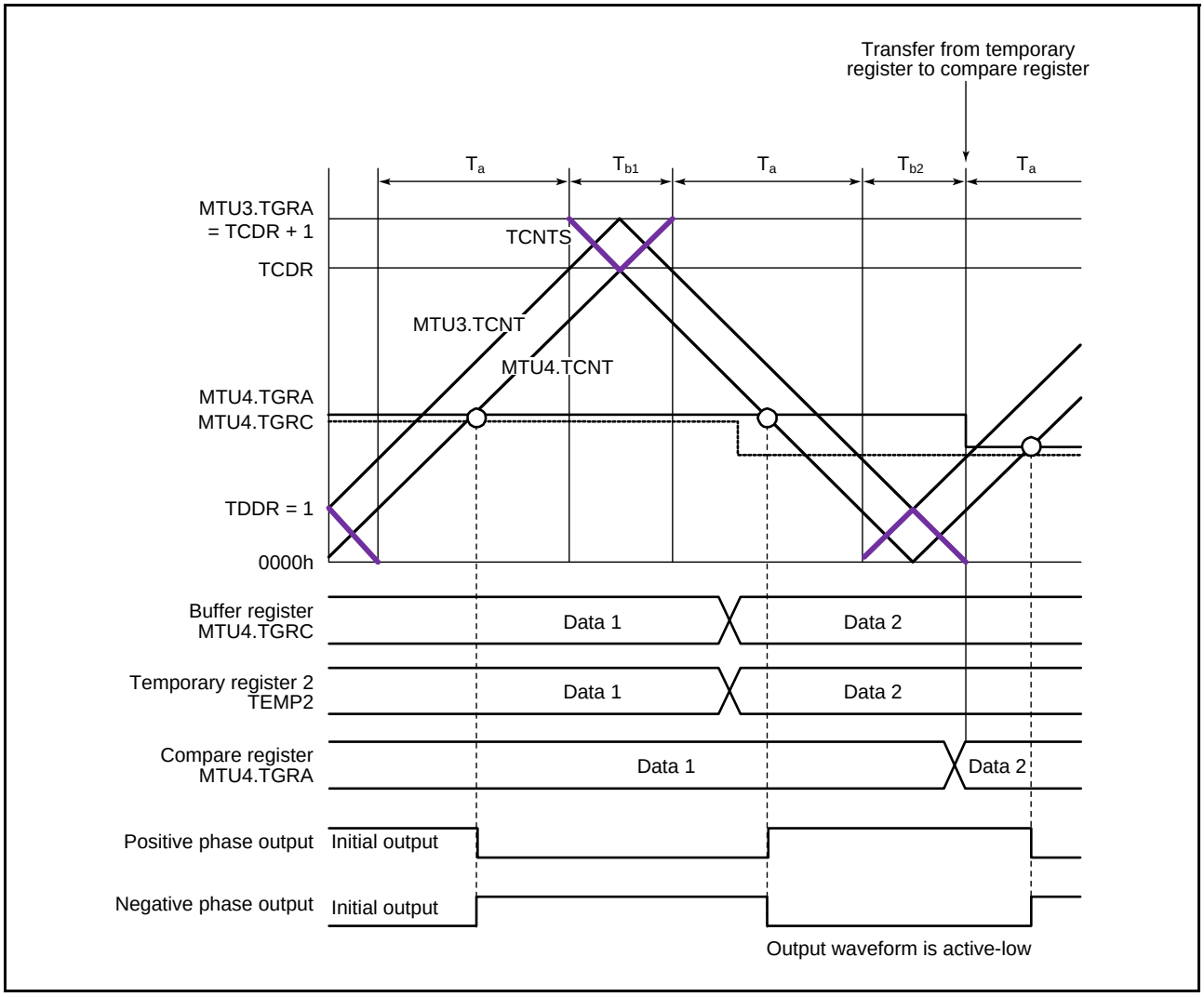


Figure 21.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse 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 MTU3.TCDR settings are made by setting values in buffer registers MTU3.TGRC and MTU3.TCBR. The values set in MTU3.TGRC and MTU3.TCBR are transferred simultaneously to MTU3.TGRA and MTU3.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 21.42 illustrates the operation when the PWM cycle is updated at the crest.
See the following section (h), Register Data Updating, for the method of updating the data in each buffer register.

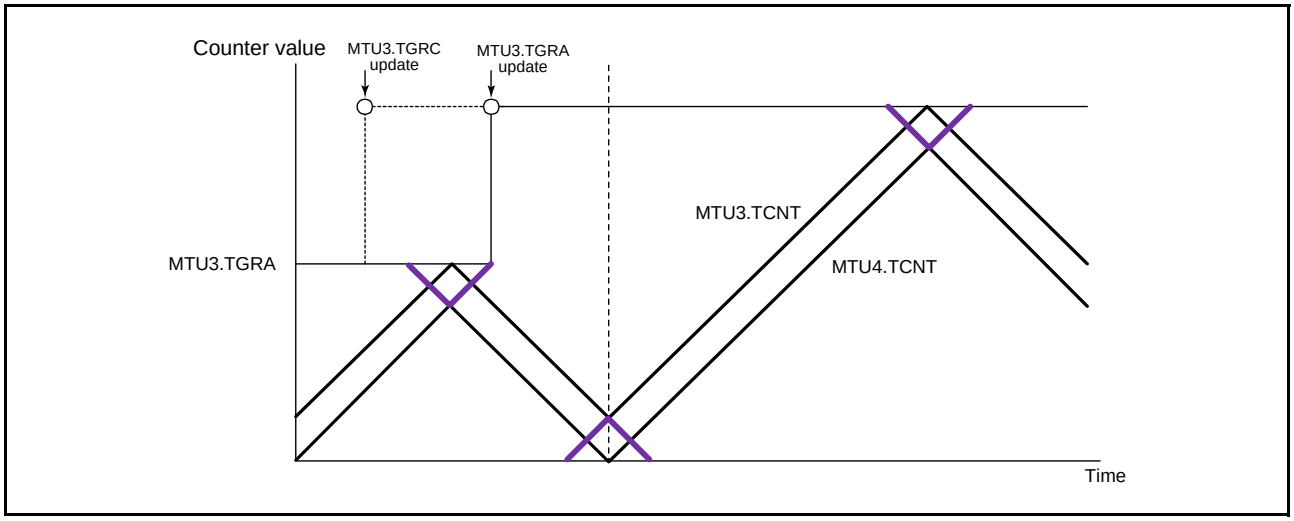


Figure 21.42 Example of PWM Cycle Updating

(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 carrier 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 bits MD[3:0] in the timer mode register (TMDR). Figure 21.43 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.

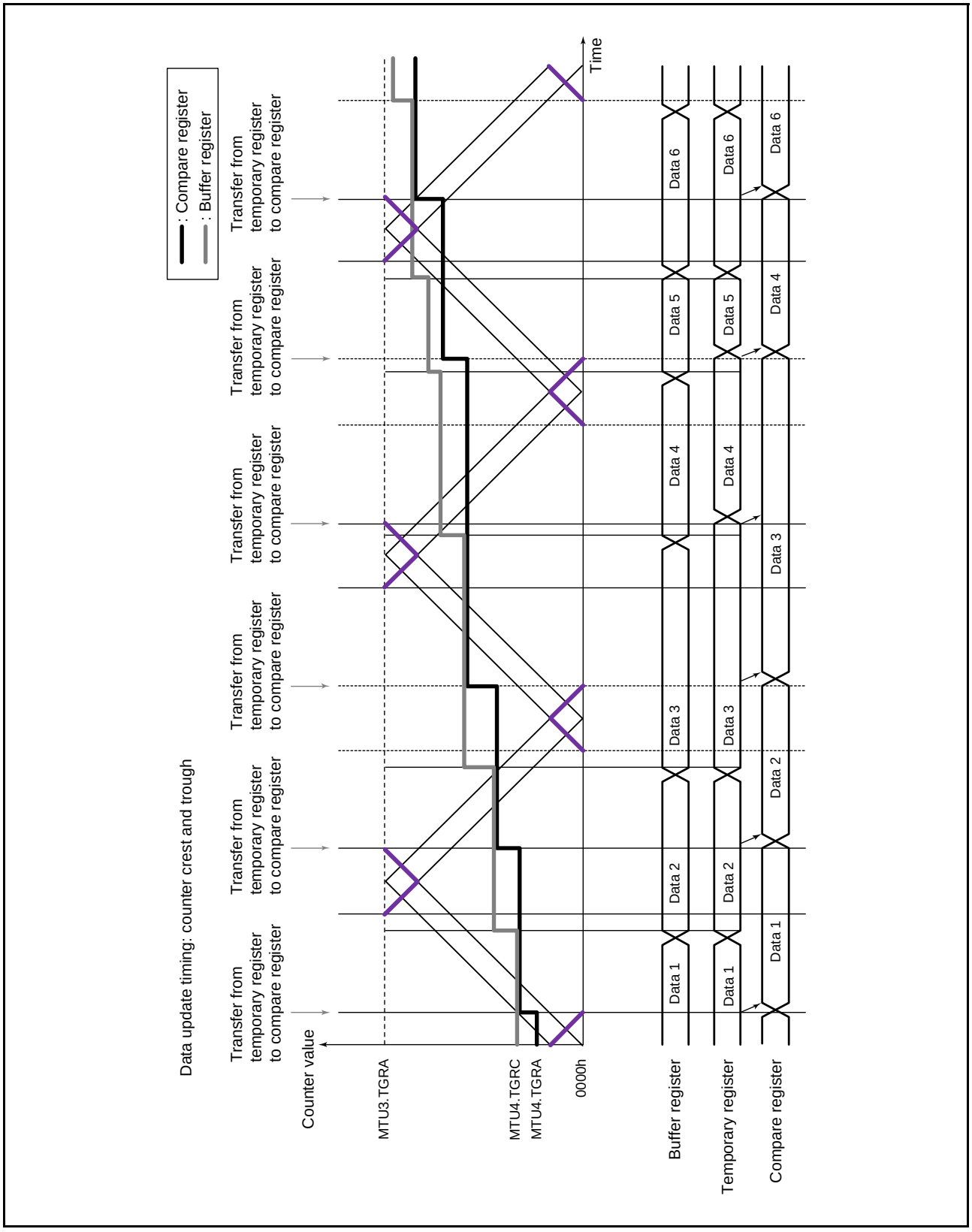


Figure 21.43 Example of Data Updating in Complementary PWM Mode

(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 pulse 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 21.44 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 21.45.

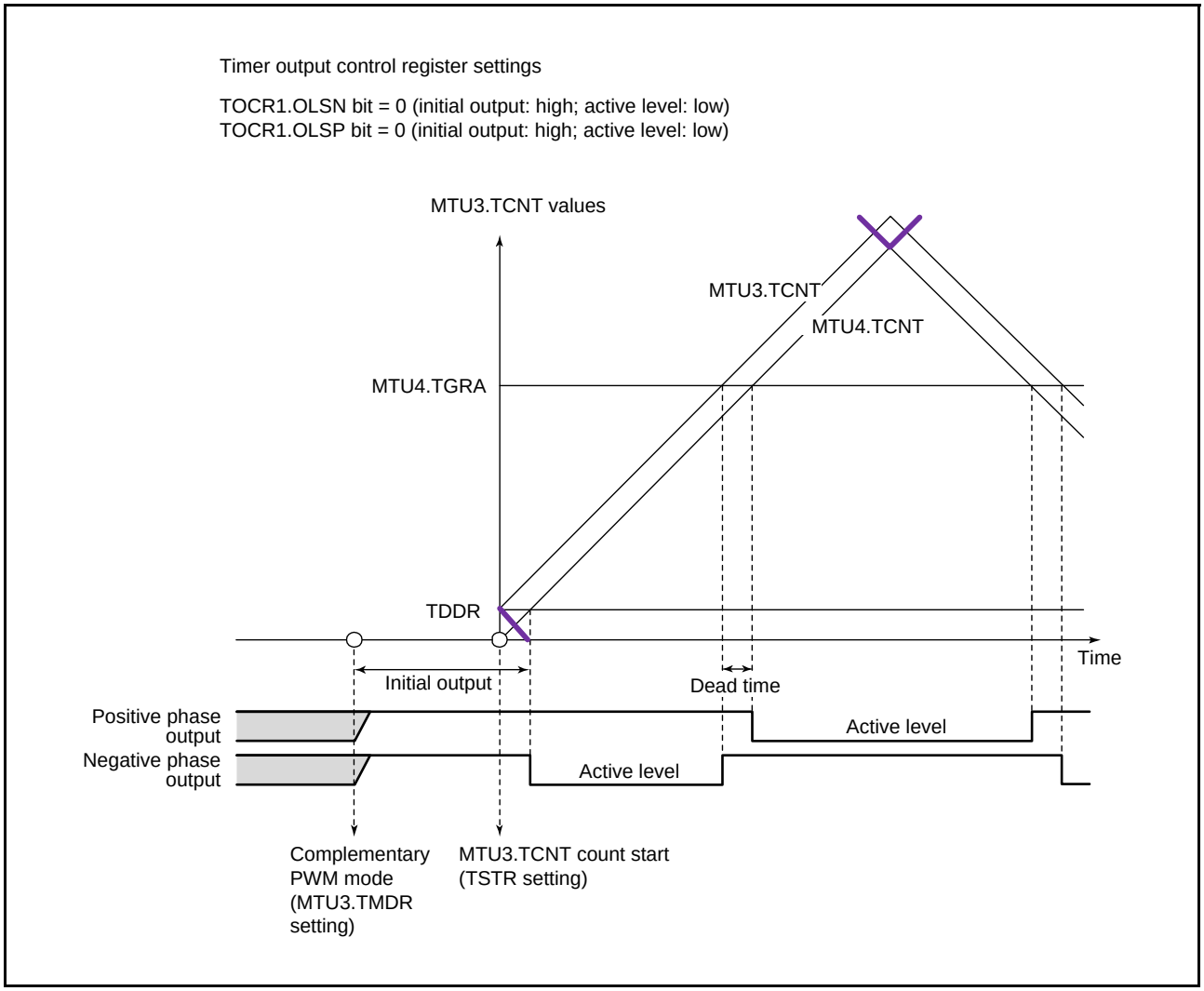


Figure 21.44 Example of Initial Output in Complementary PWM Mode (1)

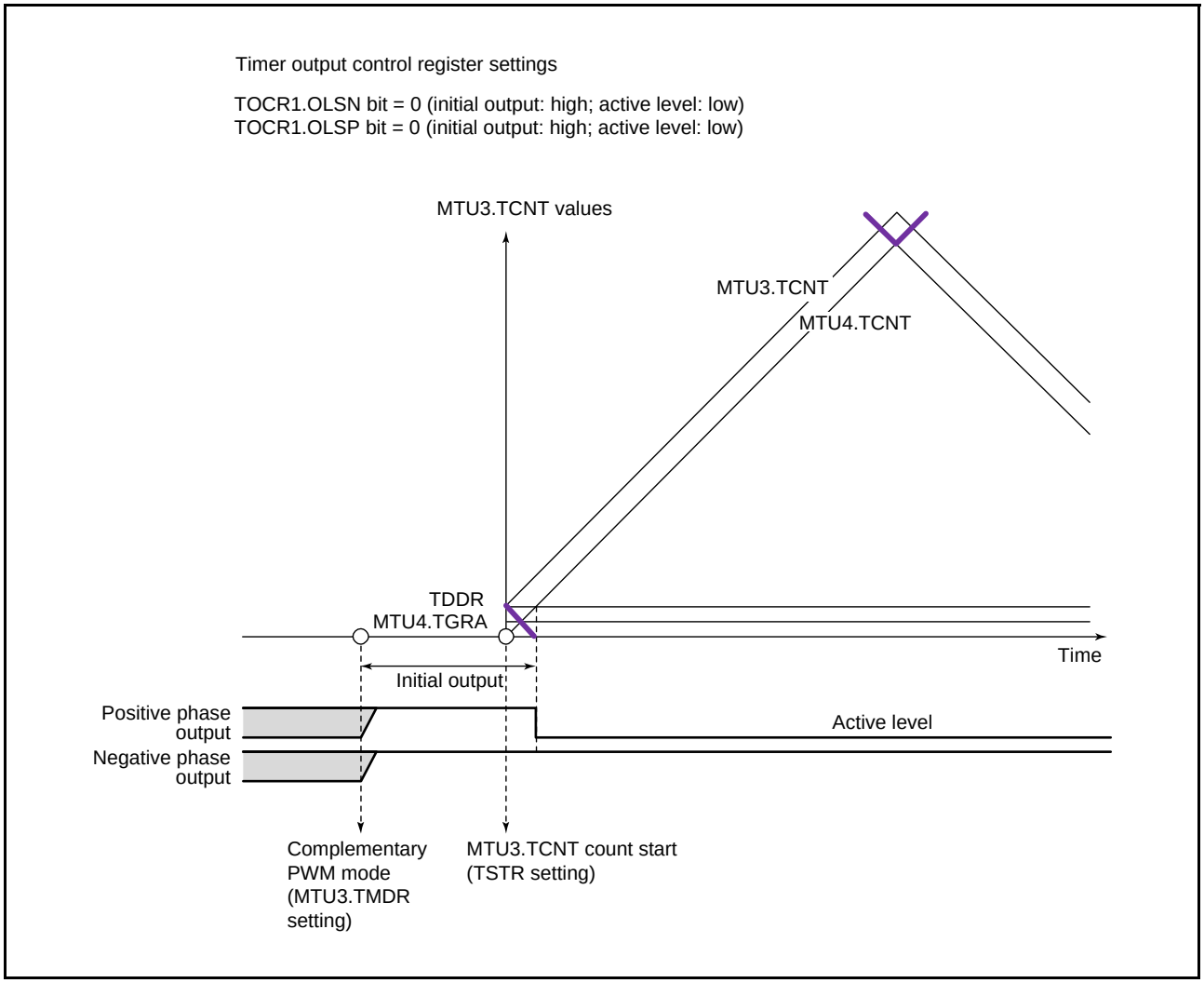


Figure 21.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, three phases of PWM waveforms are output with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.
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 create consecutive PWM pulses from 0 to 100%. 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 21.46 to Figure 21.48 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 MTU3.TCNT counter, and the turn-on timing by a compare match with the MTU4.TCNT counter, which operates with a delay of the dead time behind the MTU3.TCNT counter. 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 21.46.
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 21.47, 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 21.48, compare match a' with new data in the temporary register occurs before compare match c, but until compare match c, which turns off the positive phase, other compare matches are ignored. 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.

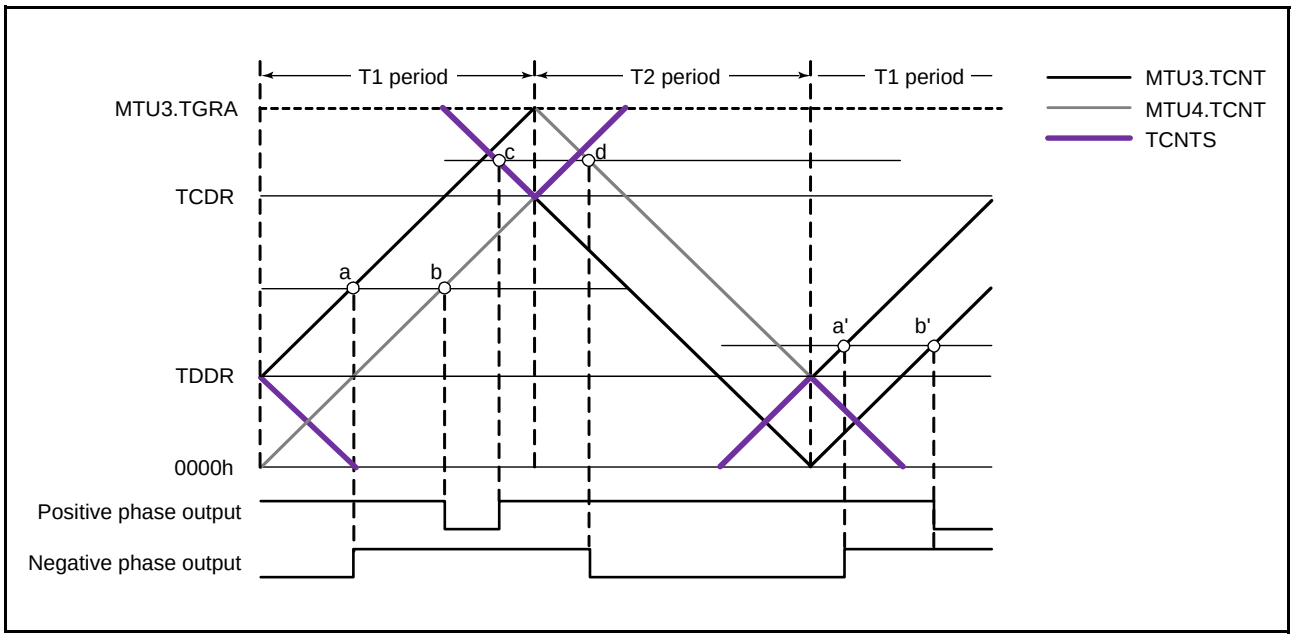


Figure 21.46 Example of Waveform Output in Complementary PWM Mode (1)

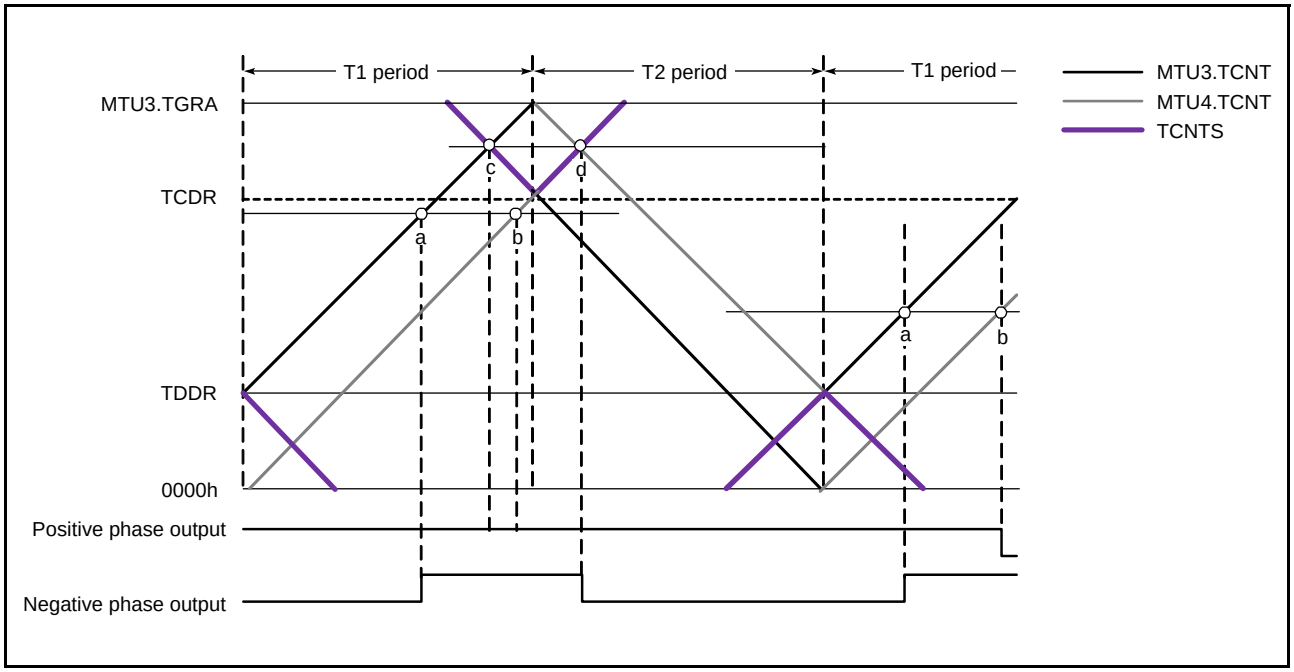


Figure 21.47 Example of Waveform Output in Complementary PWM Mode (2)

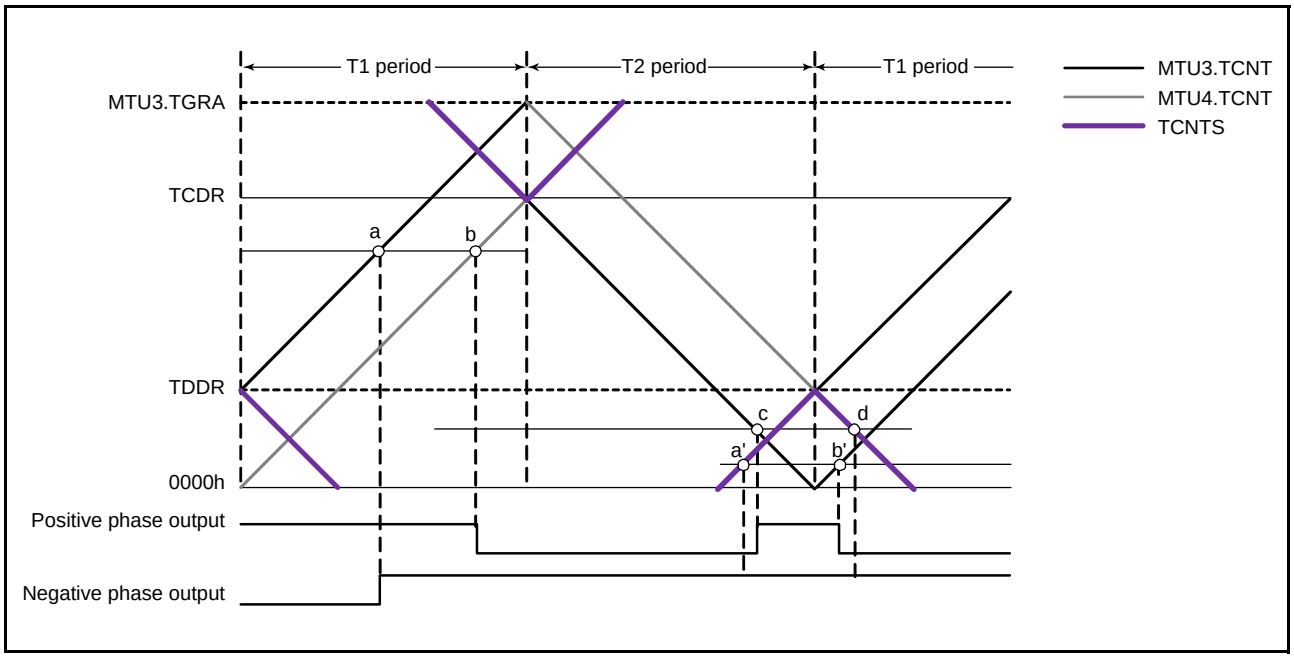


Figure 21.48 Example of Waveform Output in Complementary PWM Mode (3)

(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 21.49 to Figure 21.53 show output examples.

A 100% duty cycle waveform is output when the compare register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty cycle waveform is output when the compare 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.

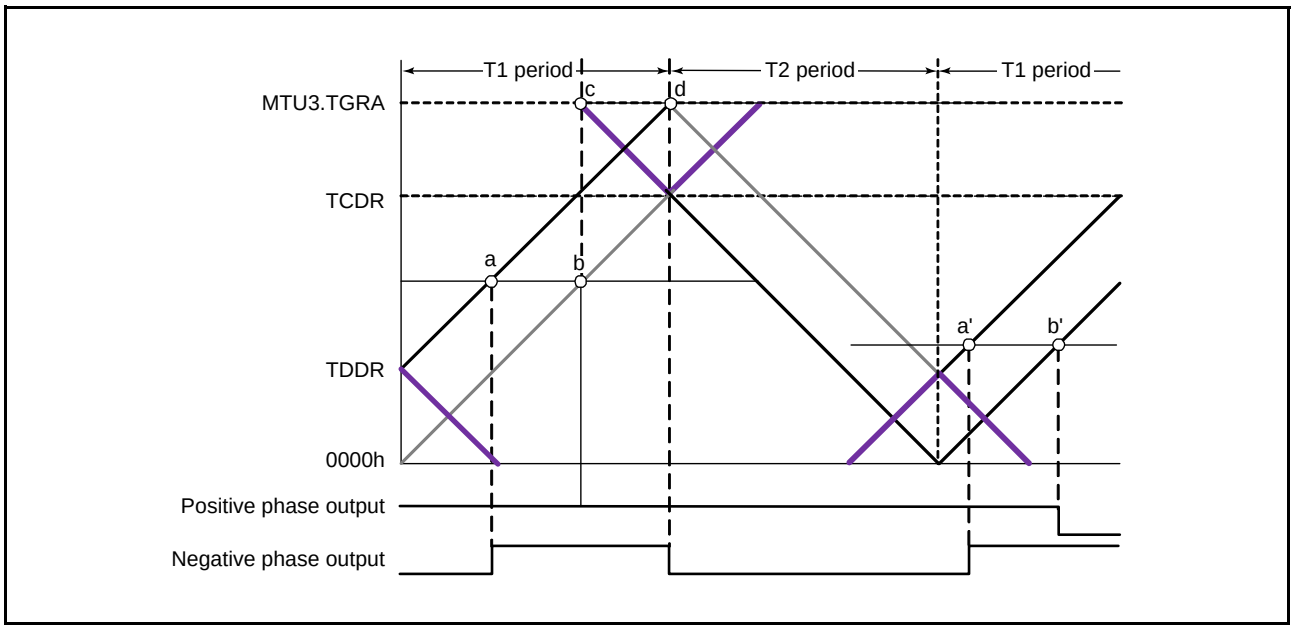


Figure 21.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

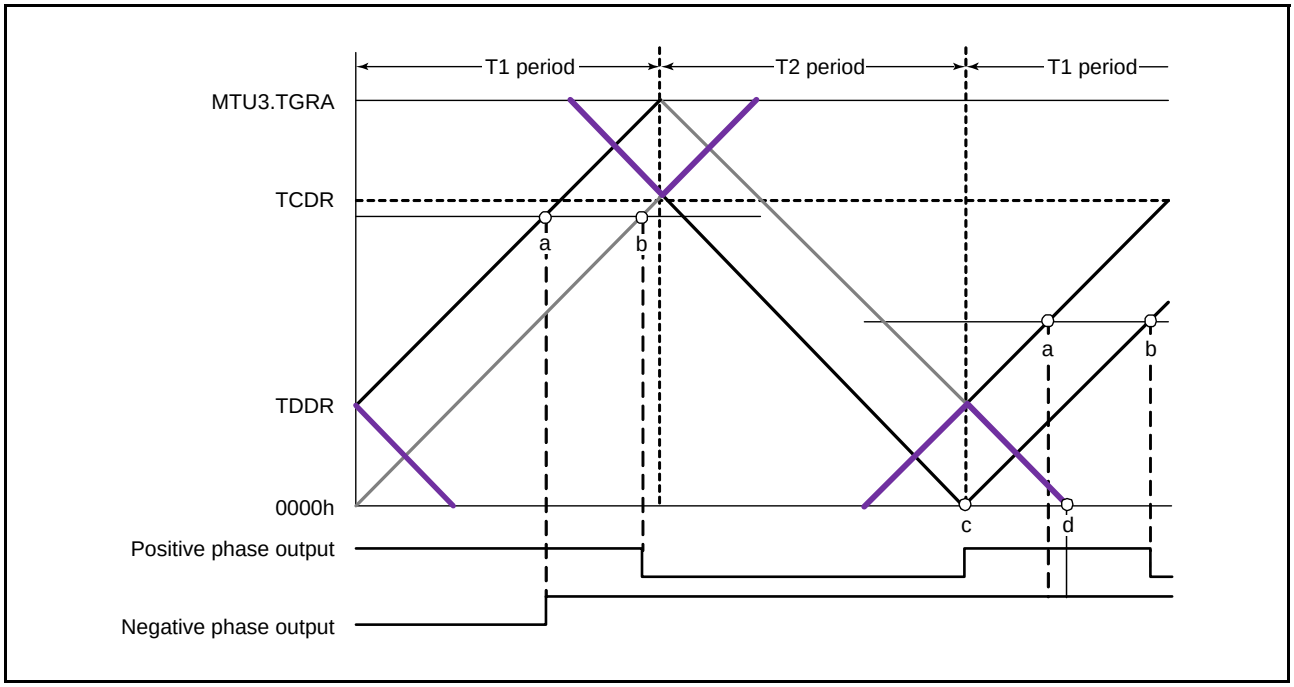


Figure 21.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

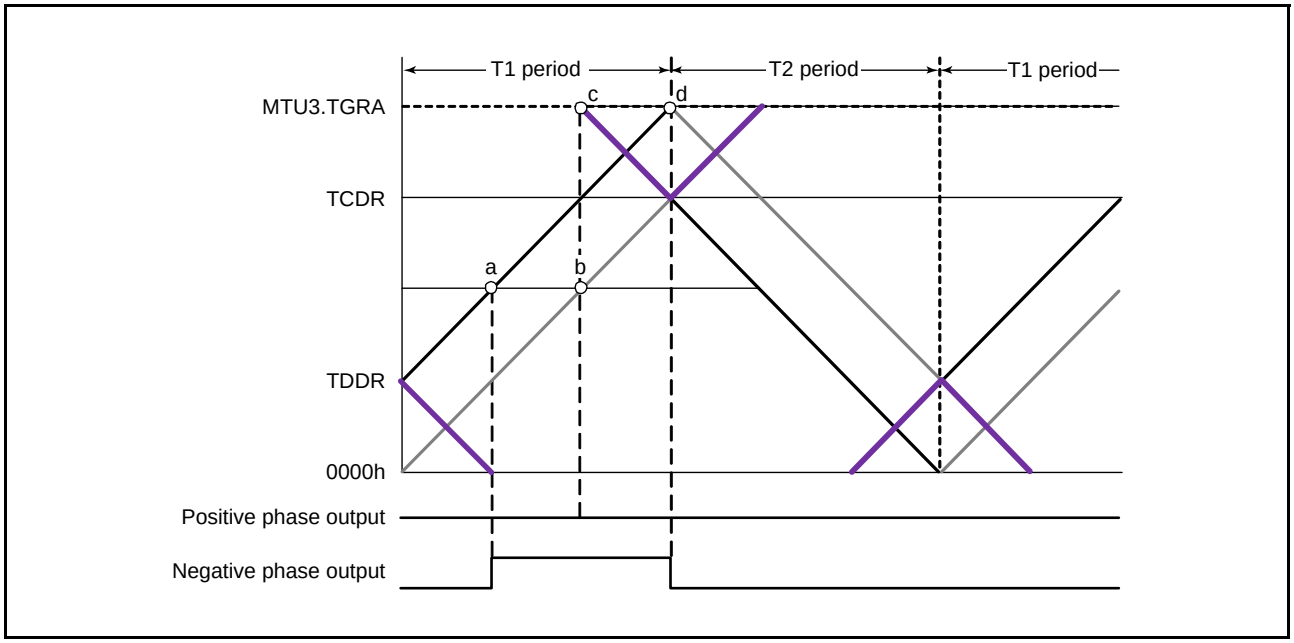


Figure 21.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

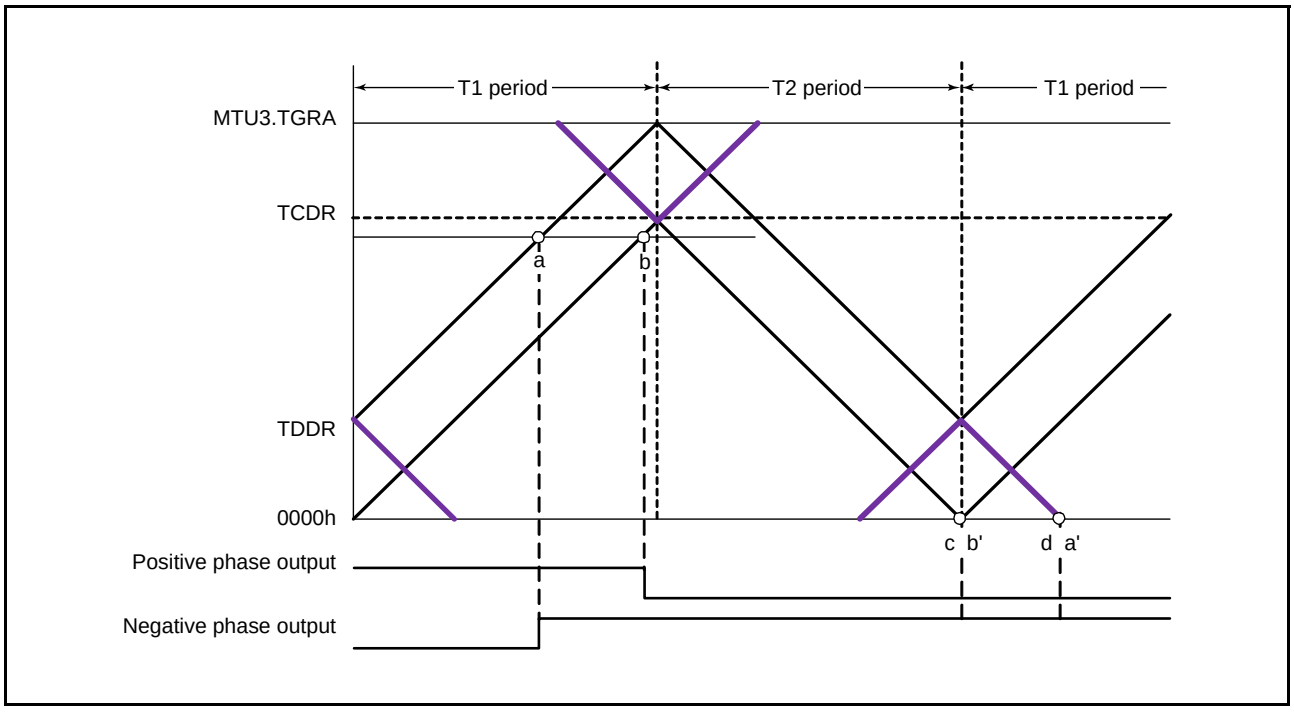


Figure 21.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

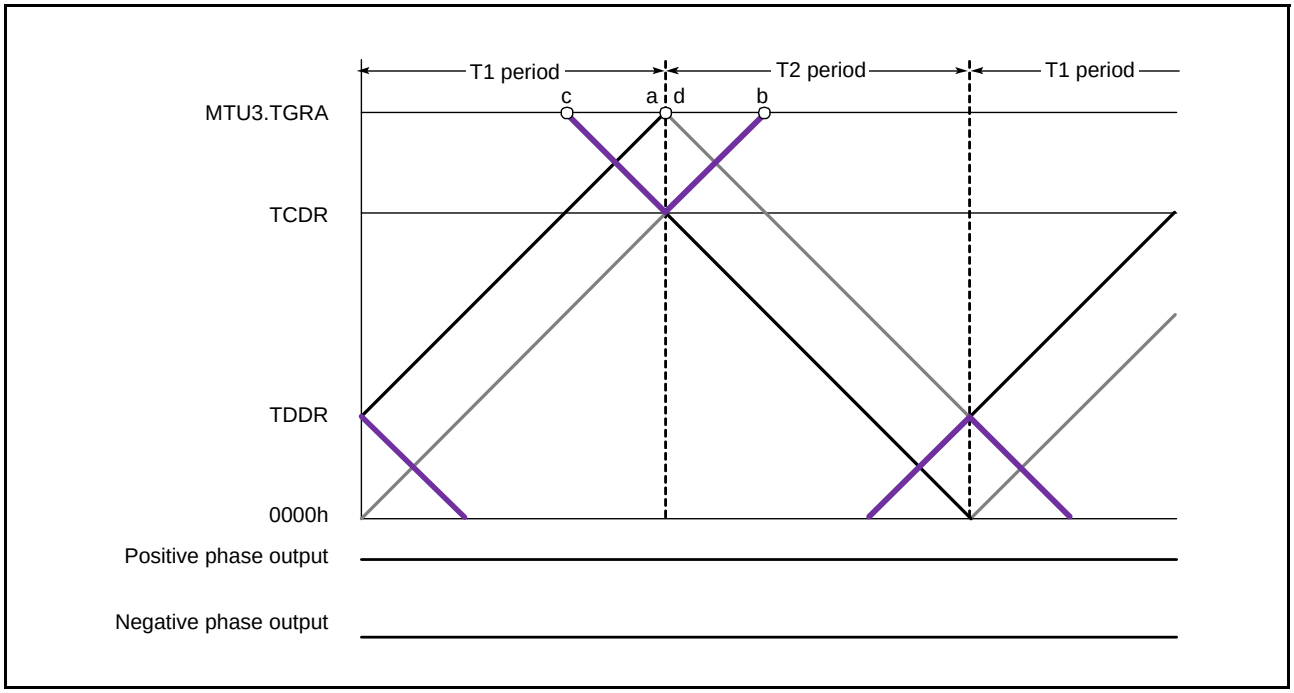


Figure 21.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output in synchronization with the PWM carrier cycle can be generated by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 21.54.

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 High.

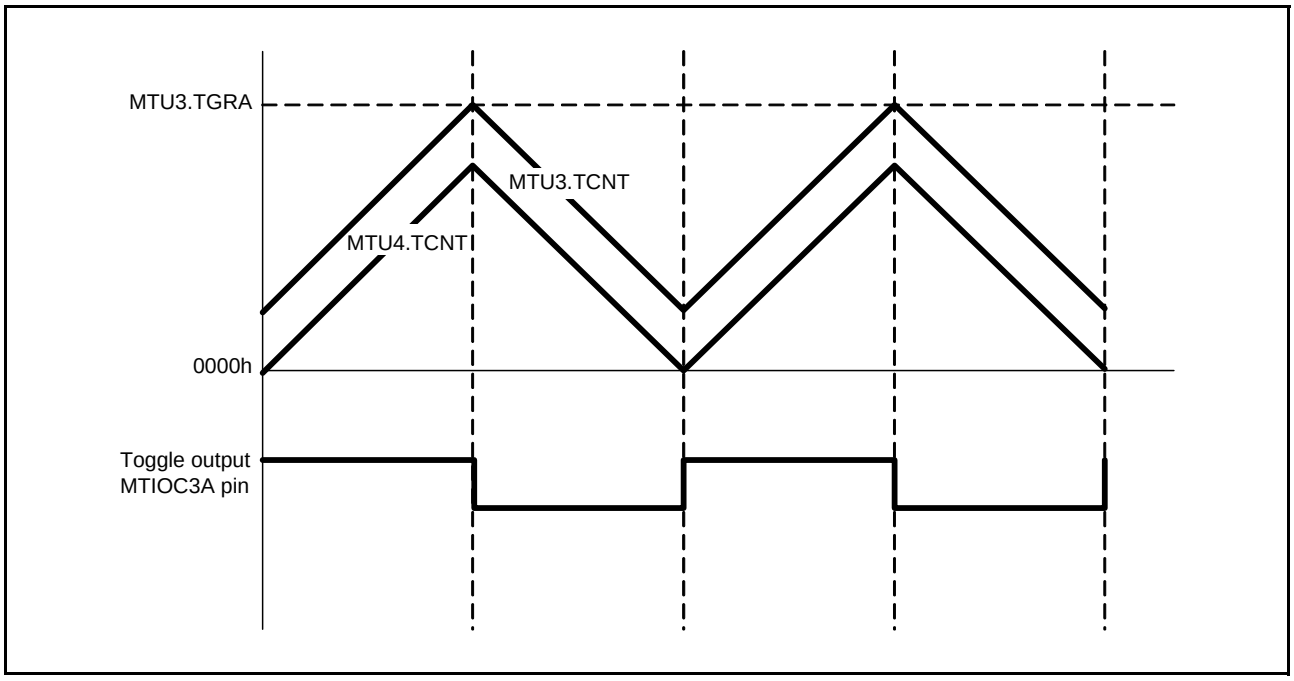


Figure 21.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYR) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR).

Figure 21.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

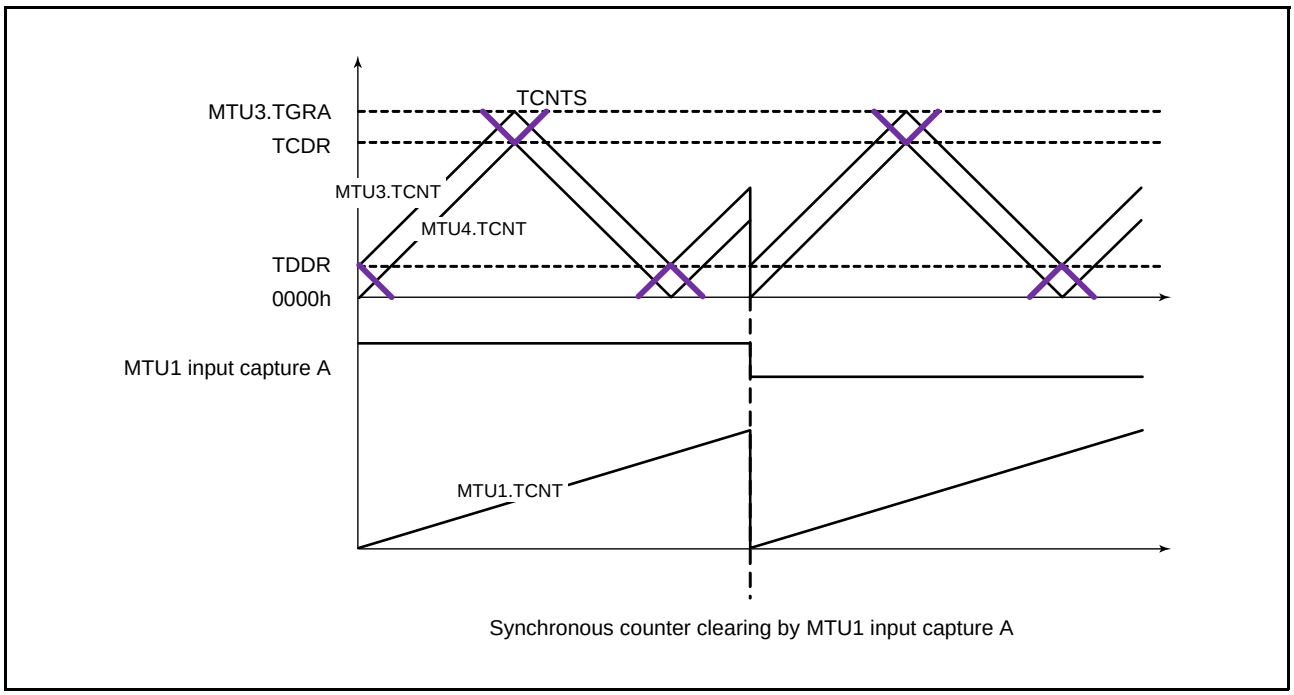


Figure 21.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing. Initial output suppression through setting WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 21.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1 is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 21.56) 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.

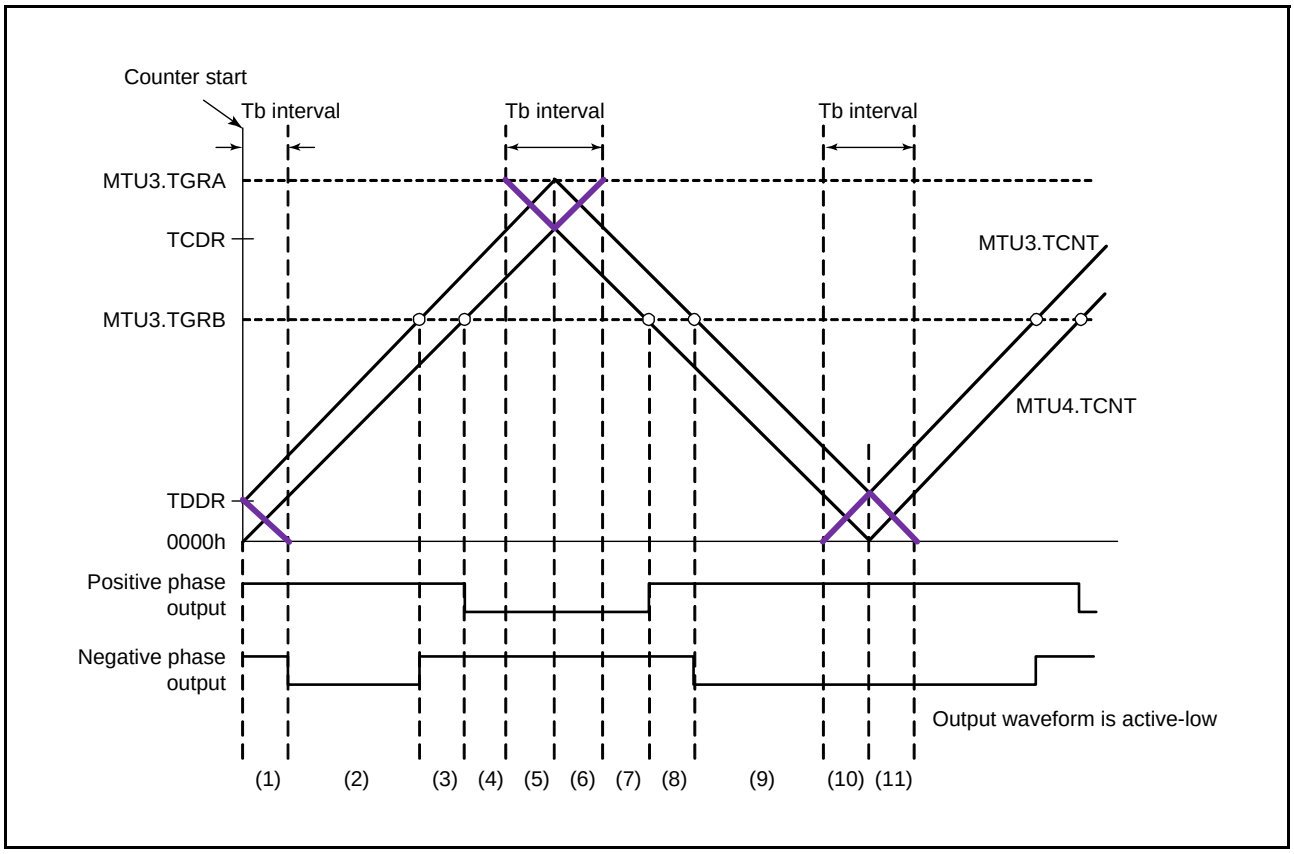


Figure 21.56 Timing for Synchronous Counter Clearing

- 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 21.57.

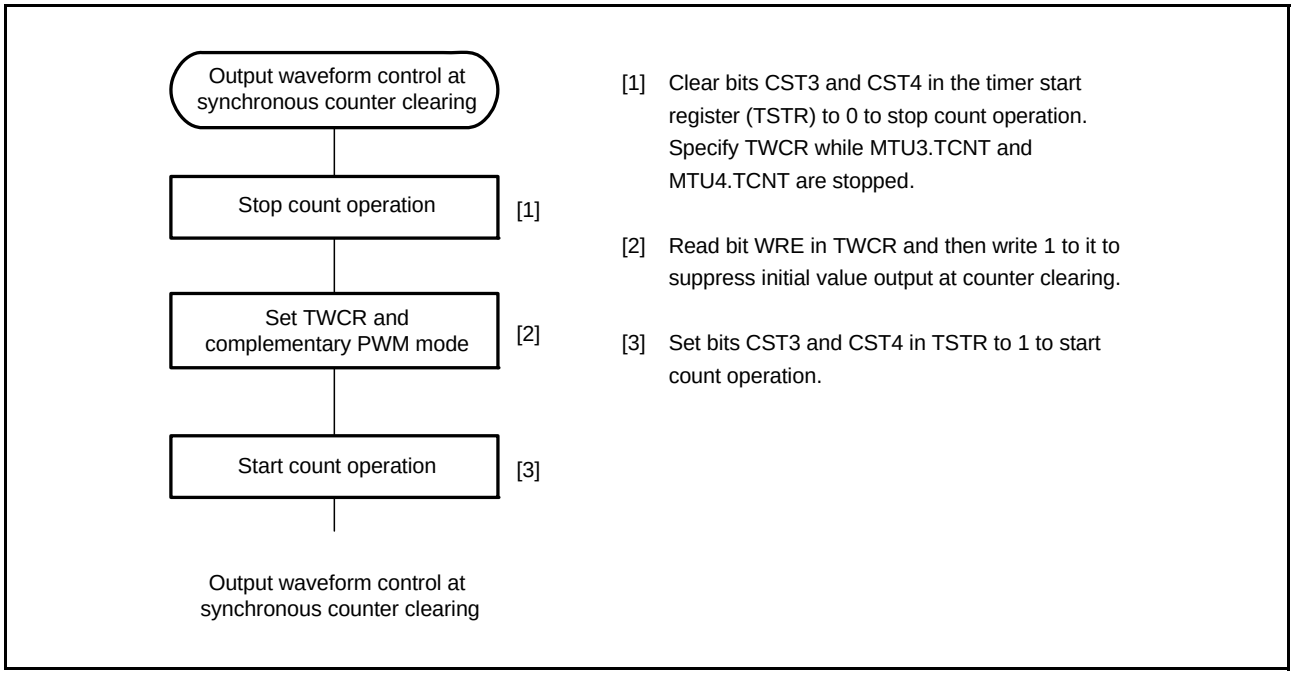


Figure 21.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 21.58 to Figure 21.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in Figure 21.58 to Figure 21.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 21.56, respectively.

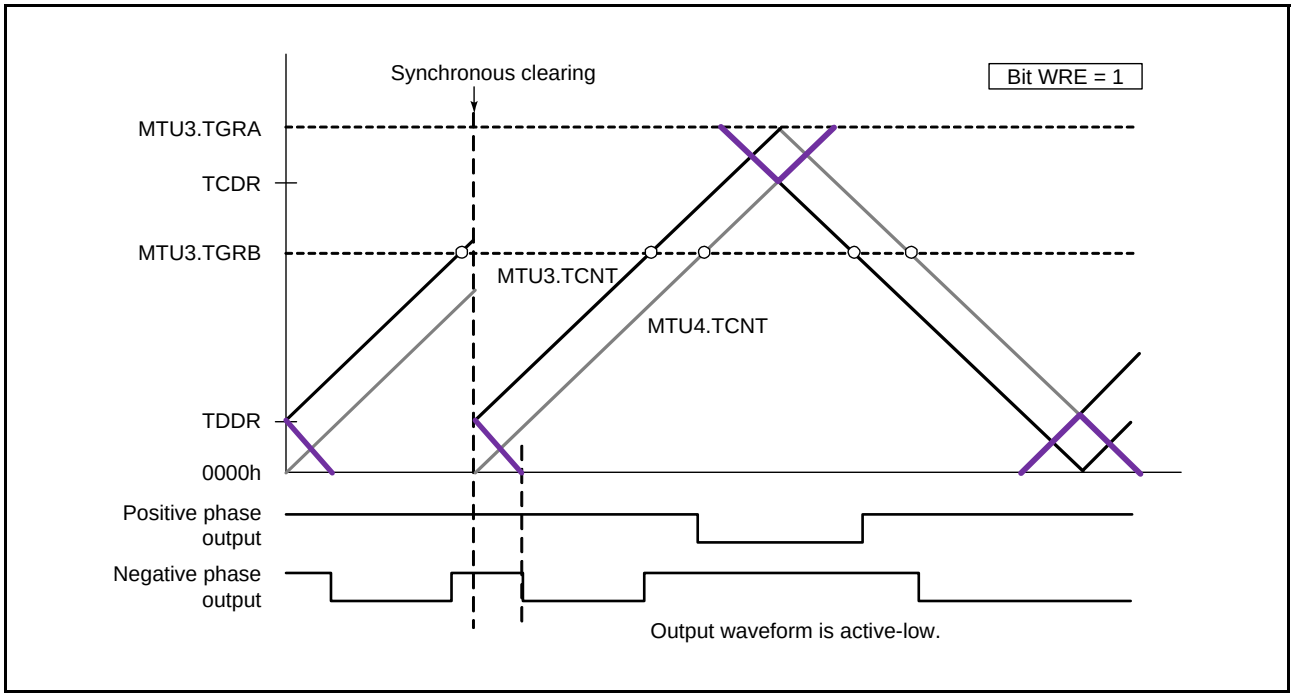


Figure 21.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 21.56; Bit WRE of TWCR is 1 in the MTU)

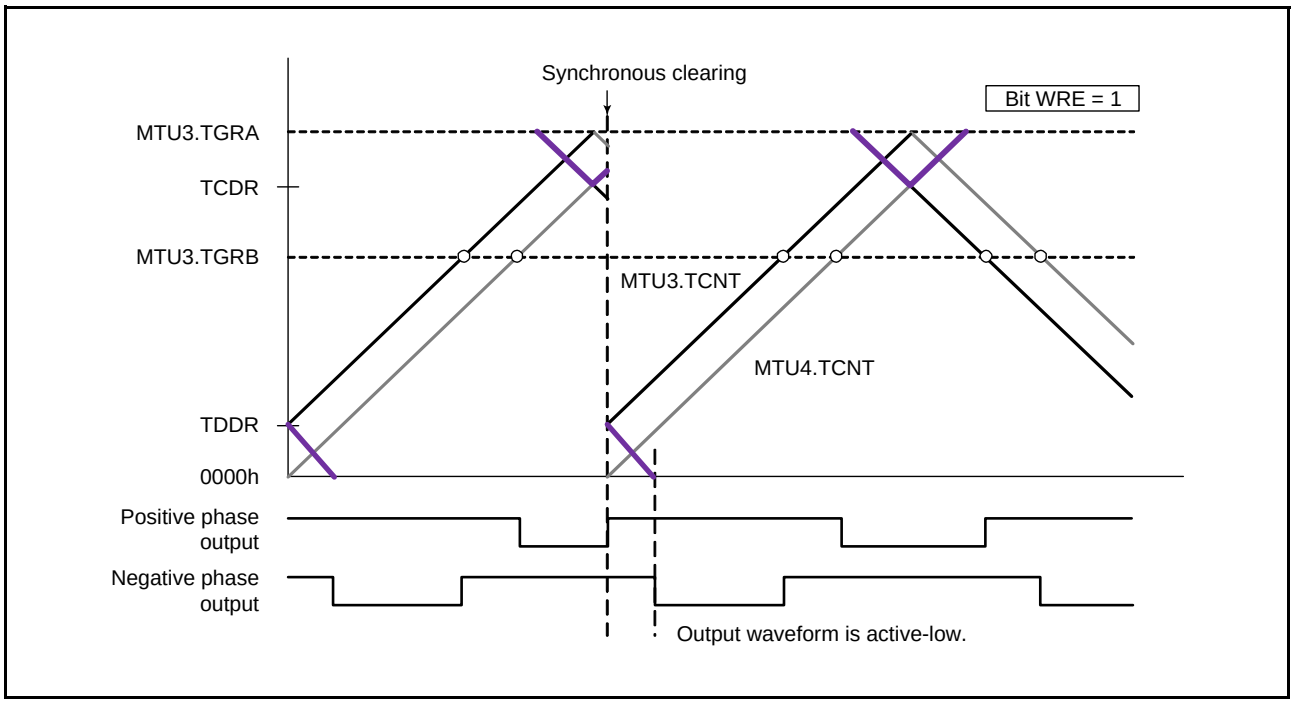


Figure 21.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 21.56; Bit WRE of TWCR is 1 in the MTU)

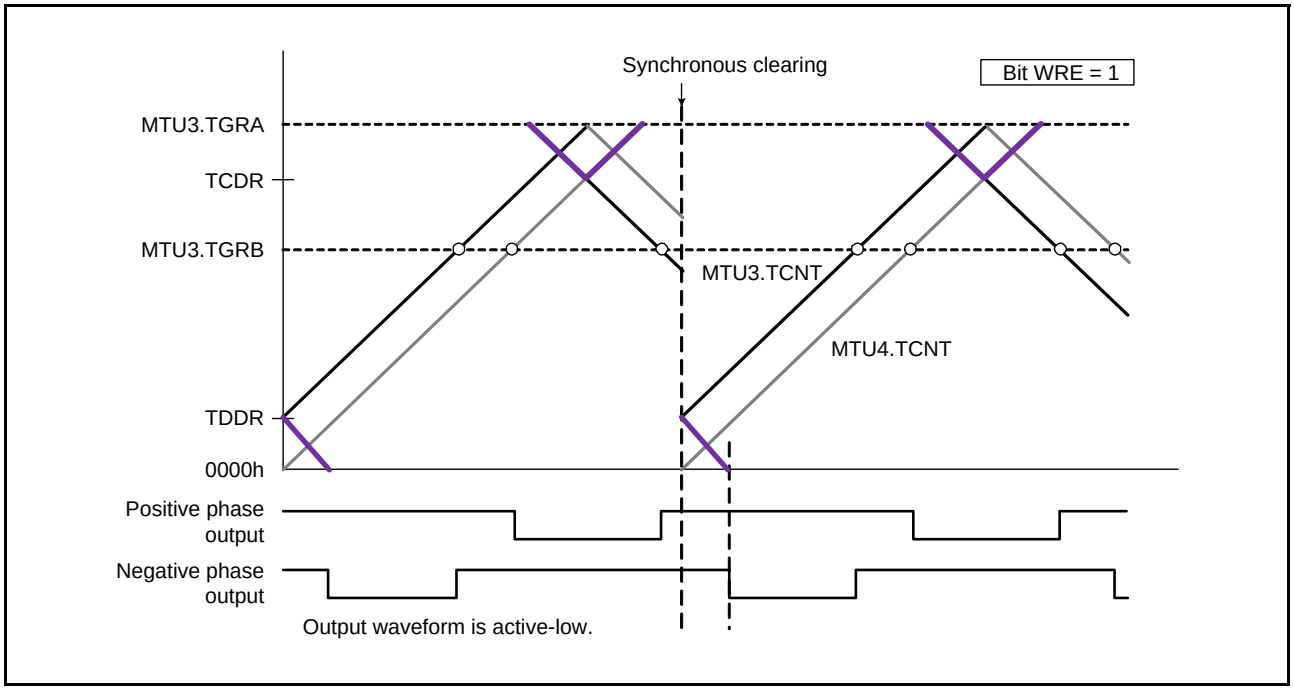


Figure 21.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 21.56; Bit WRE of TWCR is 1)

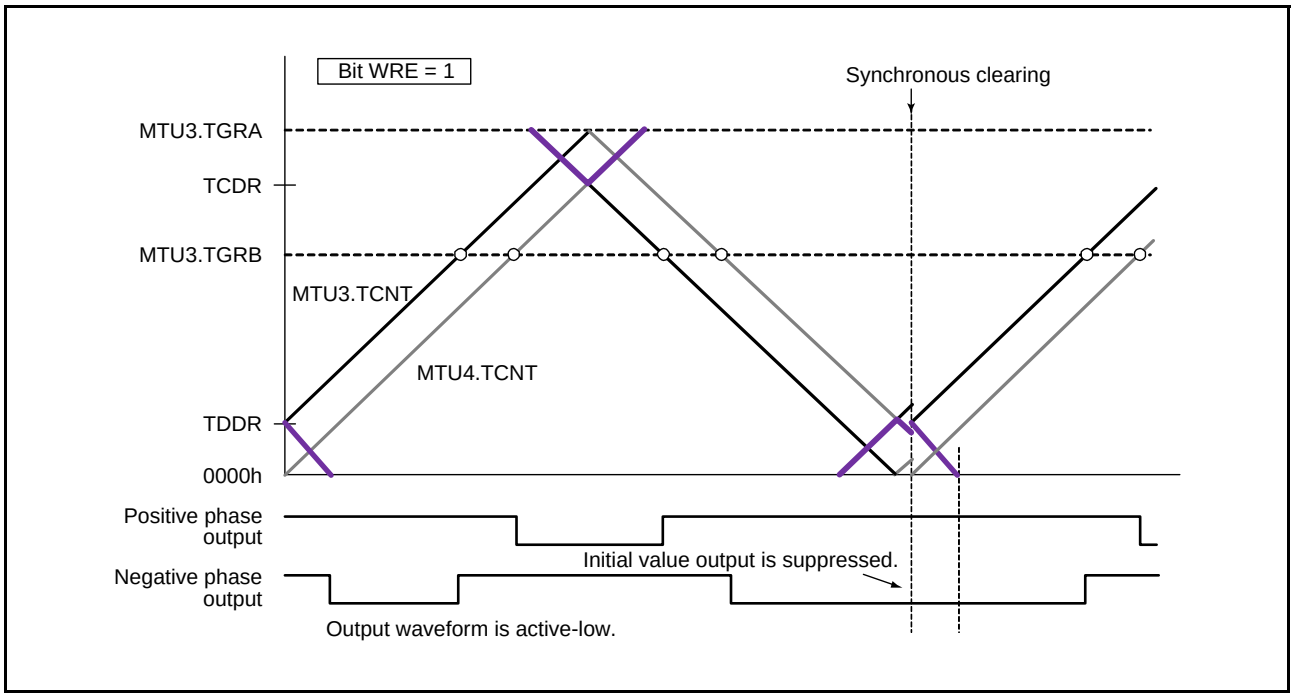


Figure 21.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 21.56; Bit WRE of TWCR is 1)

(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 CCE bit is set in the timer waveform control register (TWCR).

Figure 21.62 illustrates 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 PSYE bit in timer output control register 1 (TOCR1) to 1.

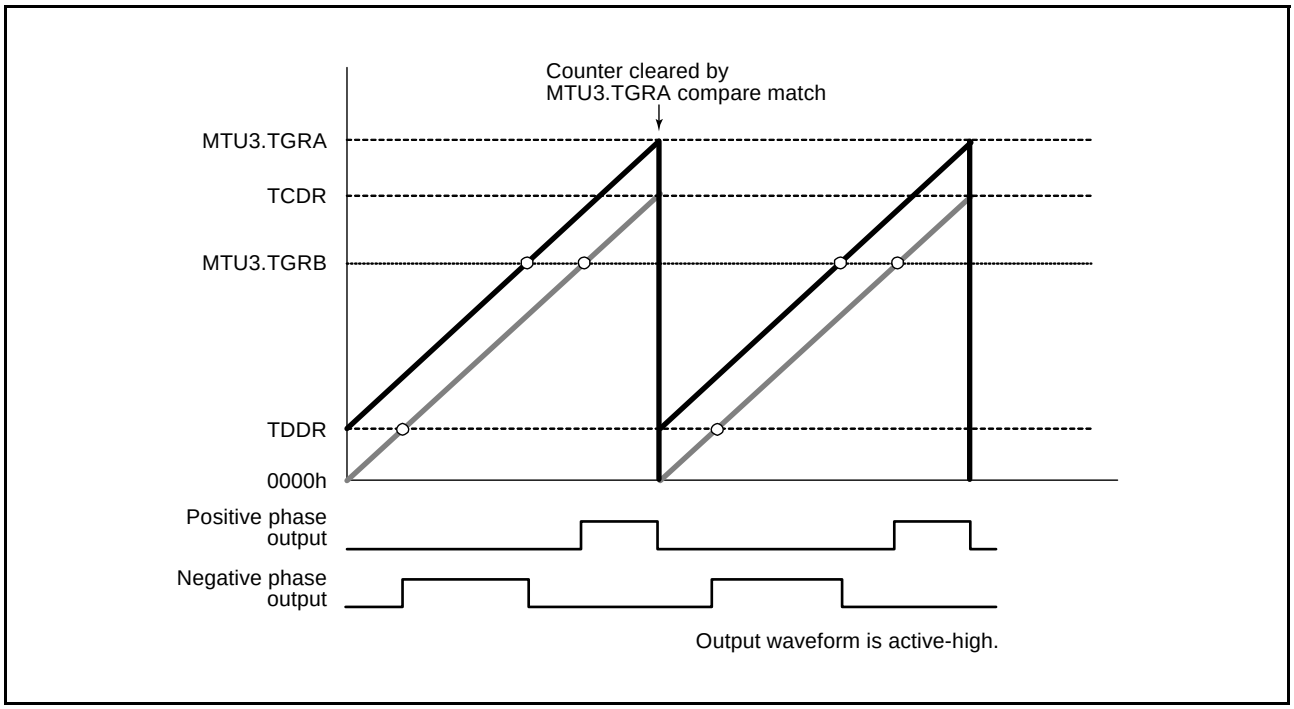


Figure 21.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(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 21.63 to Figure 21.66 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 FB bit in TGCR 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 (set with PFS). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.
When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.
The driving waveforms are output from the 6-phase output pins for complementary PWM mode.
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCR to 1. When the N bit or P bit is 0, the level output is selected.
The active level of the 6-phase output (on output level) can be set with the OLSN and OLSP bits in the timer output control register1 (TOCR1) regardless of the setting of the N and P bits.

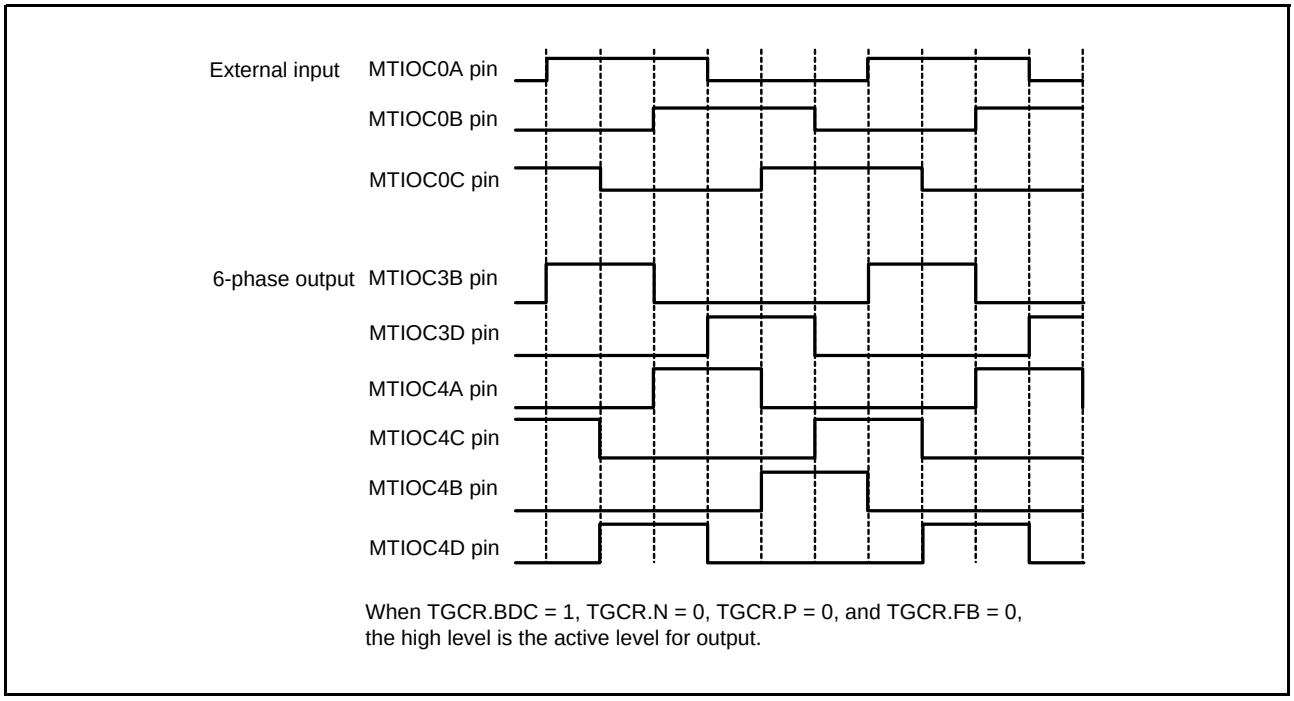


Figure 21.63 Example of Output Phase Switching by External Input (1)

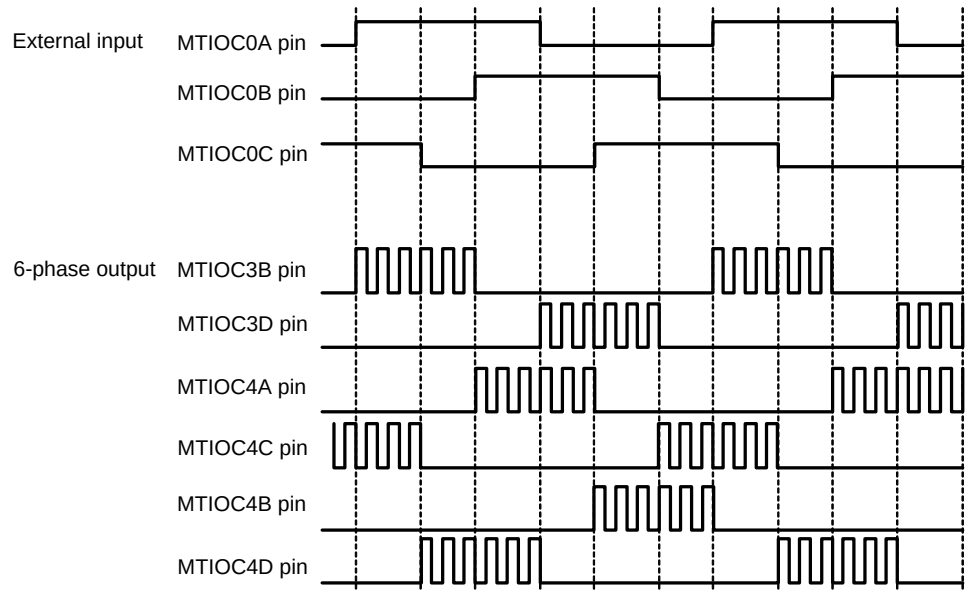


Figure 21.64 Example of Output Phase Switching by External Input (2)

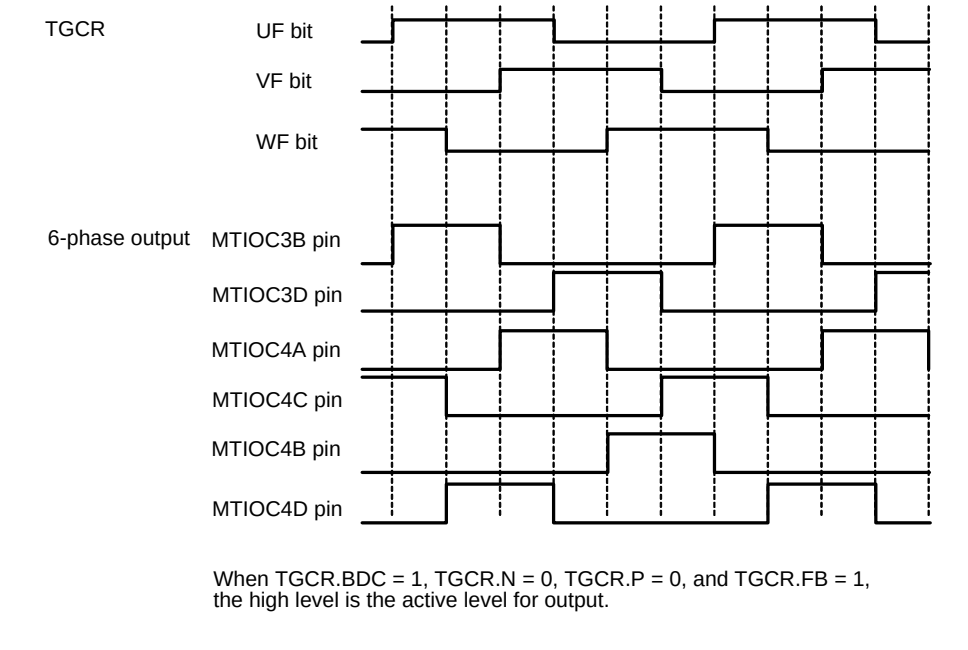


Figure 21.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

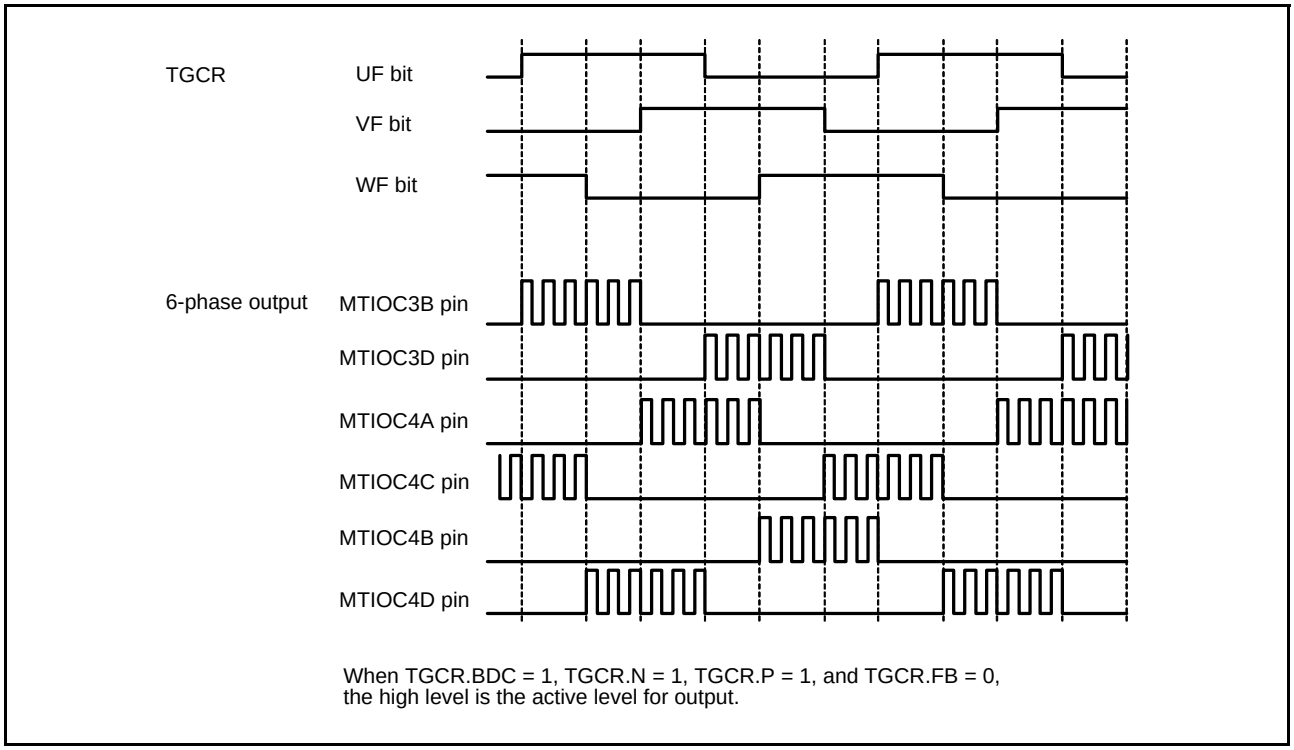


Figure 21.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(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 TTGE bit to 1 in the timer interrupt enable register (TIER).
To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the TTGE2 bit in MTU4.TIER 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 making settings in 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 21.3.9, 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 T3AEN and T4VEN bits in TITCR to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 21.67 shows an example of the interrupt skipping operation setting procedure. Figure 21.68 shows the periods during which interrupt skipping count can be changed.

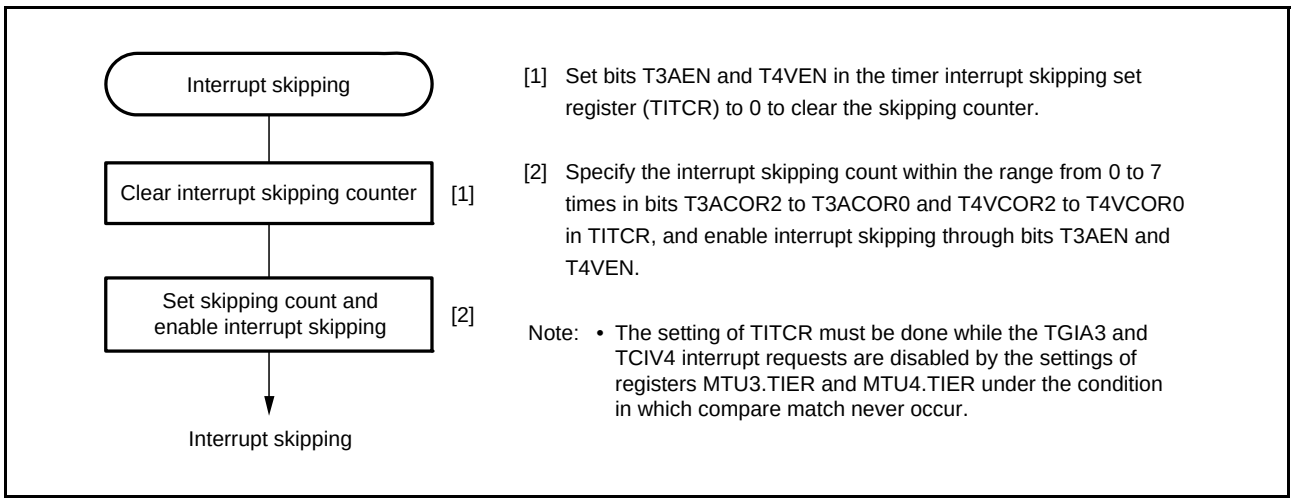


Figure 21.67 Example of Interrupt Skipping Operation Setting Procedure

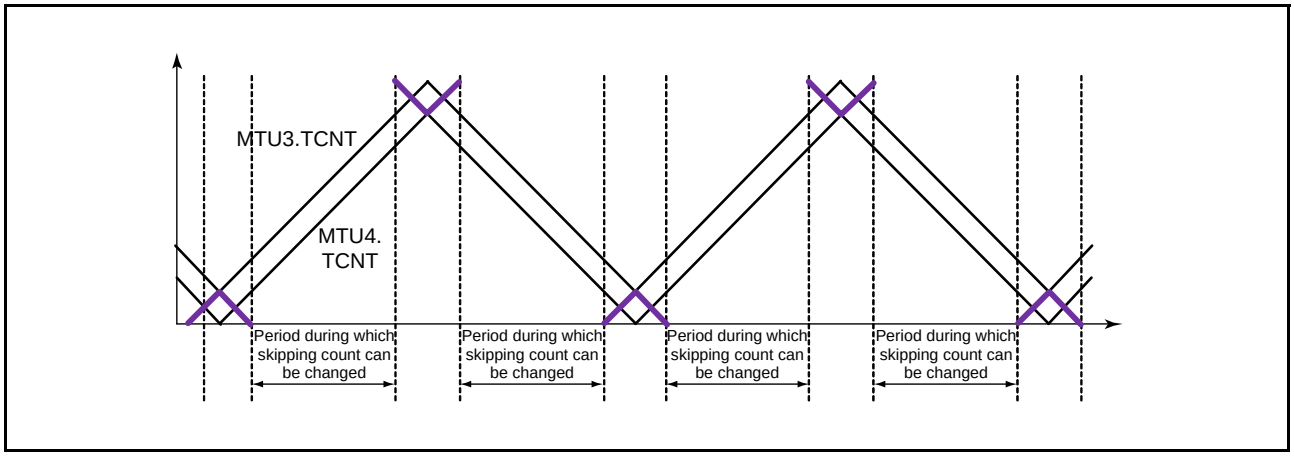


Figure 21.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 21.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR bits and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

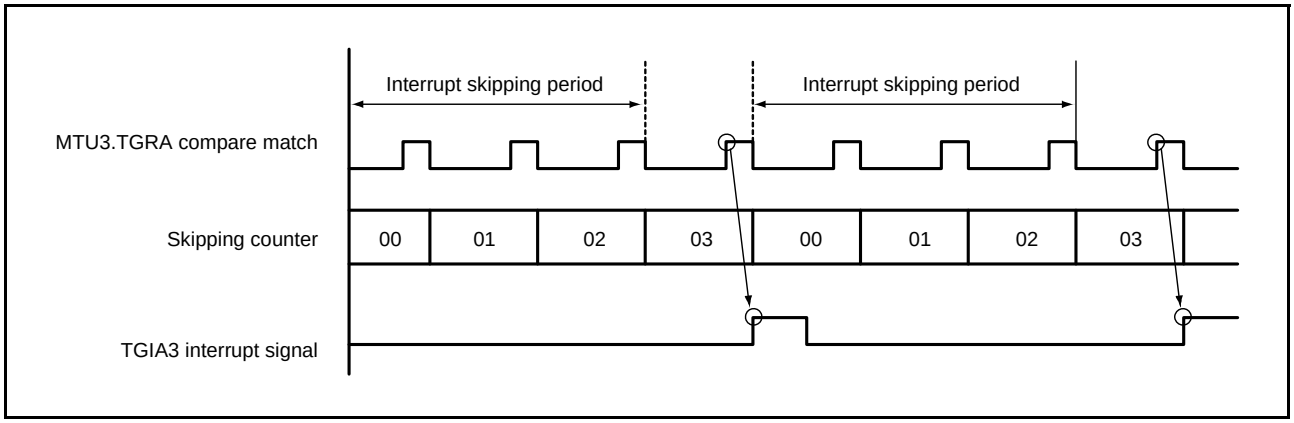


Figure 21.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

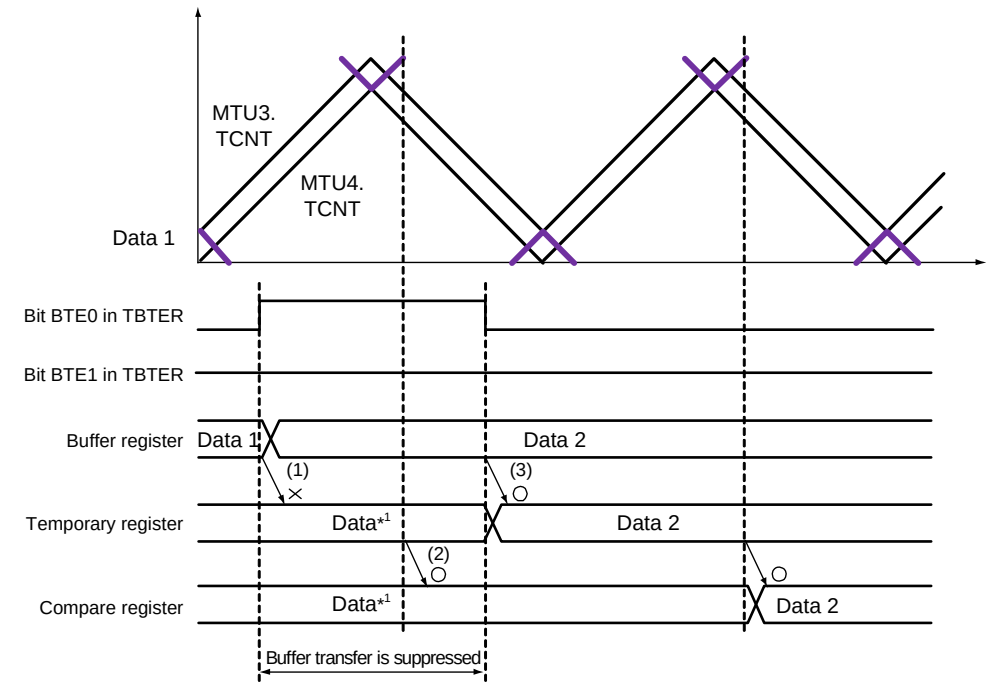
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the timer buffer transfer set register (TBTER).

Figure 21.70 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 21.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register to the temporary register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 21.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR 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 BTE1 bit in the timer buffer transfer set register (TBTER) 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 21.70 Example of Operation when Buffer Transfer is Disabled (BTE[1:0] = 01b)

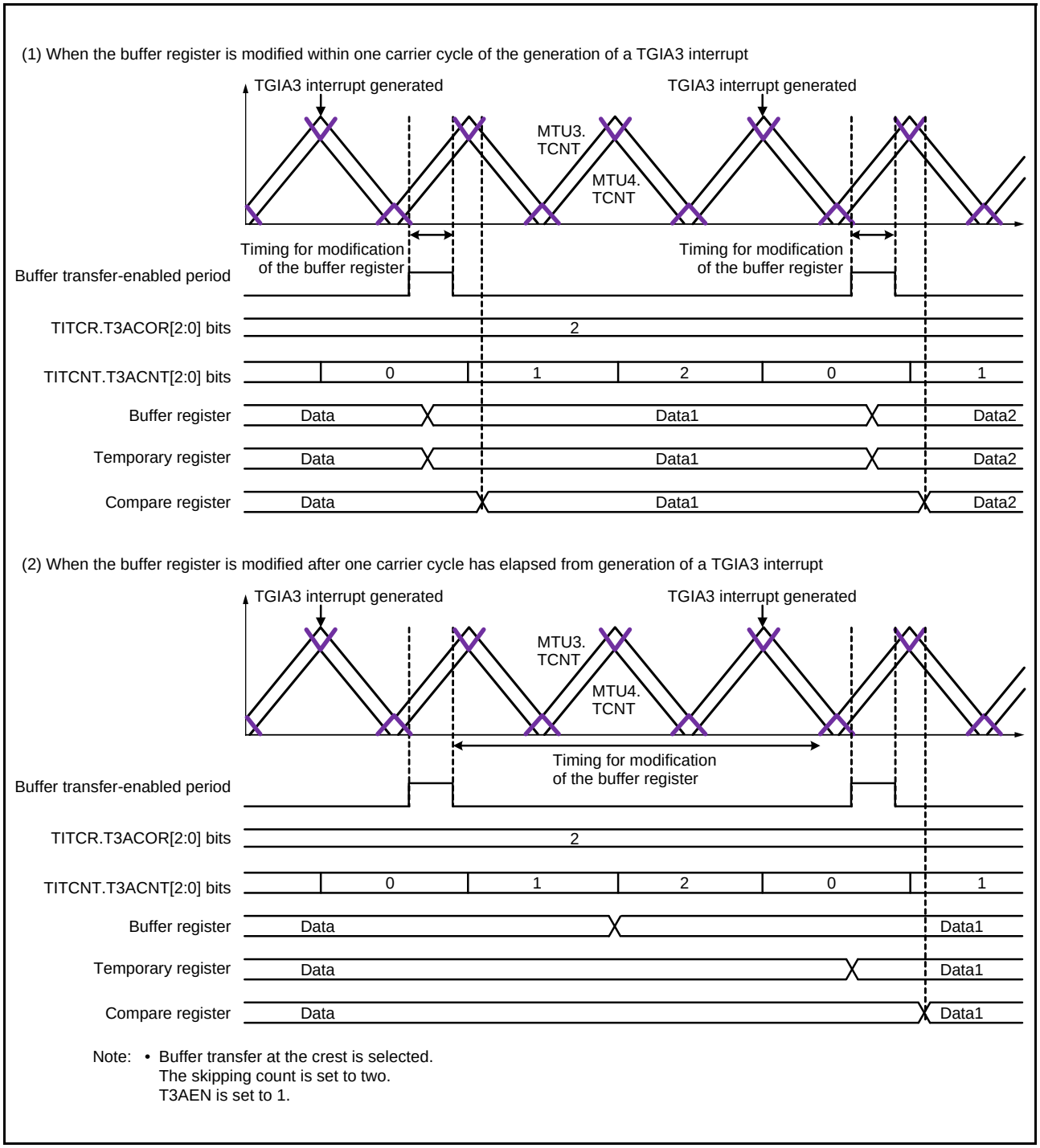


Figure 21.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

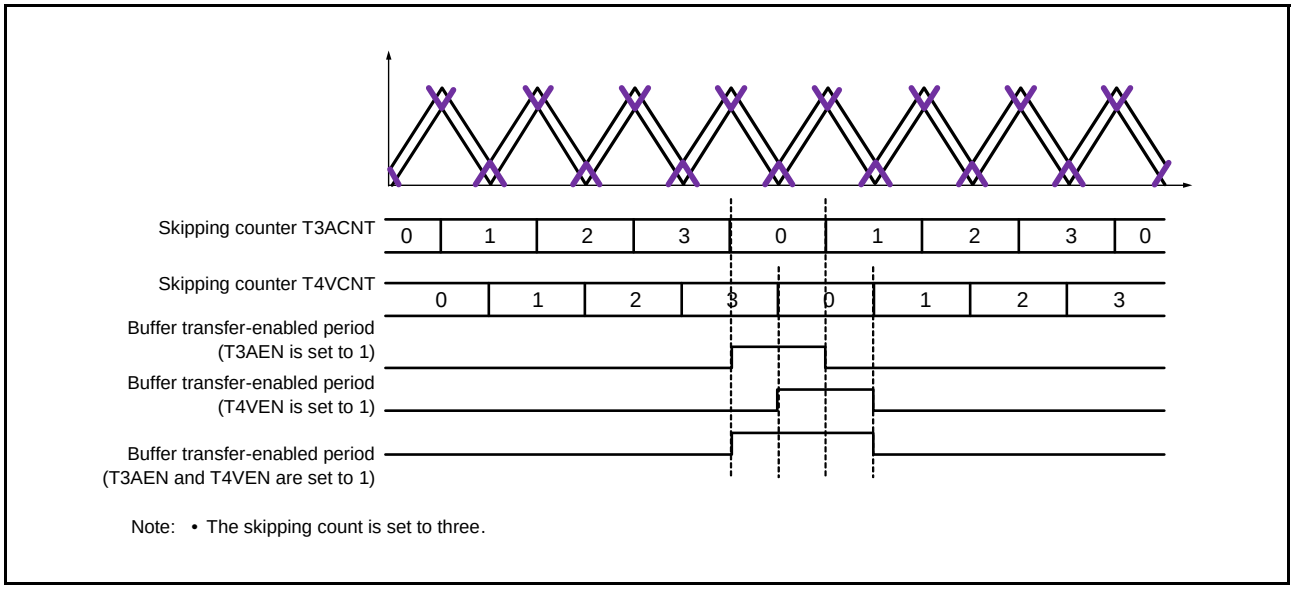


Figure 21.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(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

With the exception of the buffer registers, which can be modified at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). 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 by External Signal

The 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.
See section 22, Port Output Enable 2 (POE2a), for details.

(c) Halting of PWM Output when Oscillator is Stopped

Upon detecting that the clock input to RX220 has stopped, the 6-phase PWM output pins are automatically set to the high-impedance state. Note that the pin states are not guaranteed when the clock is restarted.
See section 9.5, Oscillation Stop Detection Function.

21.3.9 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 ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 21.73 shows an example of procedure for specifying the A/D converter start request delaying function.

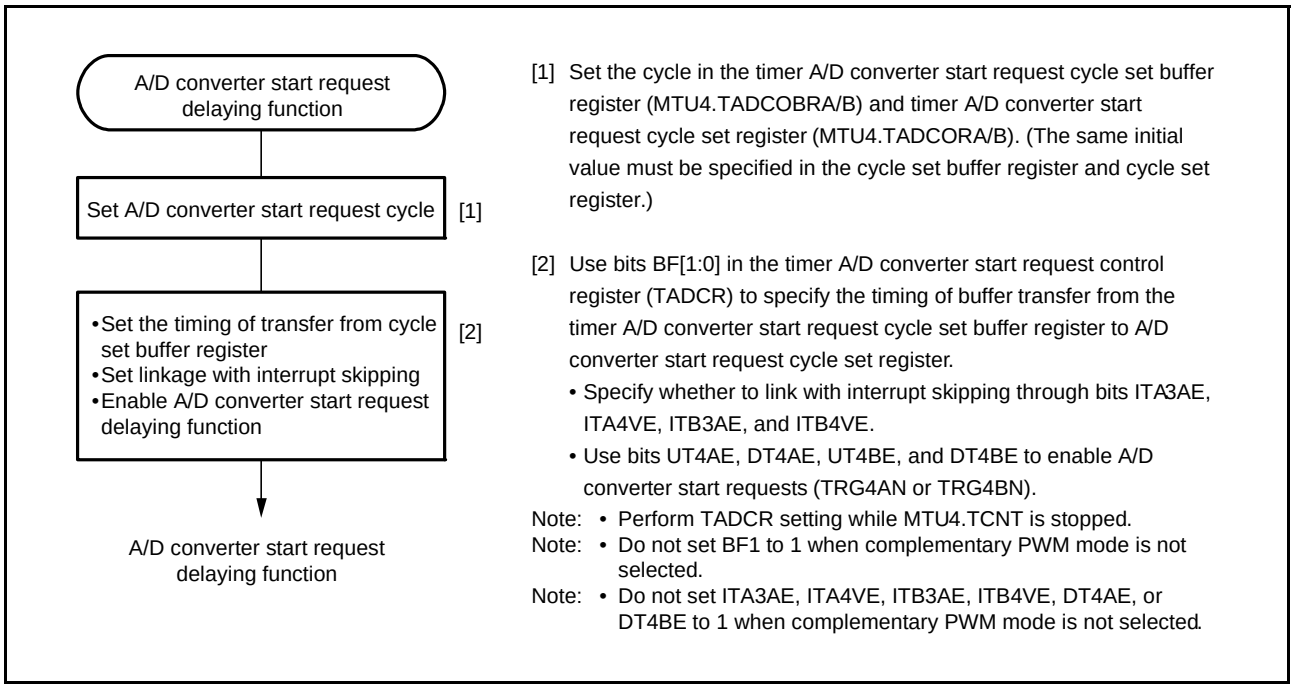


Figure 21.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 21.74 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.

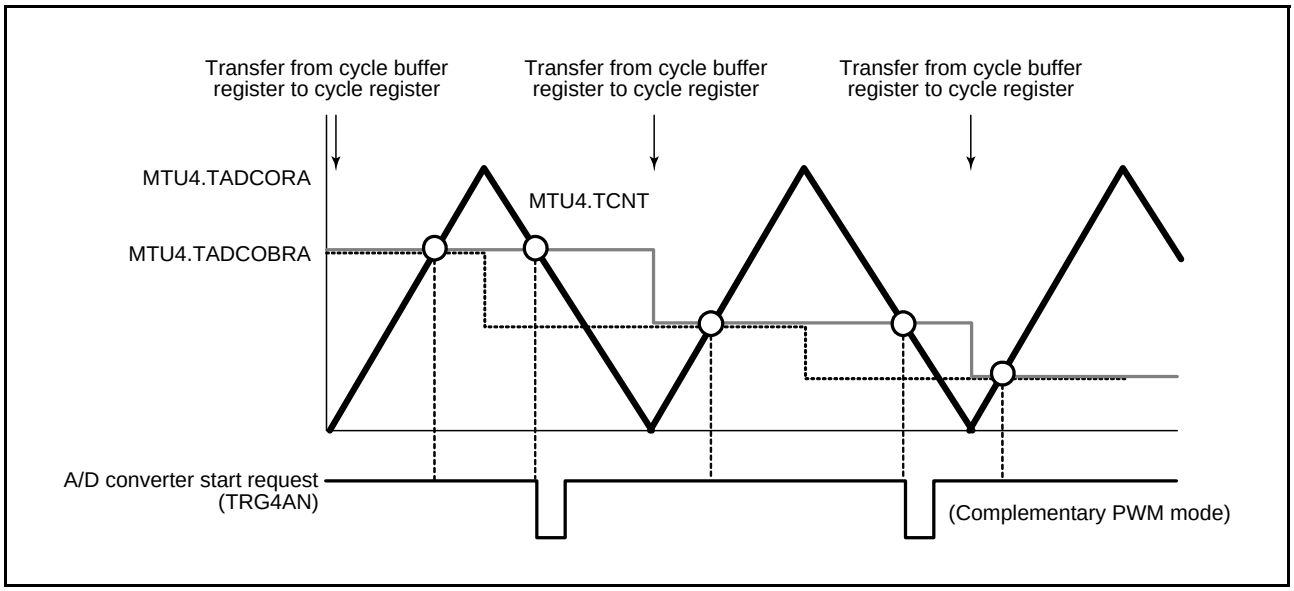


Figure 21.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) 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 BF[1:0] bits in the timer A/D converter start request control register (MTU4.TADCR).

(4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR). Figure 21.75 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 21.76 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.

Note: • This function should 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 A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.

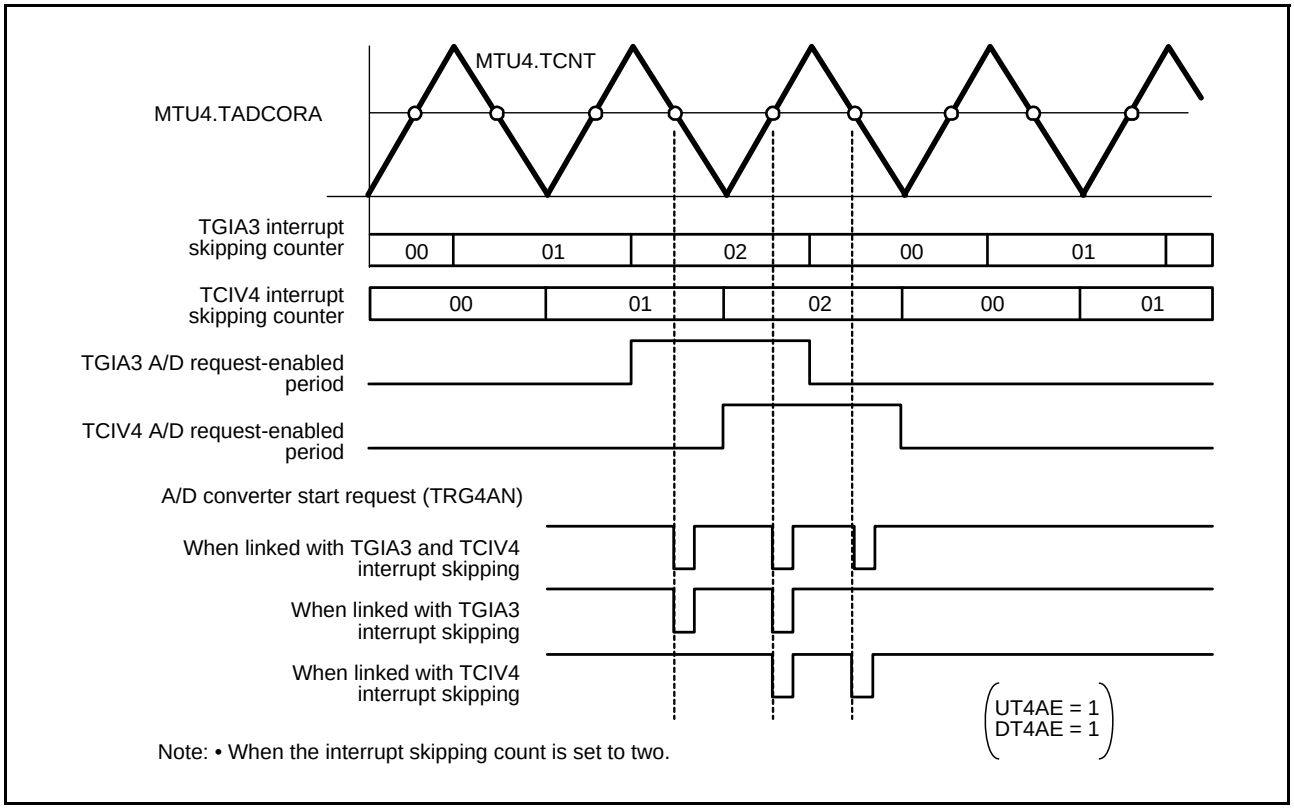


Figure 21.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)

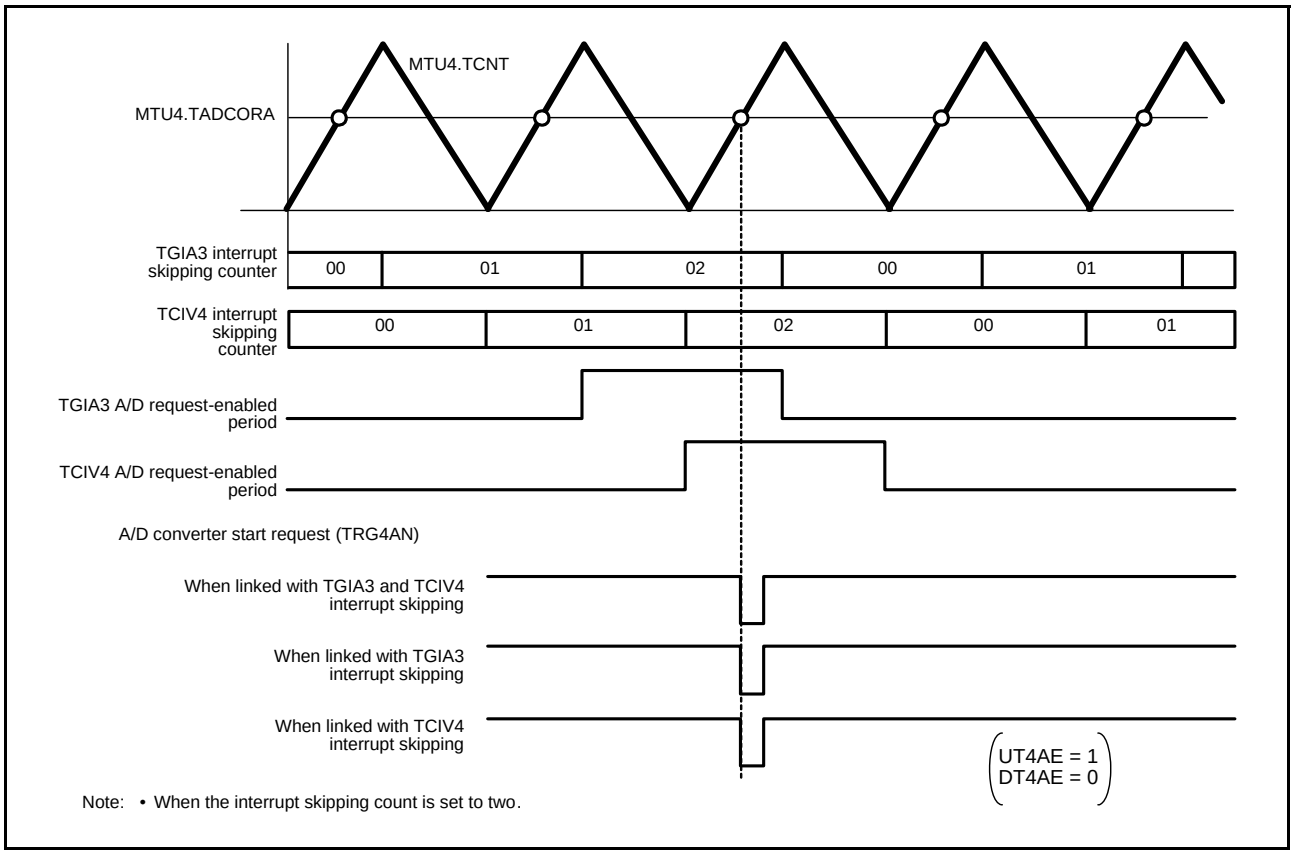


Figure 21.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)

21.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.

(1) Example of External Pulse Width Measurement Setting Procedure

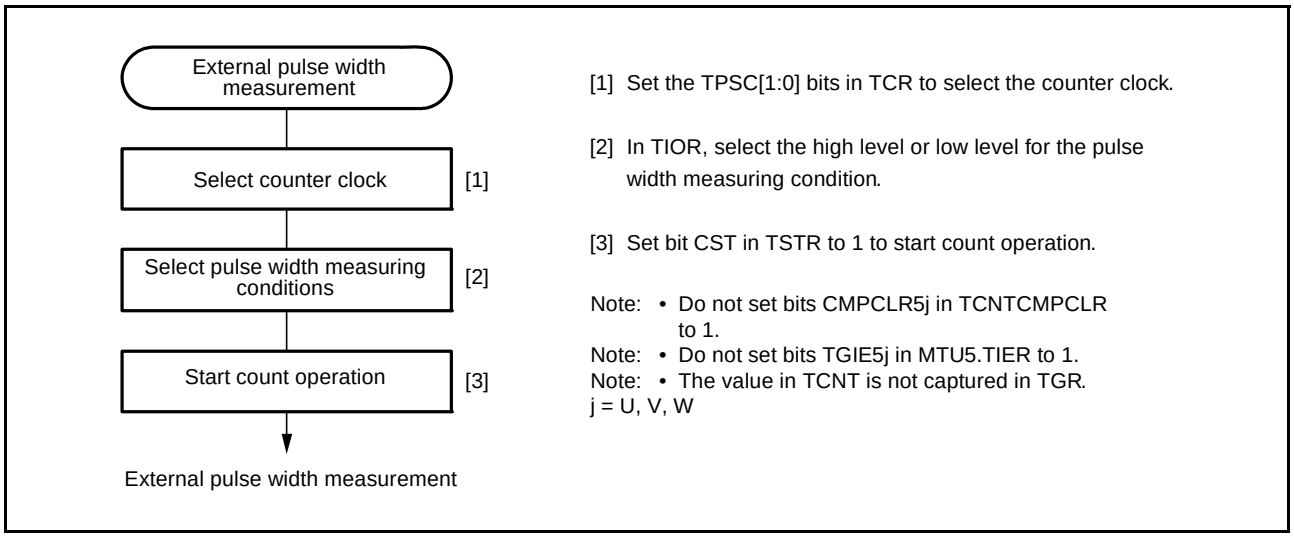


Figure 21.77 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

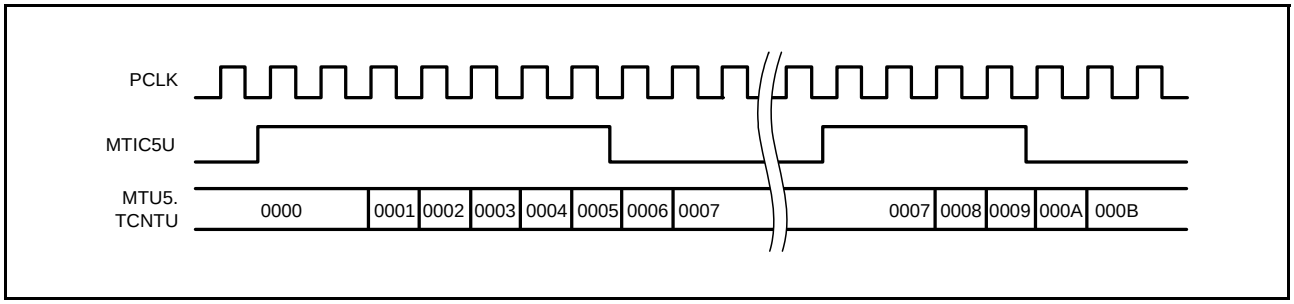


Figure 21.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

21.3.11 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function to PWM output waveform while the complementary PWM mode is in operation.

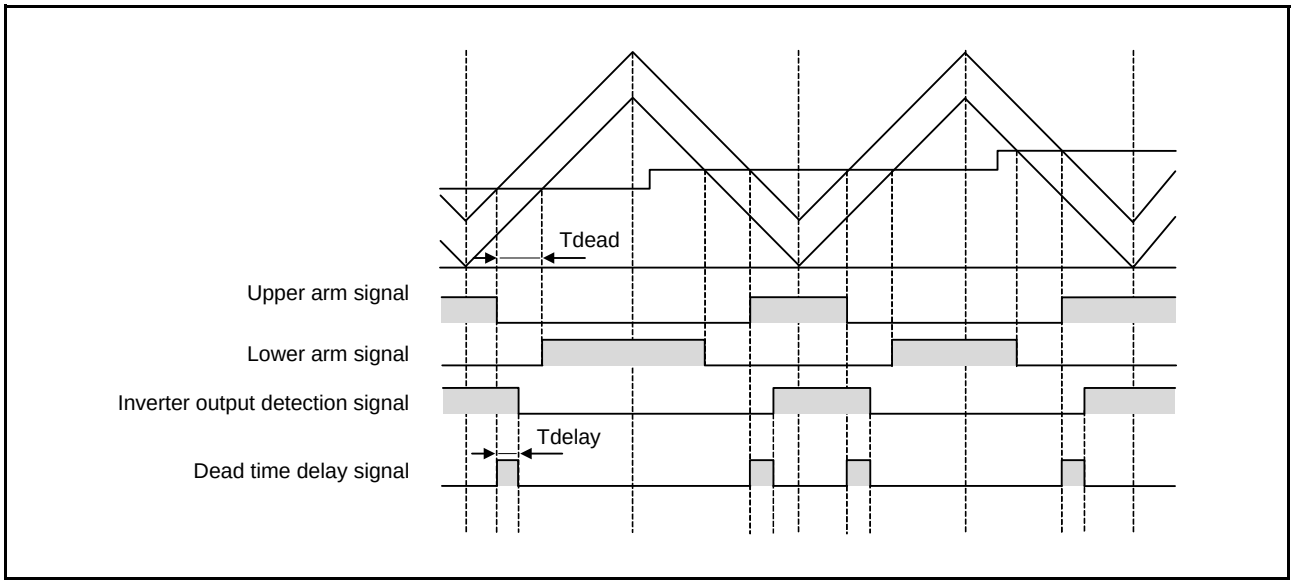
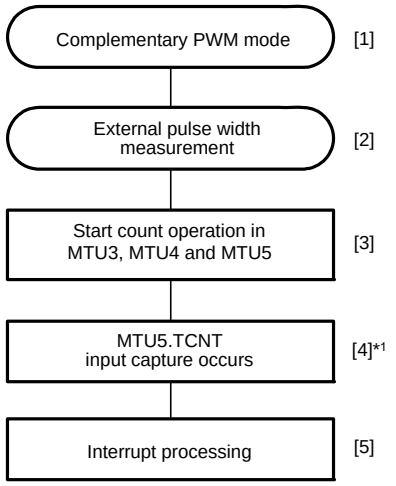


Figure 21.79 Delay in Dead Time in Complementary PWM Mode Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 21.80 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 processing.
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: • Be sure to complete PFS setting in advance.
Note 1. As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 21.80 Example of Dead Time Compensation Setting Procedure

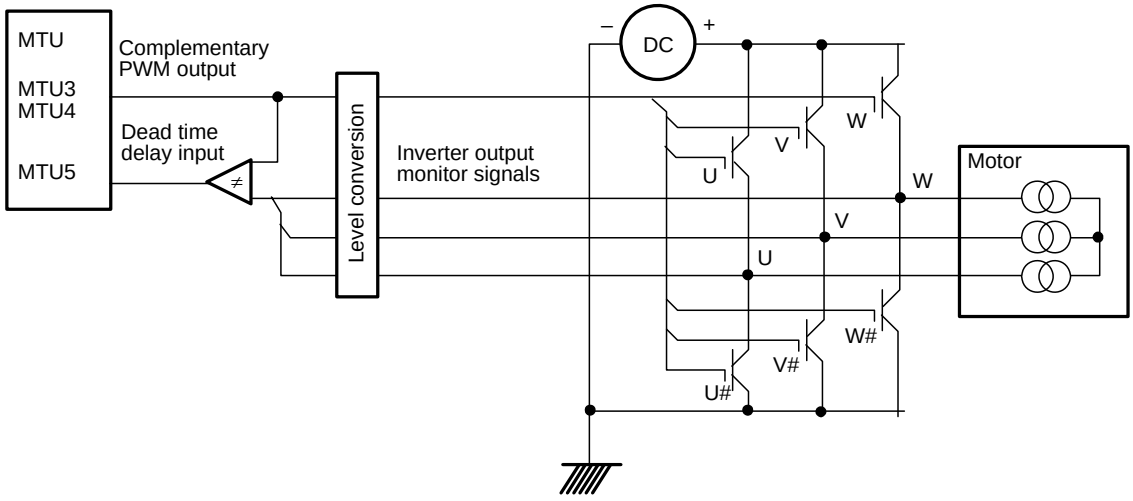


Figure 21.81 Example of Motor Control Circuit Configuration

(2) TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

The MTU5.TCNT value is captured in MTU5.TGR at either the crest or trough or at both the crest and trough during complementary PWM mode operation. The timing for capturing in MTU5.TGR can be selected by TIOR.
Figure 21.82 shows operation for the capture of MTU5.TCNT on crests and in troughs in complementary PWM mode.

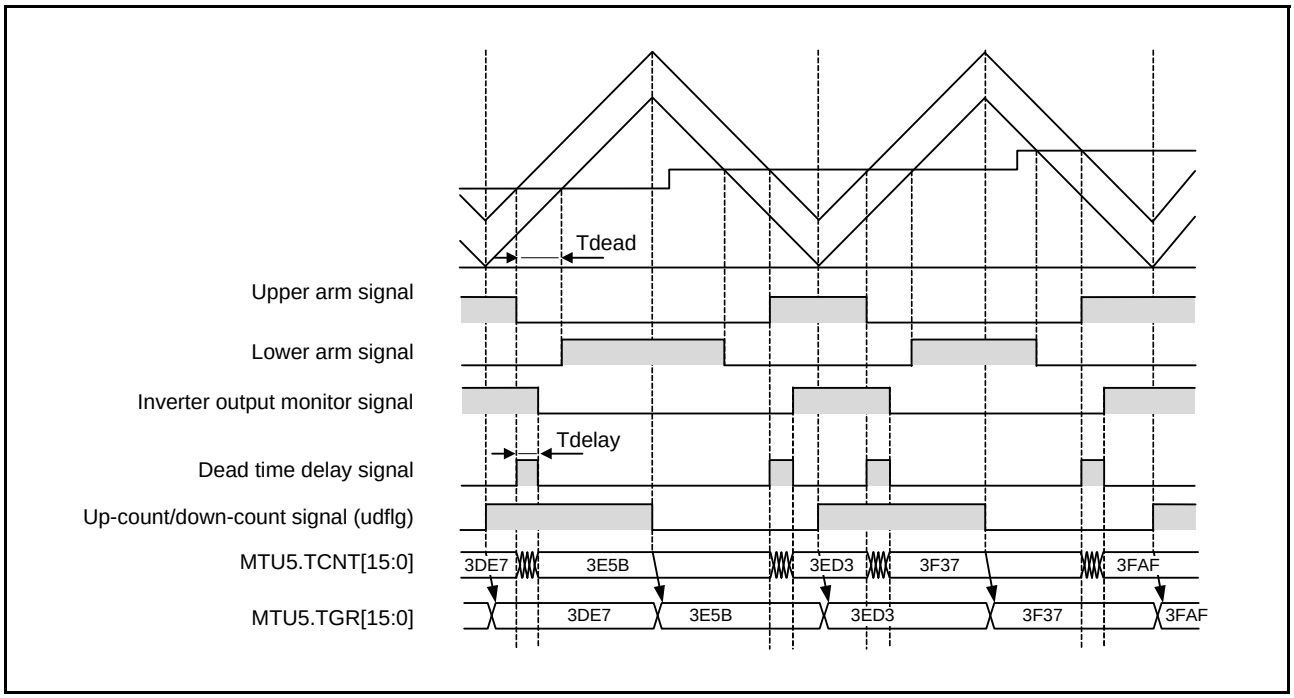


Figure 21.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

21.3.12 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 21.83 shows the timing of noise filtering.

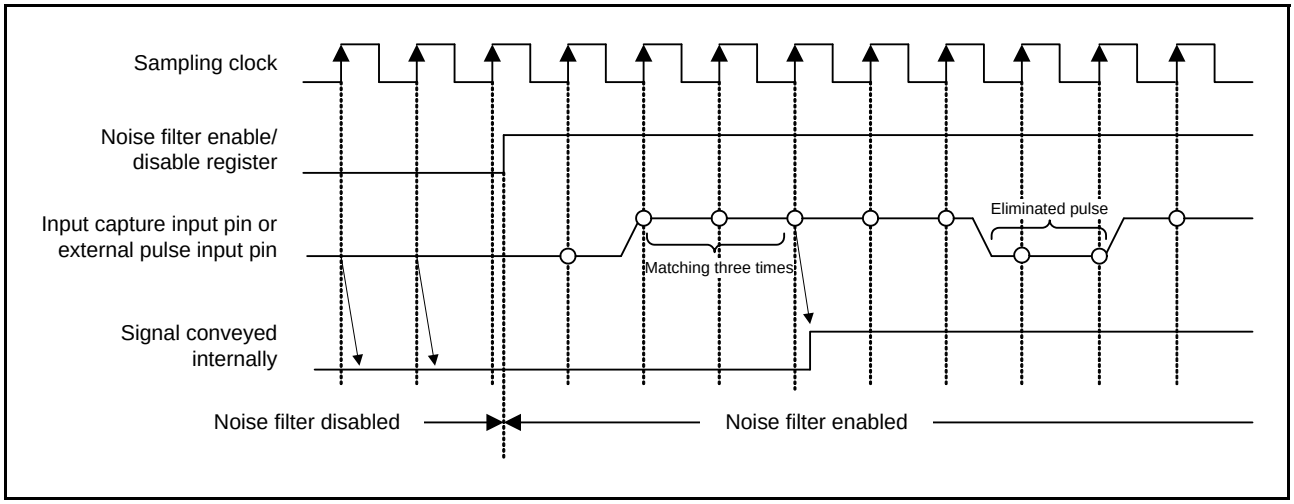


Figure 21.83 Timing of Noise Filtering

21.4 Interrupt Sources

21.4.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is 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, see section 14, Interrupt Controller (ICUb).

Table 21.57 lists the MTU interrupt sources.

Table 21.57 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 TGIE bit in TIER 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 TCIEV bit in TIER 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 TCIEU bit in TIER is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

21.4.2 DTC and 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, see section 17, 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, see section 16, DMA Controller (DMACA).
The MTU provides a total of five TGRA 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, the request for DMAC transfer may be kept pending for a certain period even after the activation source is cleared depending on the internal bus state.

21.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 21.58 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 TTGE2 bit in MTU4.TIER 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 TTGE bit in TIER is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the TTGE2 bit in MTU4.TIER 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

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 MTU0.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 UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 21.3.9, 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 21.58 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 or TADCORB and MTU4.TCNT		TRG4ABN

21.5 Operation Timing

21.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 21.84 and Figure 21.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 21.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 21.87 shows the TCNT count timing in external clock operation (phase counting mode).

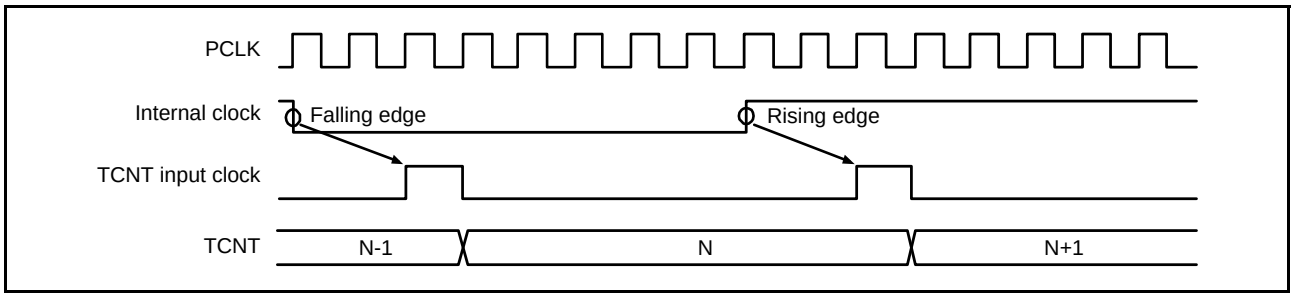


Figure 21.84 Count Timing in Internal Clock Operation (MTU0 to MTU4)

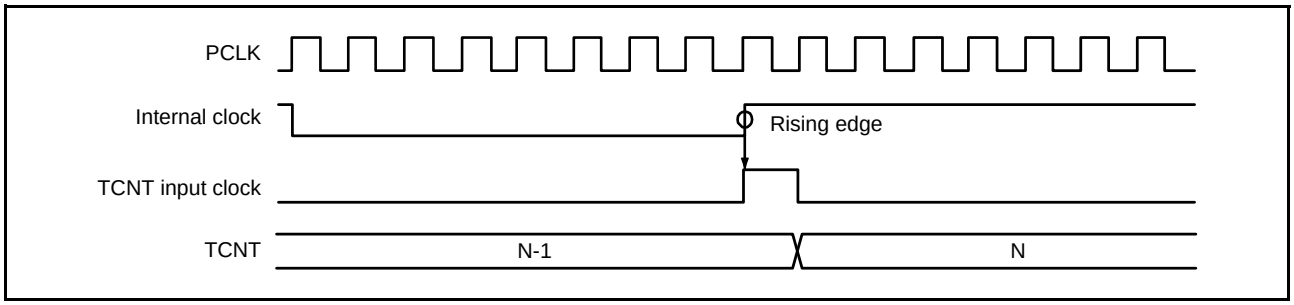


Figure 21.85 Count Timing in Internal Clock Operation (MTU5)

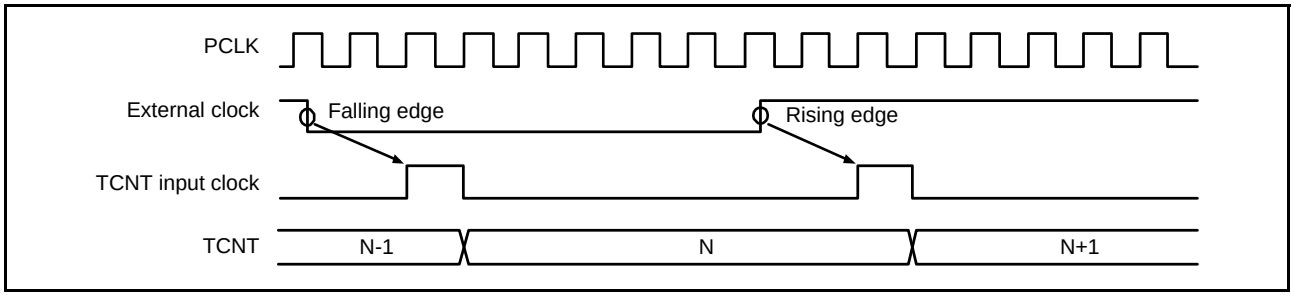


Figure 21.86 Count Timing in External Clock Operation (MTU0 to MTU 4)

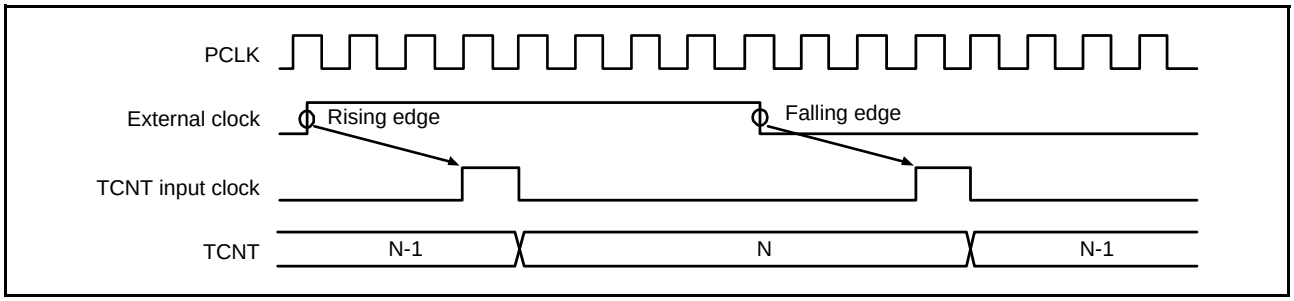


Figure 21.87 Count Timing in External Clock Operation (Phase Counting Mode)

(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 21.88 shows the output compare output timing (normal mode or PWM mode) and Figure 21.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

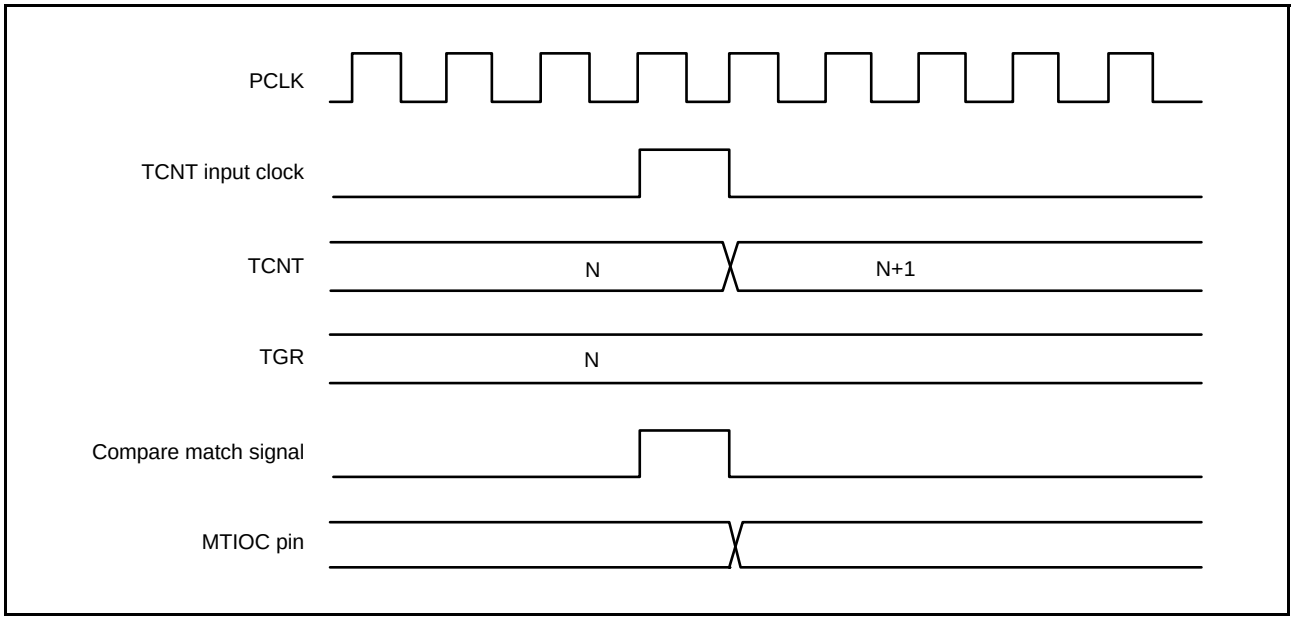


Figure 21.88 Output Compare Output Timing (Normal Mode or PWM Mode)

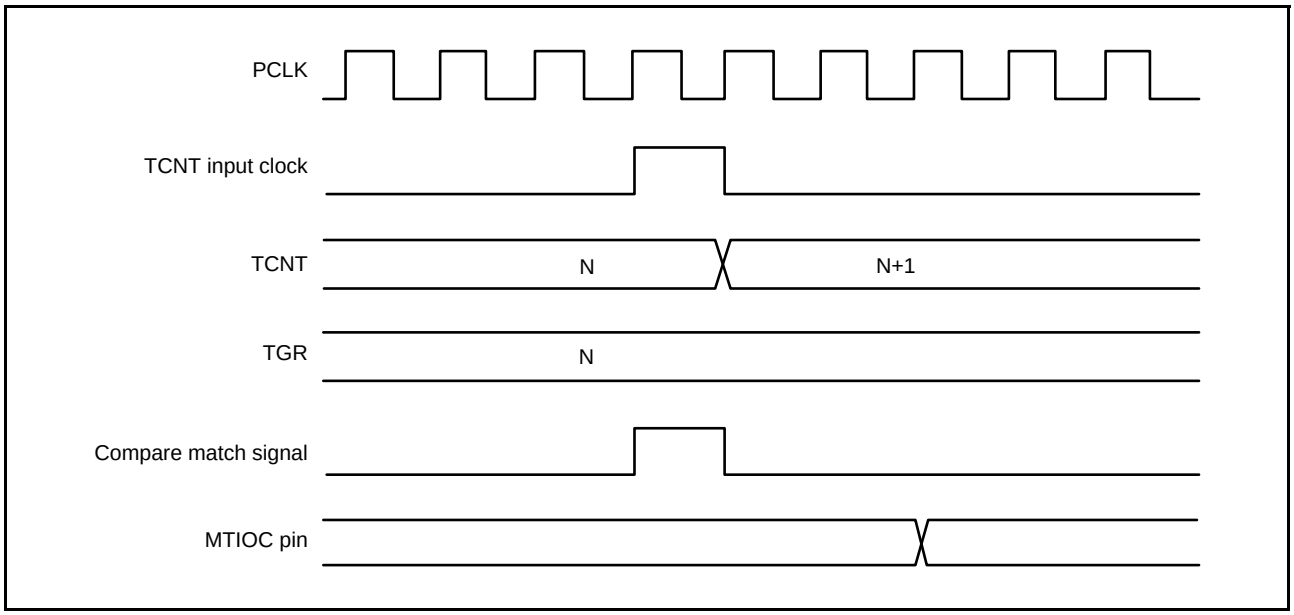


Figure 21.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 21.90 shows the input capture signal timing.

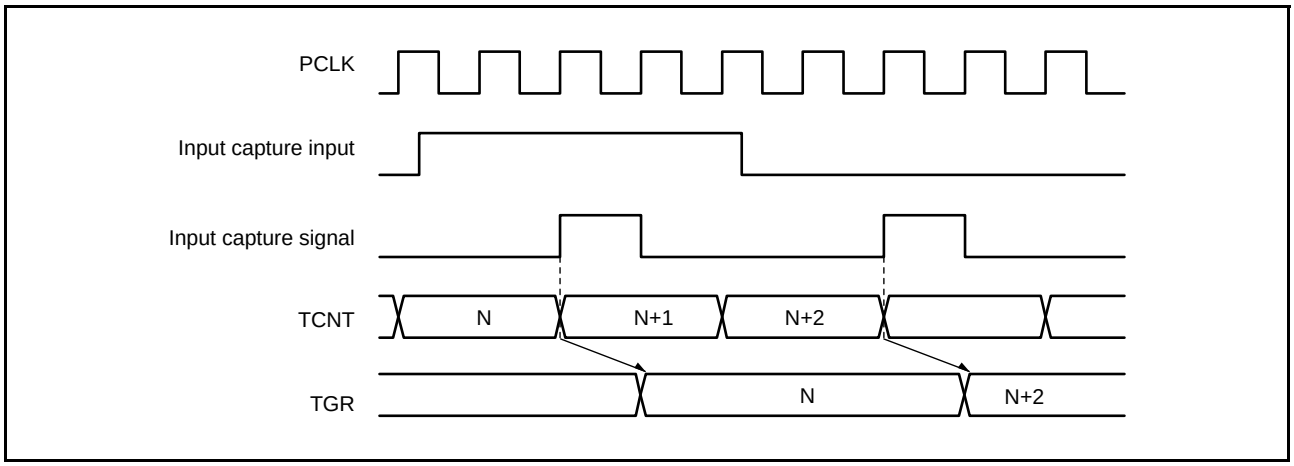


Figure 21.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 21.91 and Figure 21.92 show the timing when counter clearing on compare match is specified, and Figure 21.93 shows the timing when counter clearing on input capture is specified.

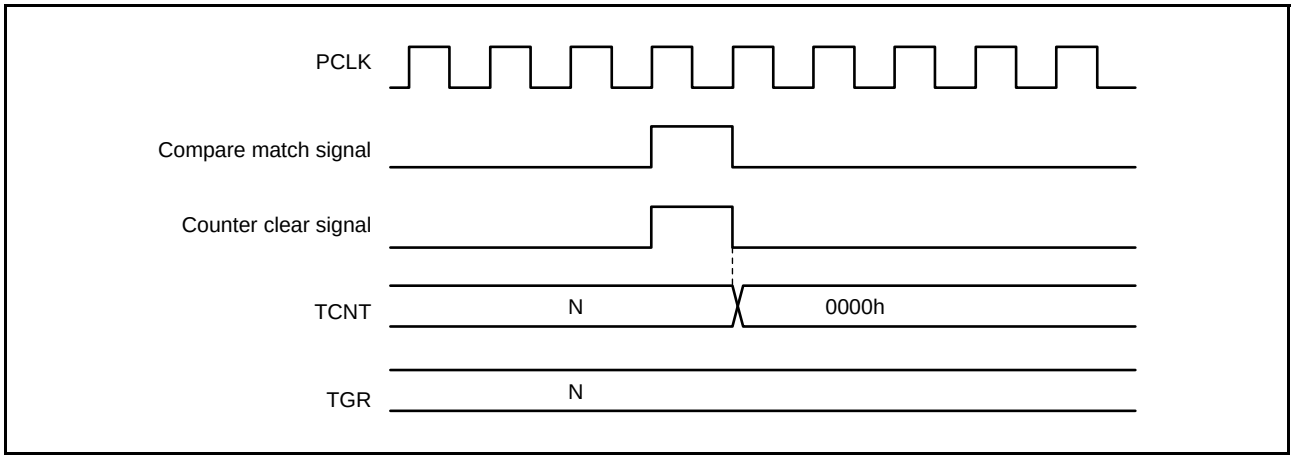


Figure 21.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

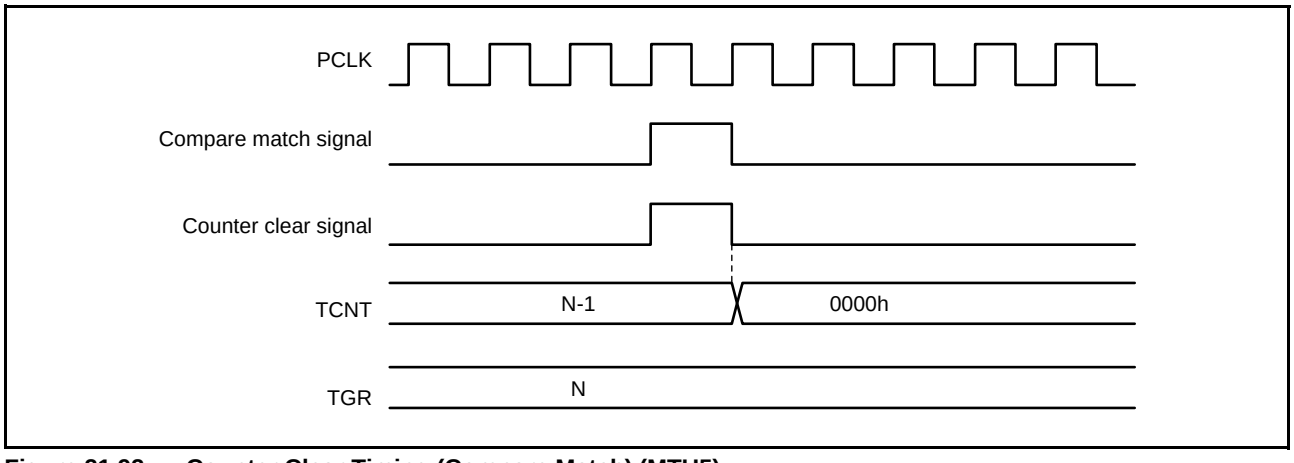


Figure 21.92 Counter Clear Timing (Compare Match) (MTU5)

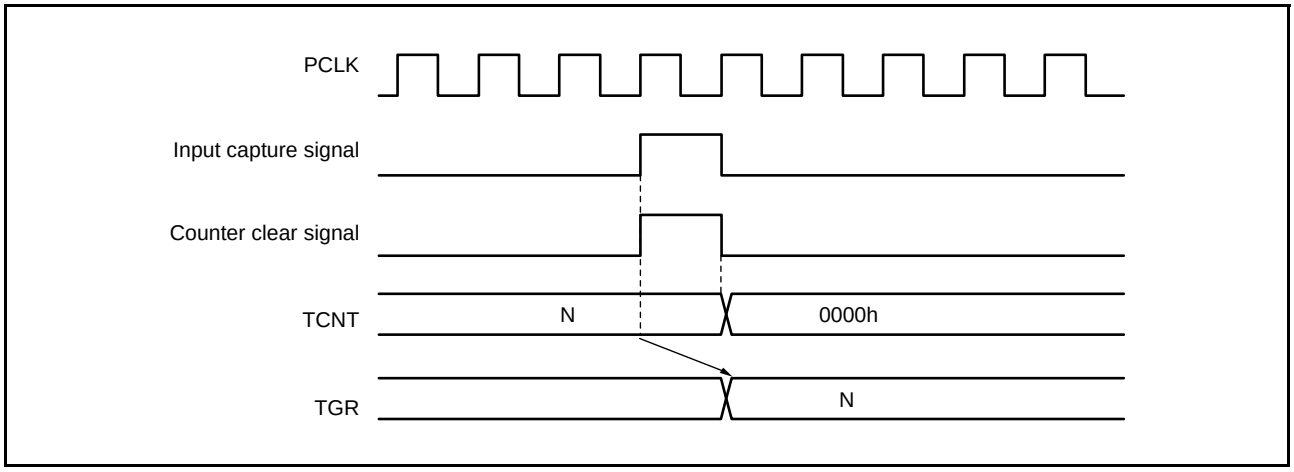


Figure 21.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 21.94 to Figure 21.96 show the timing in buffer operation.

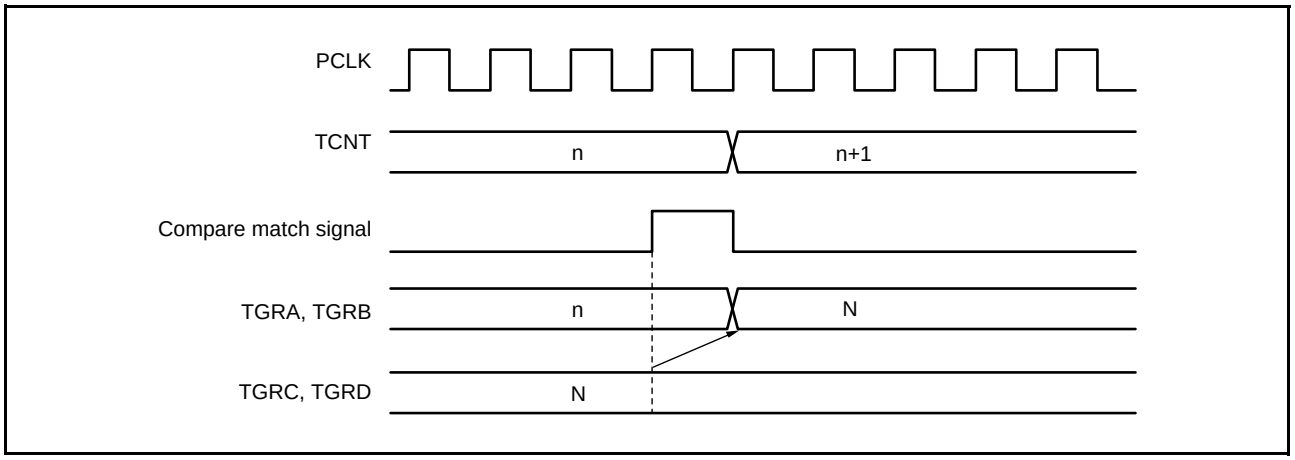


Figure 21.94 Buffer Operation Timing (Compare Match)

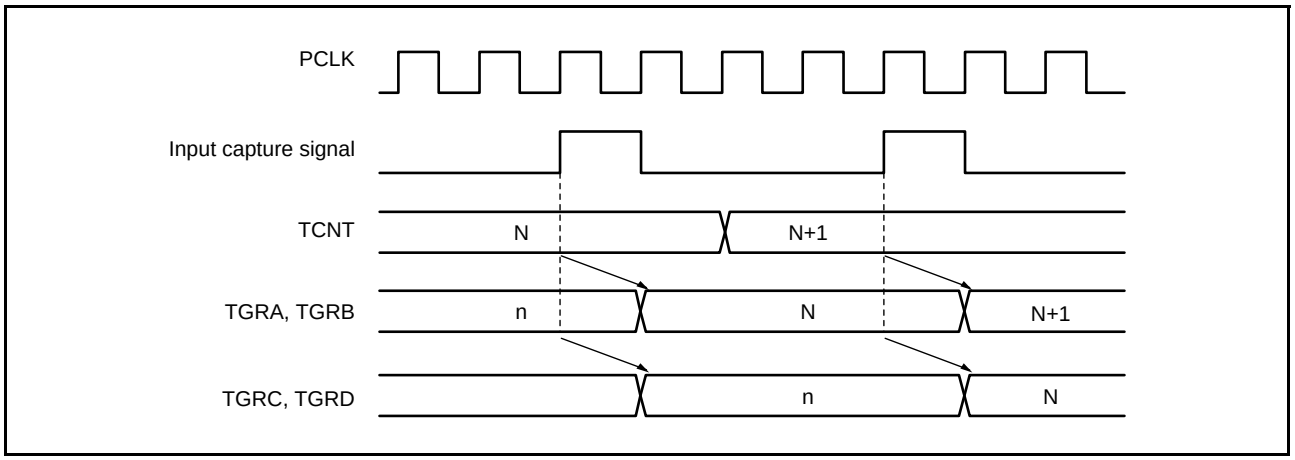


Figure 21.95 Buffer Operation Timing (Input Capture)

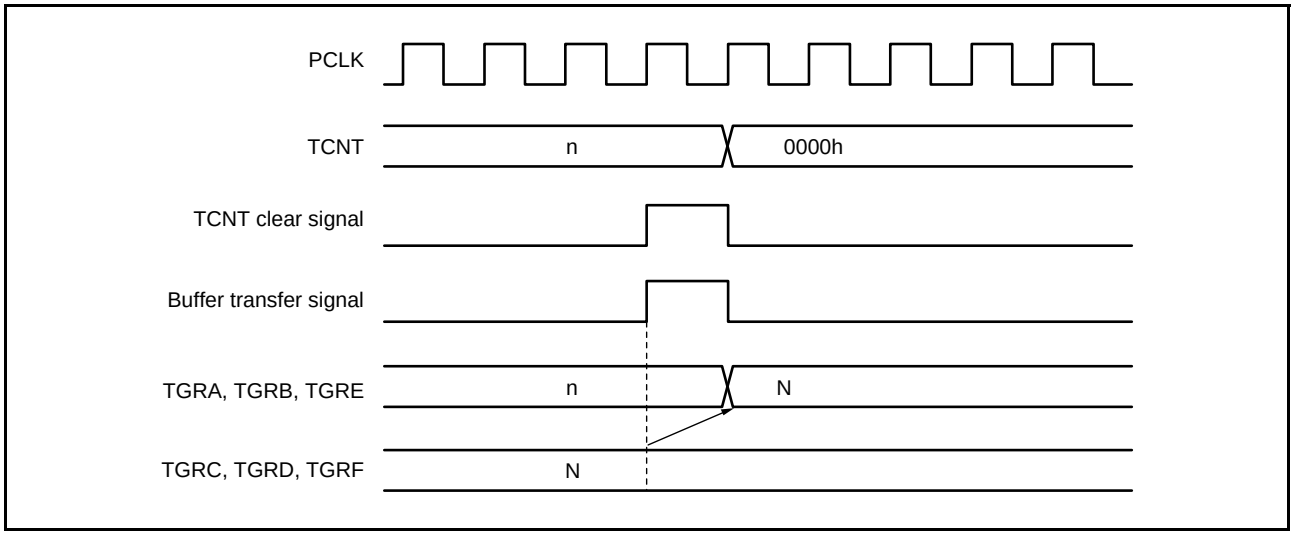


Figure 21.96 Buffer Operation Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 21.97 to Figure 21.99 show the buffer transfer timing in complementary PWM mode.

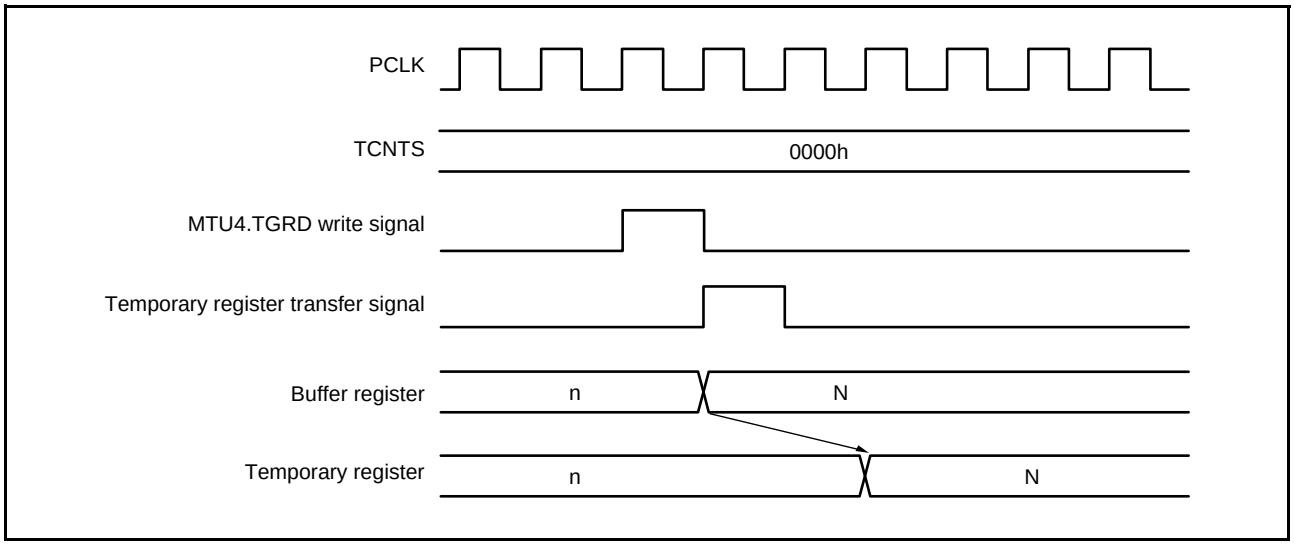


Figure 21.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

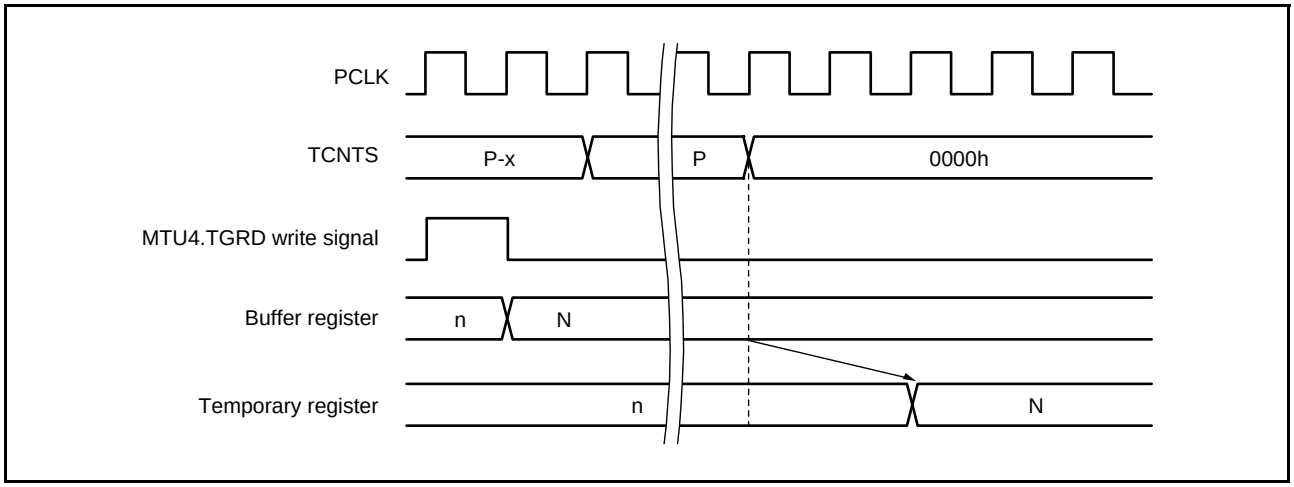


Figure 21.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

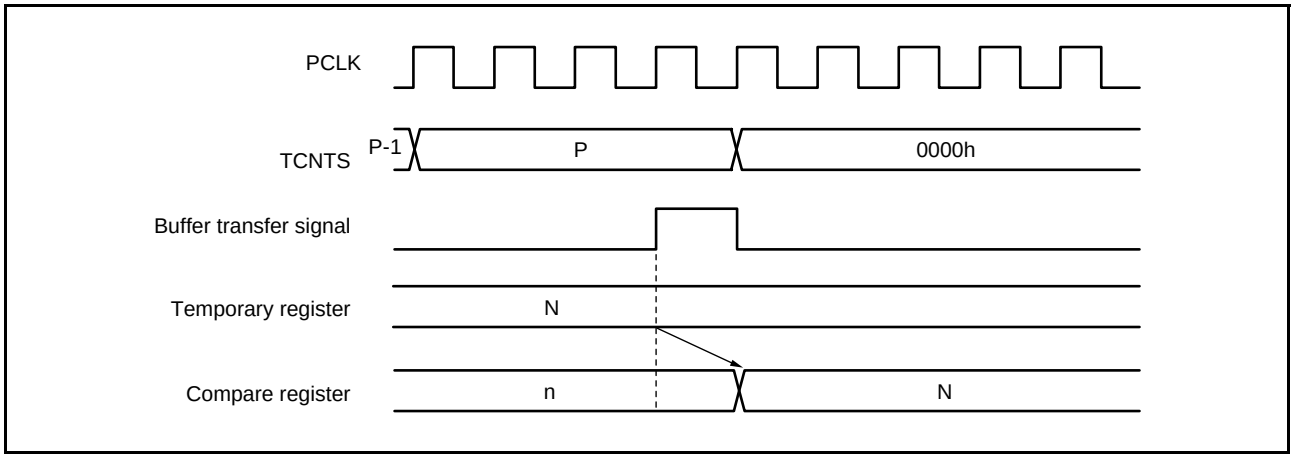


Figure 21.99 Transfer Timing from Temporary Register to Compare Register

21.5.2 Interrupt Signal Timing

(1) Timing for TGI Interrupt by Compare Match

Figure 21.100 and Figure 21.101 show the TGI interrupt request signal timing on compare match.

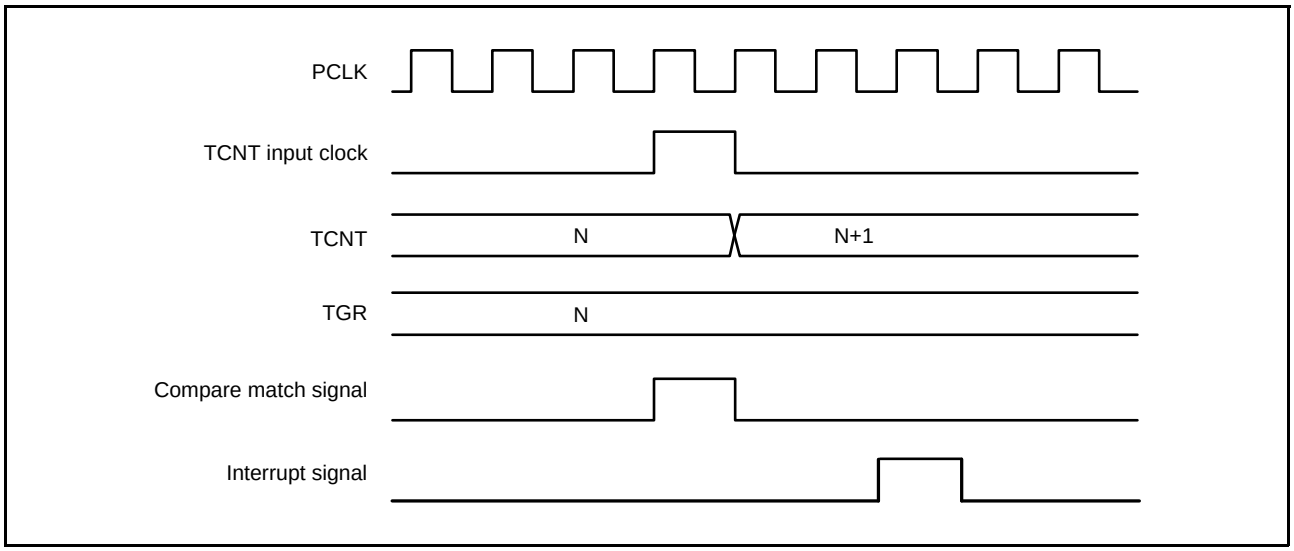


Figure 21.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

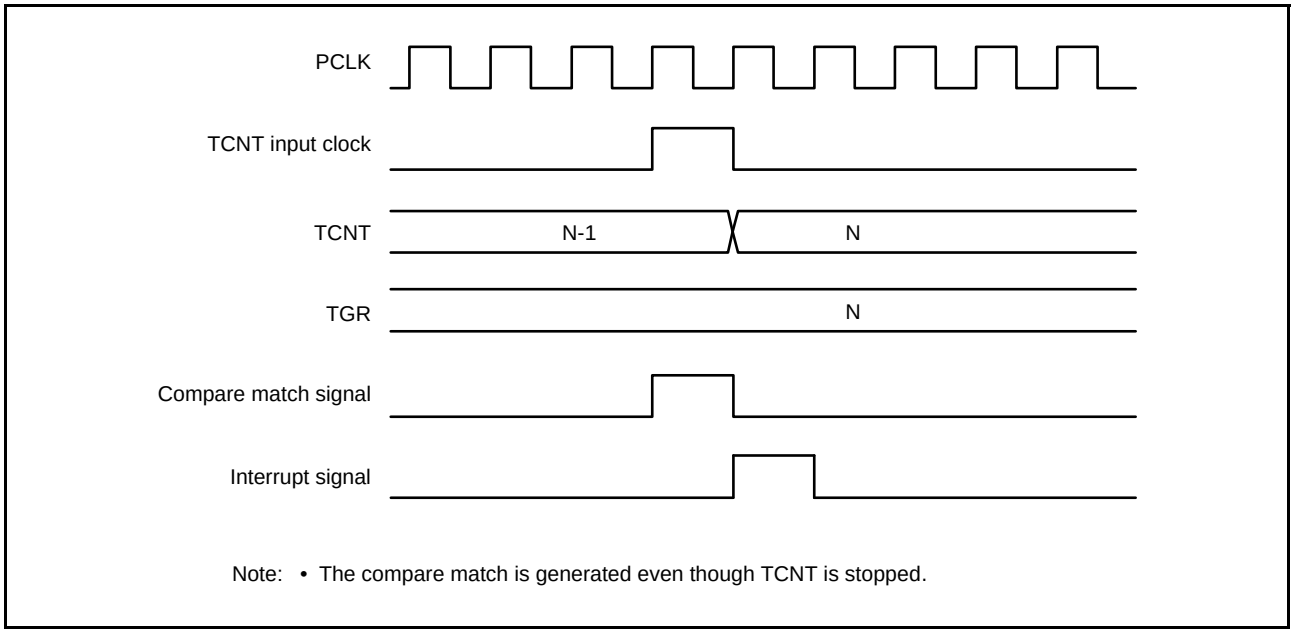


Figure 21.101 TGI Interrupt Timing (Compare Match) (MTU5)

(2) Timing for TGI Interrupt by Input Capture

Figure 21.102 and Figure 21.103 show TGI interrupt request signal timing on input capture.

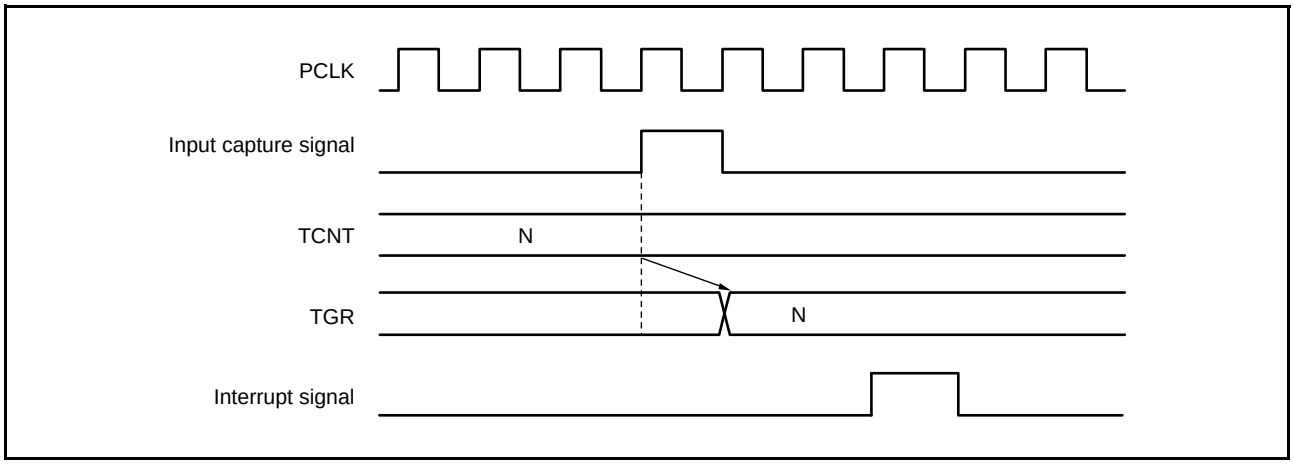


Figure 21.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

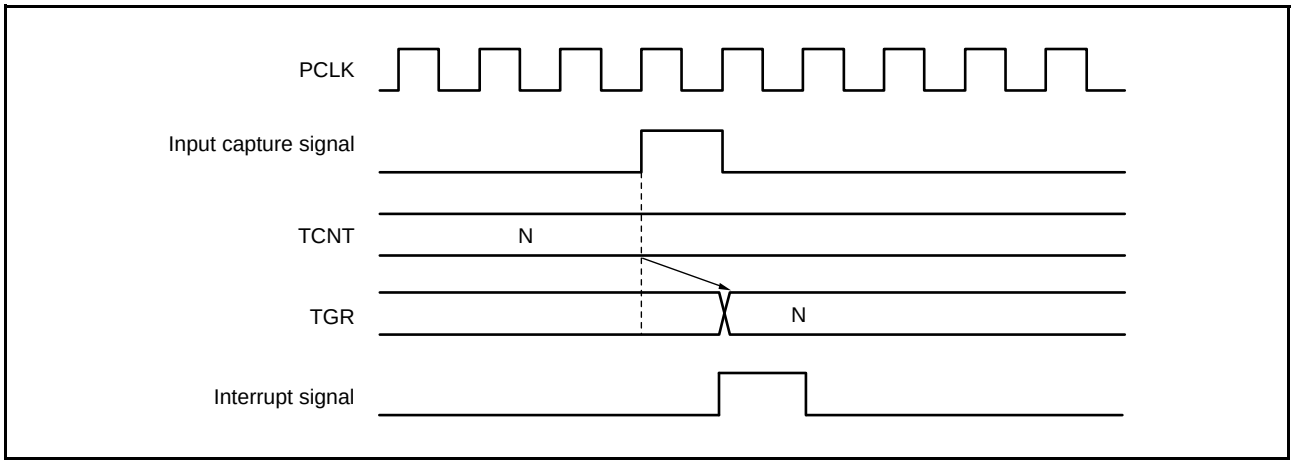


Figure 21.103 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 21.104 shows the TCIV interrupt request signal timing on overflow.
Figure 21.105 shows the TCIU interrupt request signal timing on underflow.

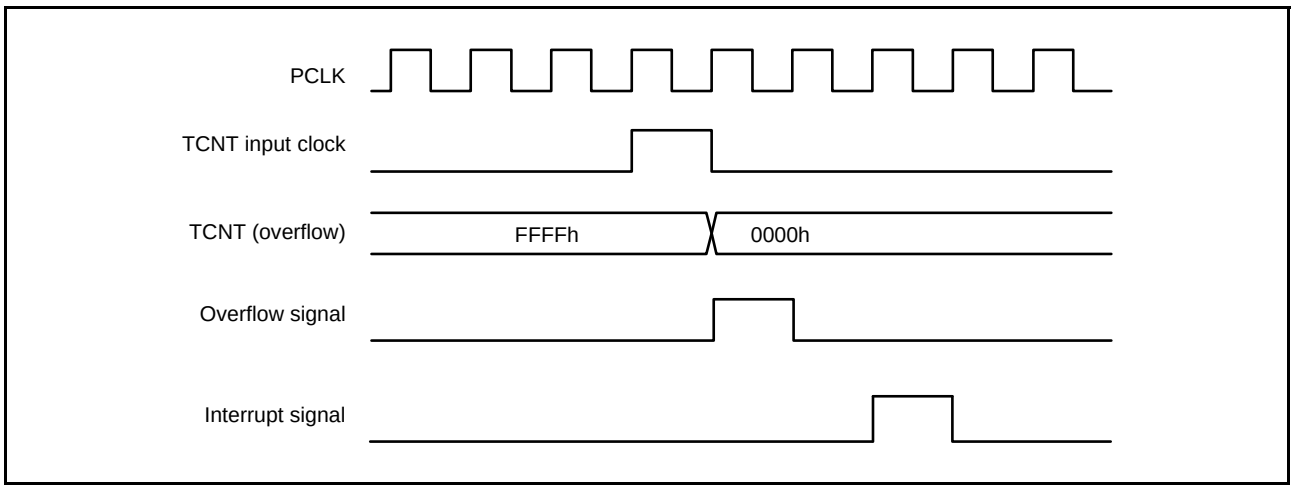


Figure 21.104 TCIV Interrupt Timing

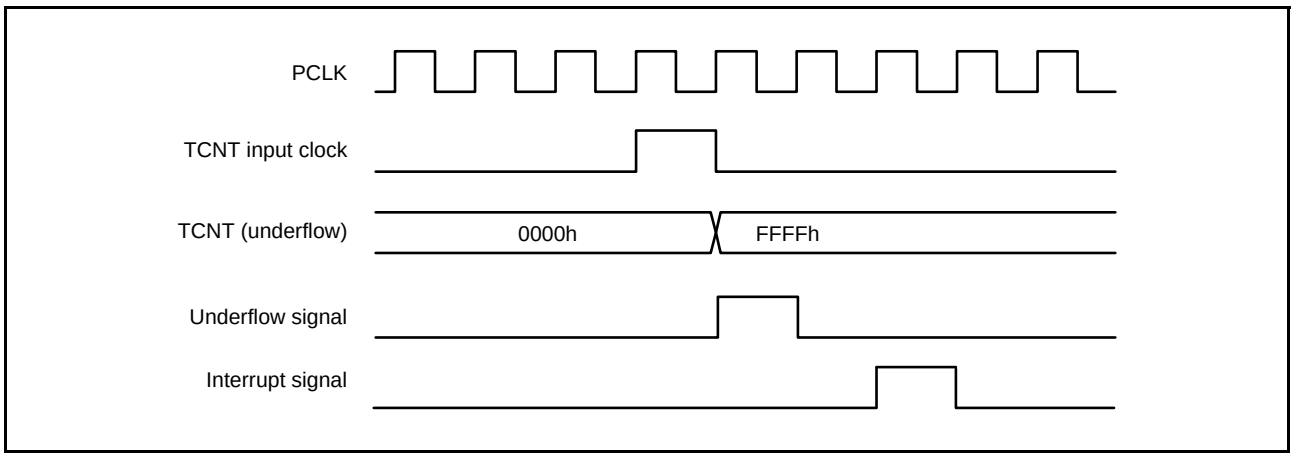


Figure 21.105 TCIU Interrupt Timing

21.6 Usage Notes

21.6.1 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 clearing the module clock stop mode. For details, refer to section 11, Low Power Consumption.

21.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK for single-edge detection, and at least 2.5 PCLK 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, and the pulse width must be at least 2.5 PCLK. Figure 21.106 shows the input clock conditions in phase counting mode.

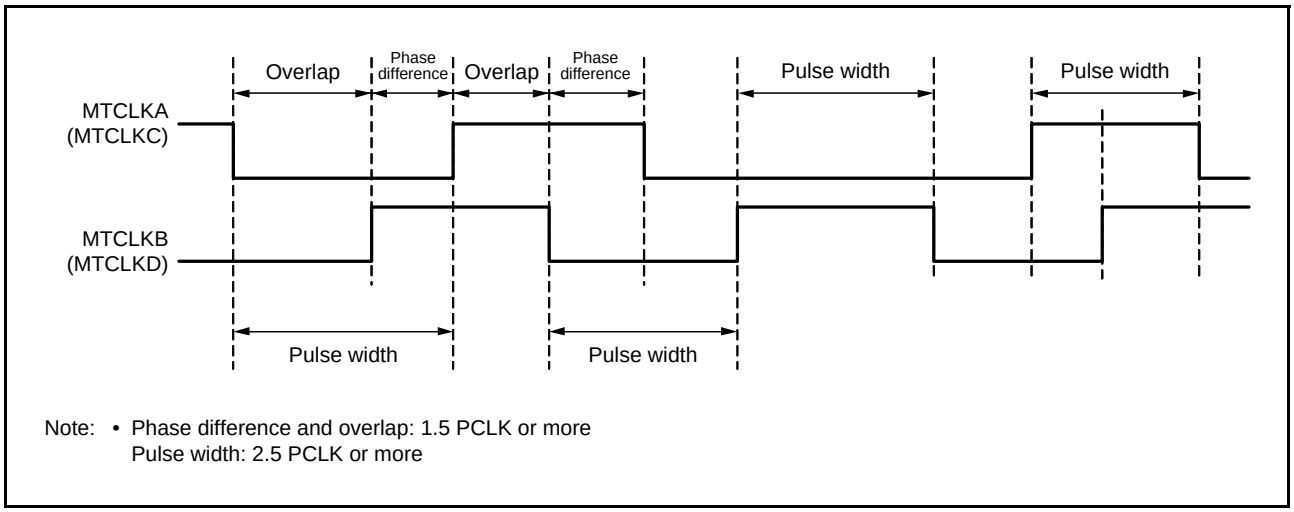


Figure 21.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

21.6.3 Note 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 counter-clock frequency set by TCR.TPSC[2:0] bits
N: TGR setting

21.6.4 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 21.107 shows the timing in this case.

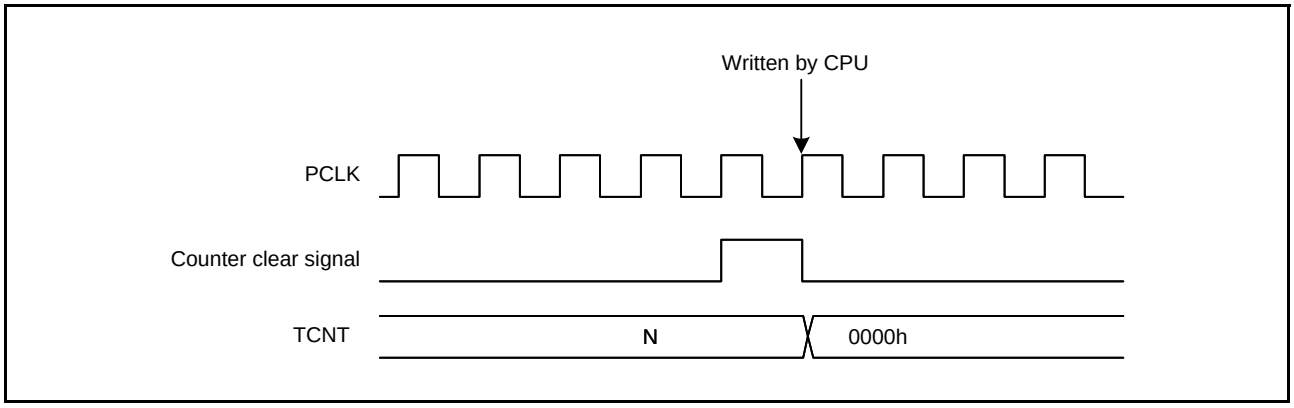


Figure 21.107 Contention between TCNT Write and Counter Clear Operations

21.6.5 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 21.108 shows the timing in this case.

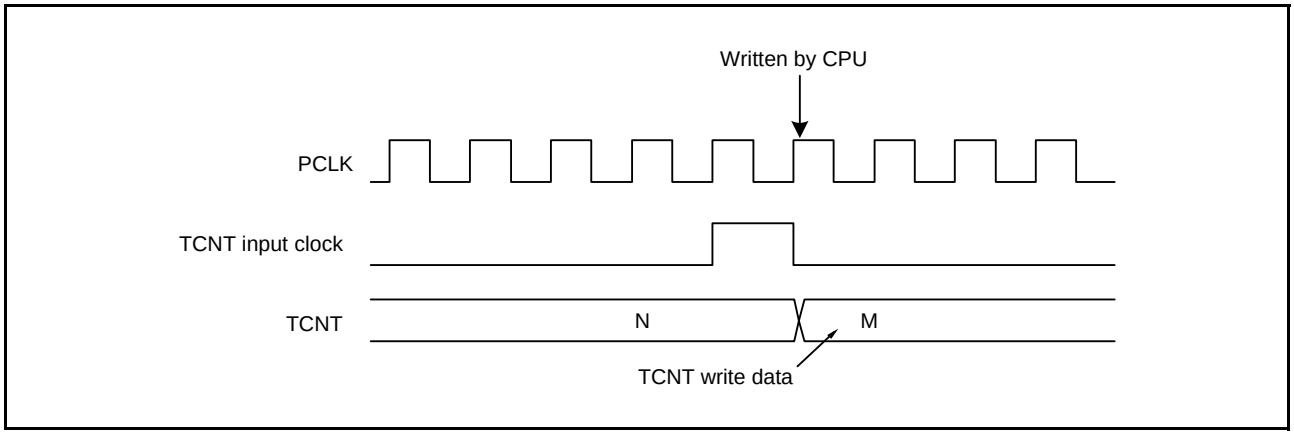


Figure 21.108 Contention between TCNT Write and Increment Operations

21.6.6 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 21.109 shows the timing in this case.

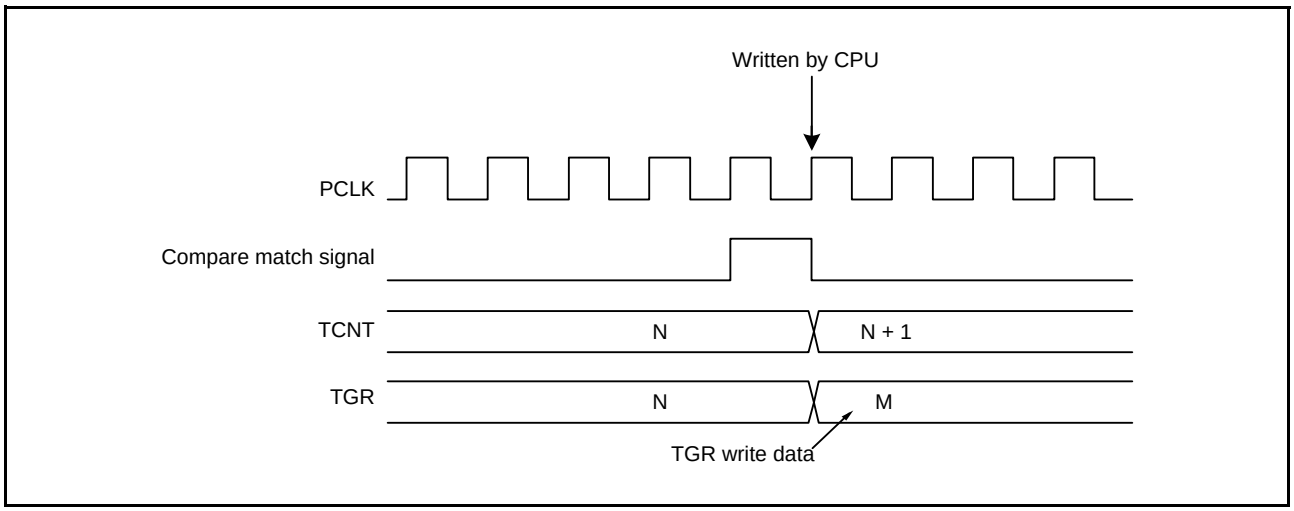


Figure 21.109 Contention between TGR Write Operation and Compare Match

21.6.7 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 21.110 shows the timing in this case.

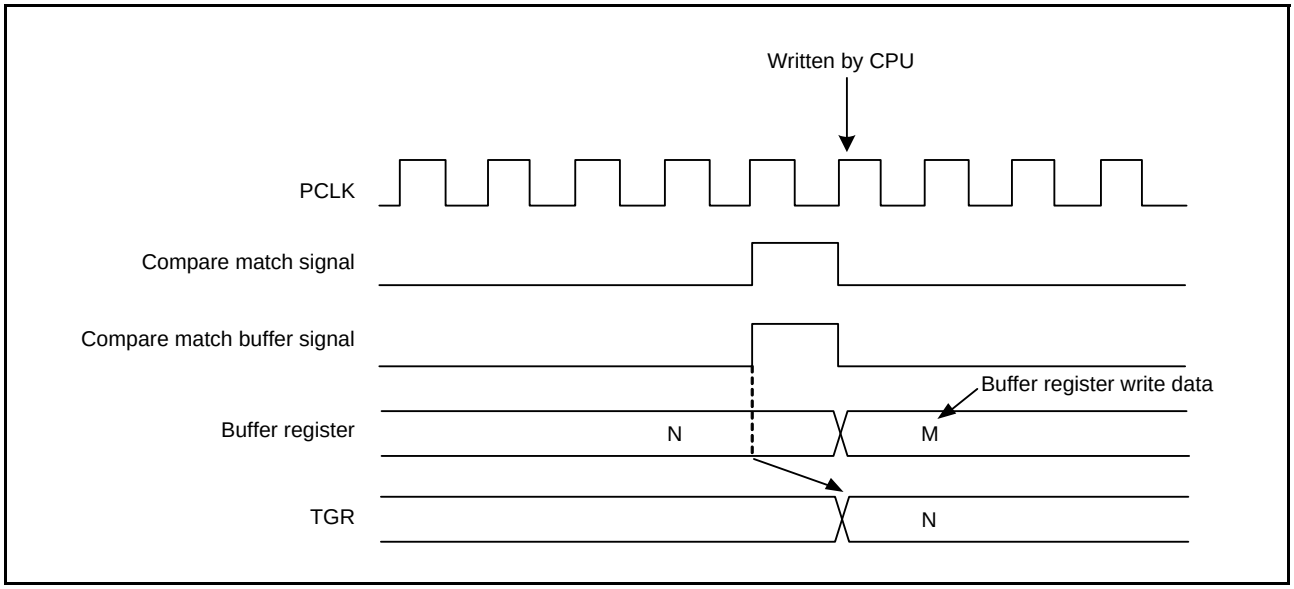


Figure 21.110 Contention between Buffer Register Write Operation and Compare Match

21.6.8 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 21.111 shows the timing in this case.

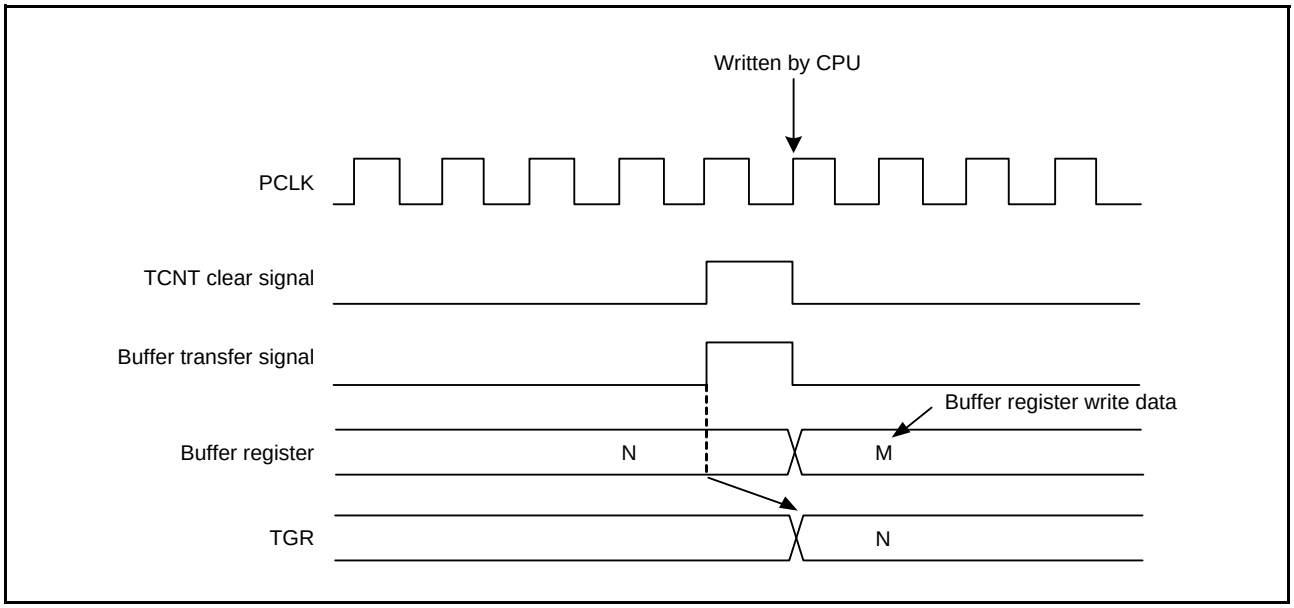


Figure 21.111 Contention between Buffer Register Write and TCNT Clear Operations

21.6.9 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 in MTU0 to MTU5.

Figure 21.112 shows the timing in this case.

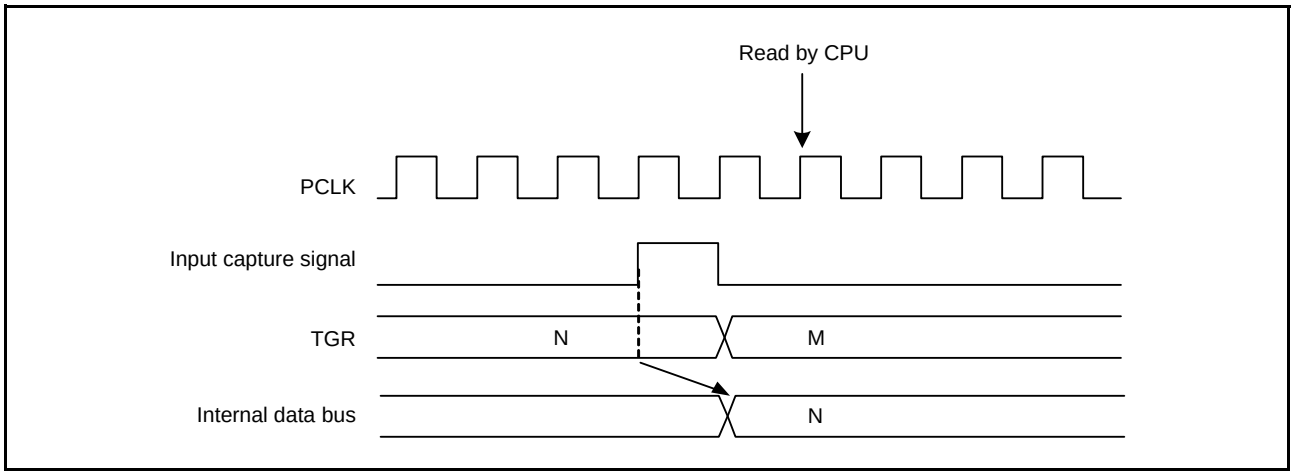


Figure 21.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

21.6.10 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 21.113 and Figure 21.114 show the timing in this case.

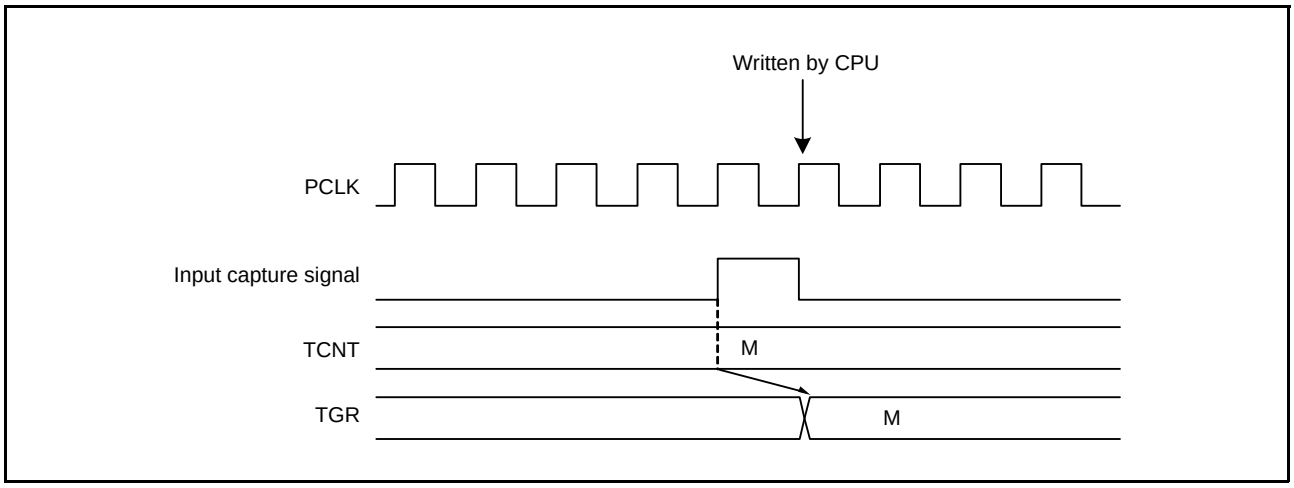


Figure 21.113 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

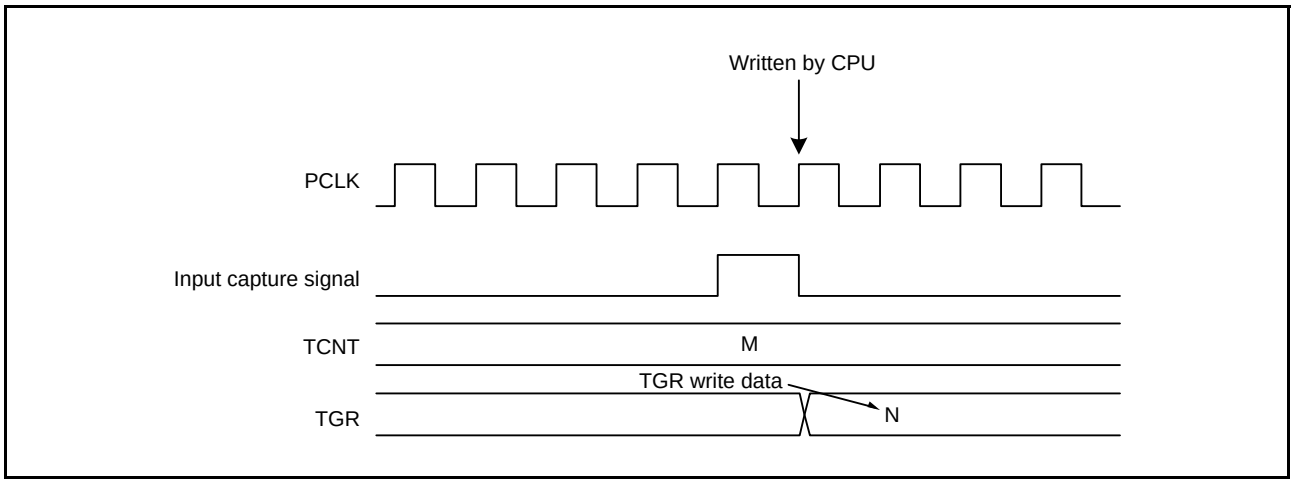


Figure 21.114 Contention between TGR Write Operation and Input Capture (MTU5)

21.6.11 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 21.115 shows the timing in this case.

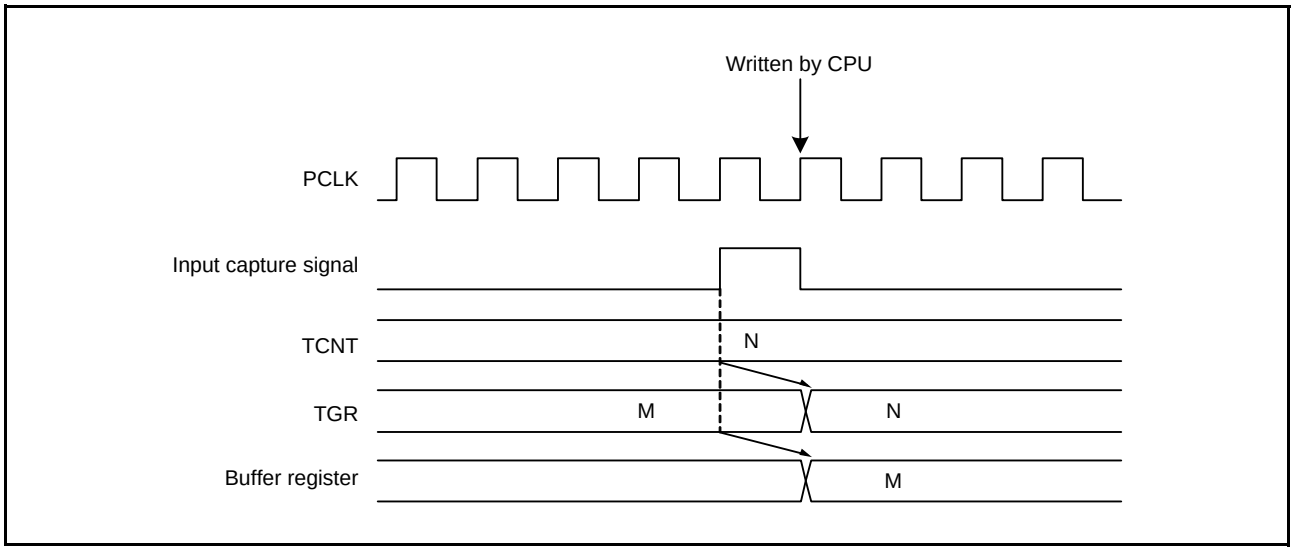


Figure 21.115 Contention between Buffer Register Write Operation and Input Capture

21.6.12 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 21.116 shows the timing in this case.
When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

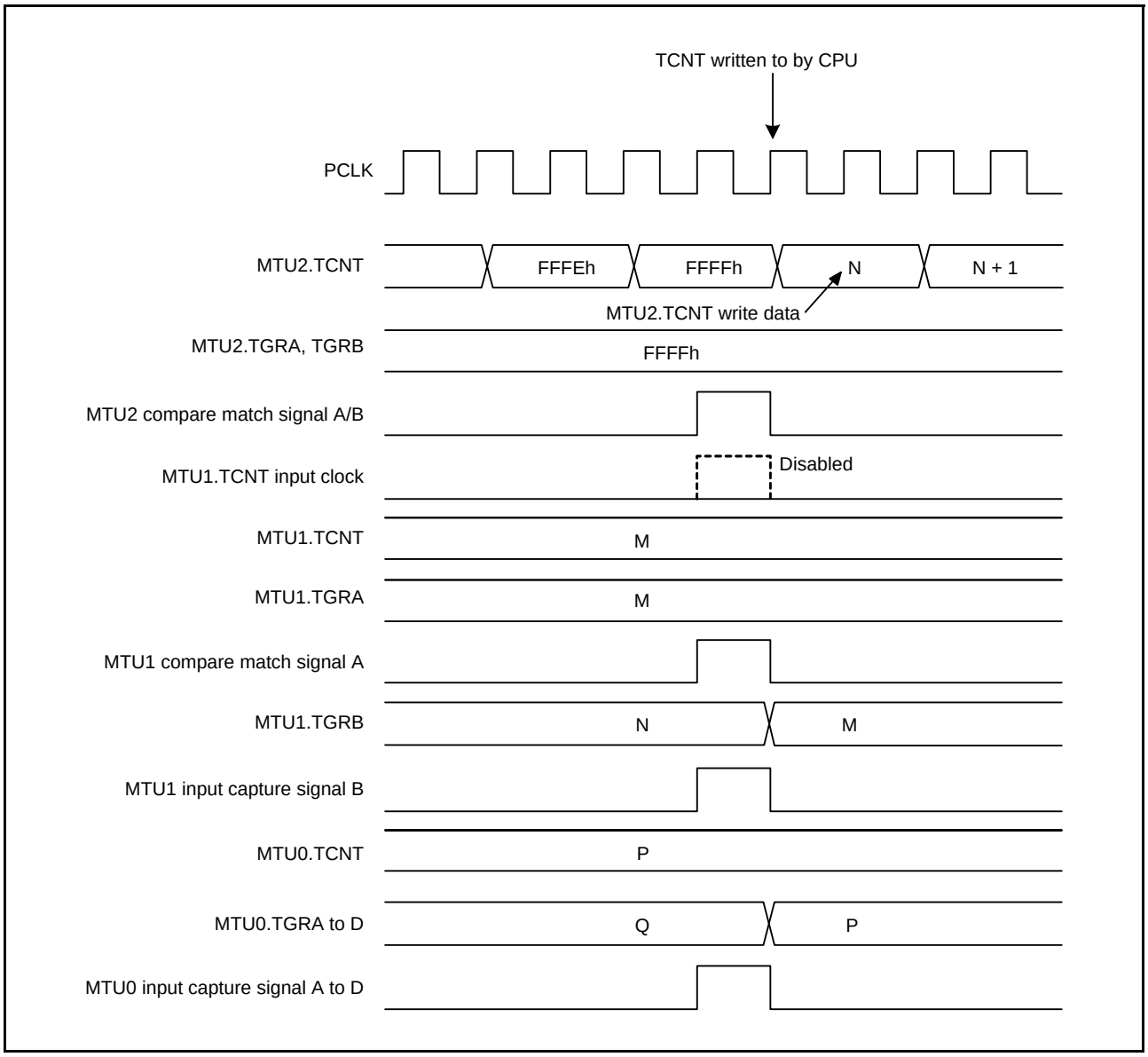


Figure 21.116 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

21.6.13 Counter Value when 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 21.117 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

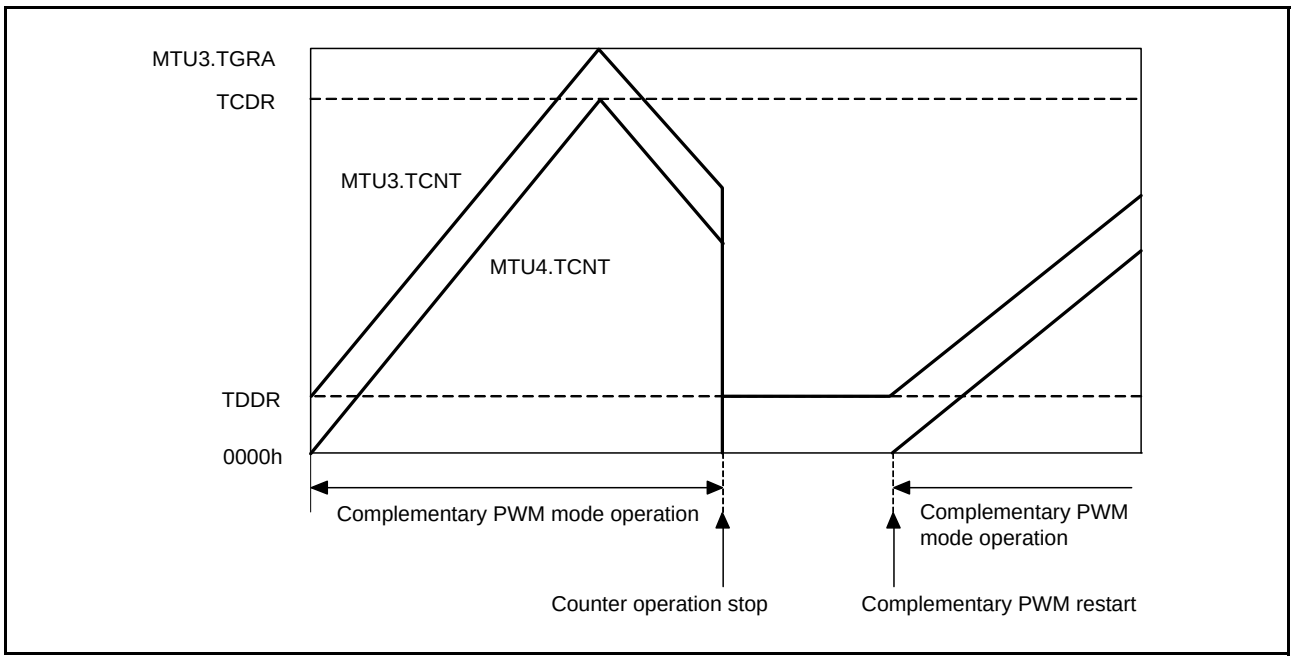


Figure 21.117 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

21.6.14 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 BFA and BFB bits in MTU4.TMDR should be set to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR 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 BFA bit in MTU3.TMDR 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.

21.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR 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 BFA and BFB bits of MTU3.TMDR. For example, if the BFA bit in MTU3.TMDR 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 21.118 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the BFA and BFB bits in MTU3.TMDR set to 1 and the BFA and BFB bits in MTU4.TMDR set to 0. (m= A to D)

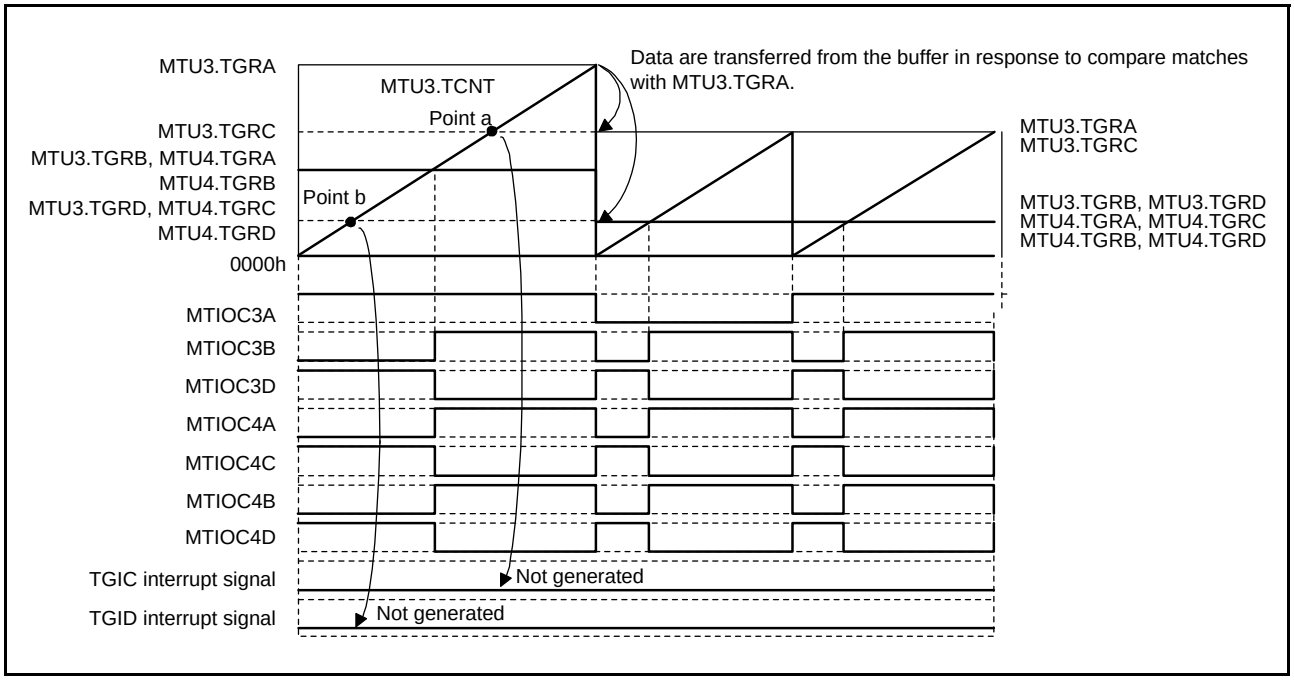


Figure 21.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

21.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the CST3 bit of TSTR 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 21.119 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 without synchronous operation setting.

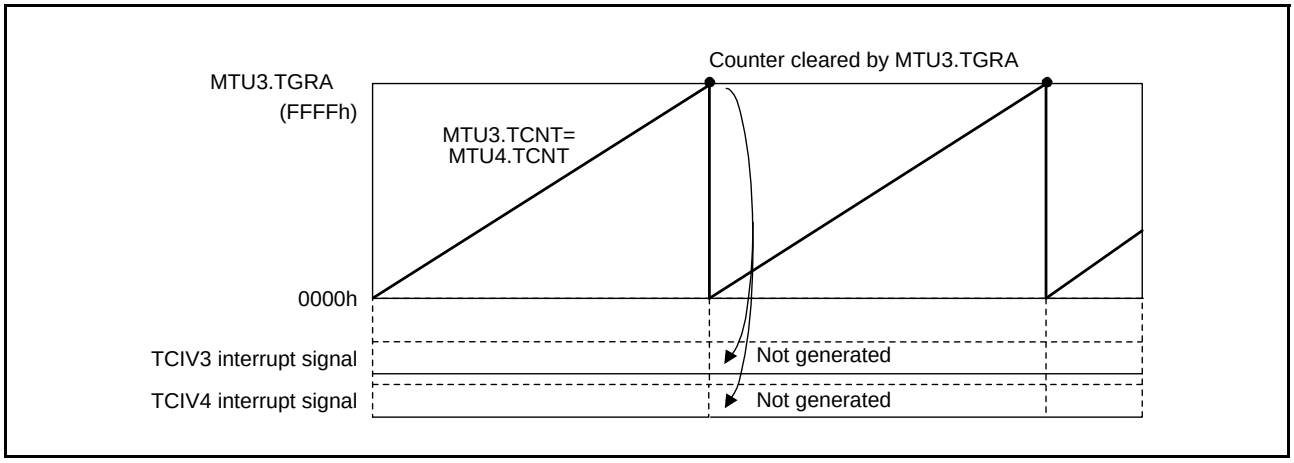


Figure 21.119 Overflow Flags in Reset-Synchronized PWM Mode

21.6.17 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.

Figure 21.120 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

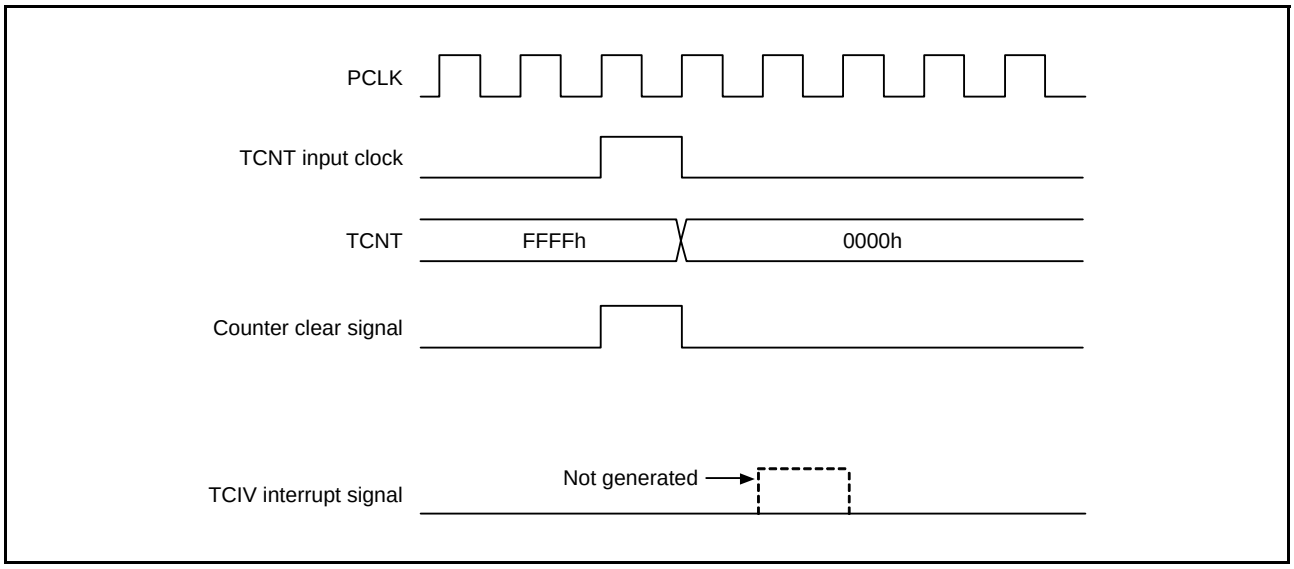


Figure 21.120 Contention between Overflow and Counter Clearing

21.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. The corresponding interrupt is not generated.
Figure 21.121 shows the operation timing when there is contention between TCNT write operation and overflow.

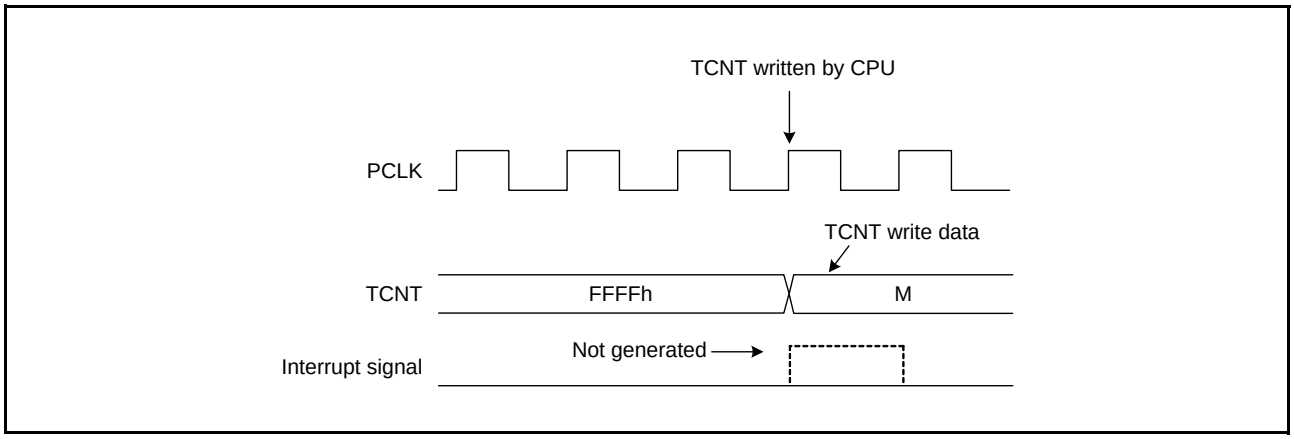


Figure 21.121 Contention between TCNT Write Operation and Overflow

21.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4, 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.

21.6.20 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 OLSP and OLSN bits in timer output control register 1 (TOCR1) to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h.

21.6.21 Interrupts During Periods in the Module-Stop State

When a 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 DMAC or DTC is not possible.
Accordingly, disable interrupts, etc. before making the settings for the module-stop state.

21.6.22 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 new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see section 21.2.8, Timer Input Capture Control Register (TICCR).

21.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used

The complementary PWM mode output protection functions are initially enabled. If the functions are not used, the POE.POECR2 register should be set to 00h.

21.6.24 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 (enabling interrupts), 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.

21.6.25 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 21.122, 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 21.123, 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).

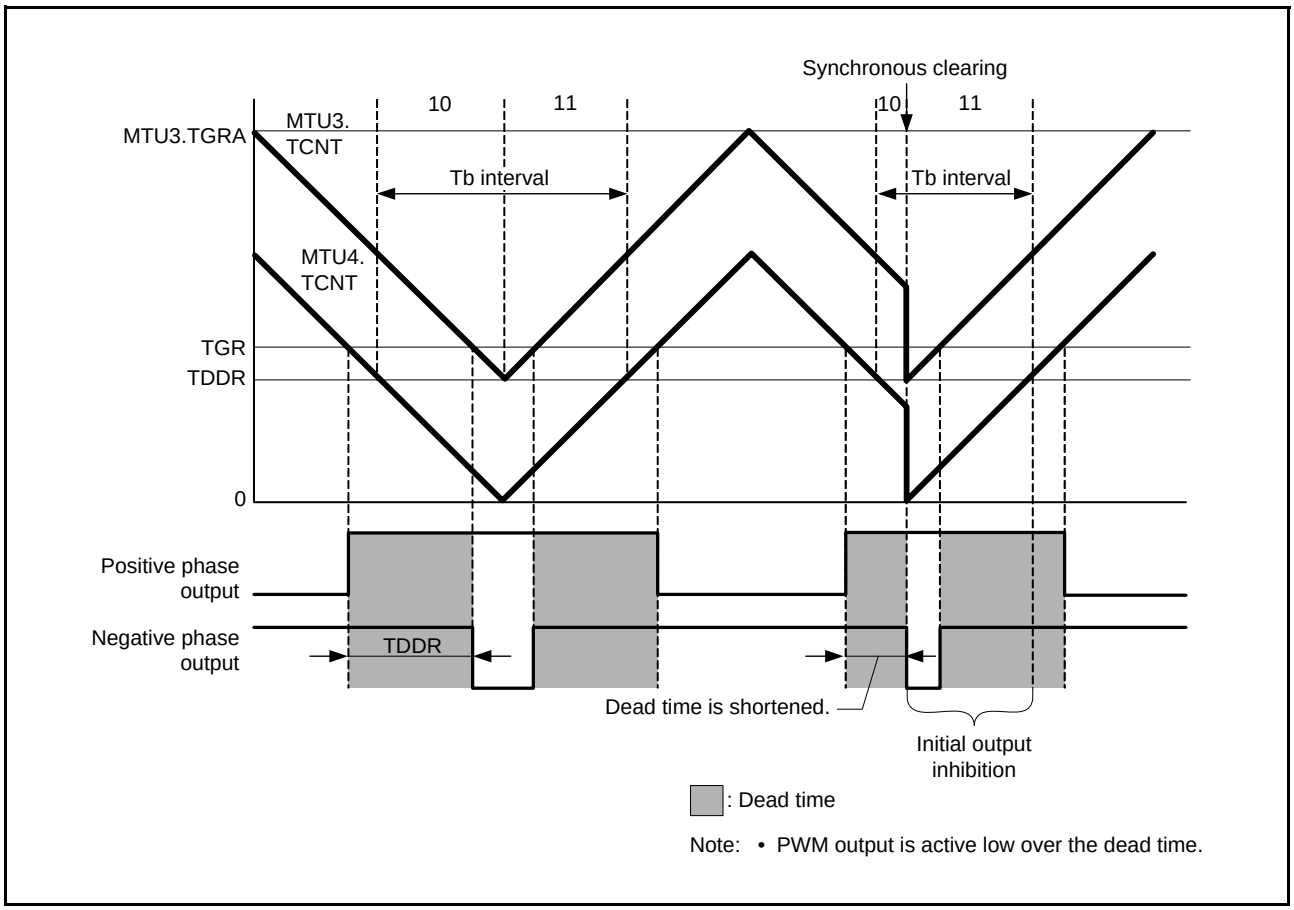


Figure 21.122 Example of Synchronous Clearing (when Condition 1 Applies)

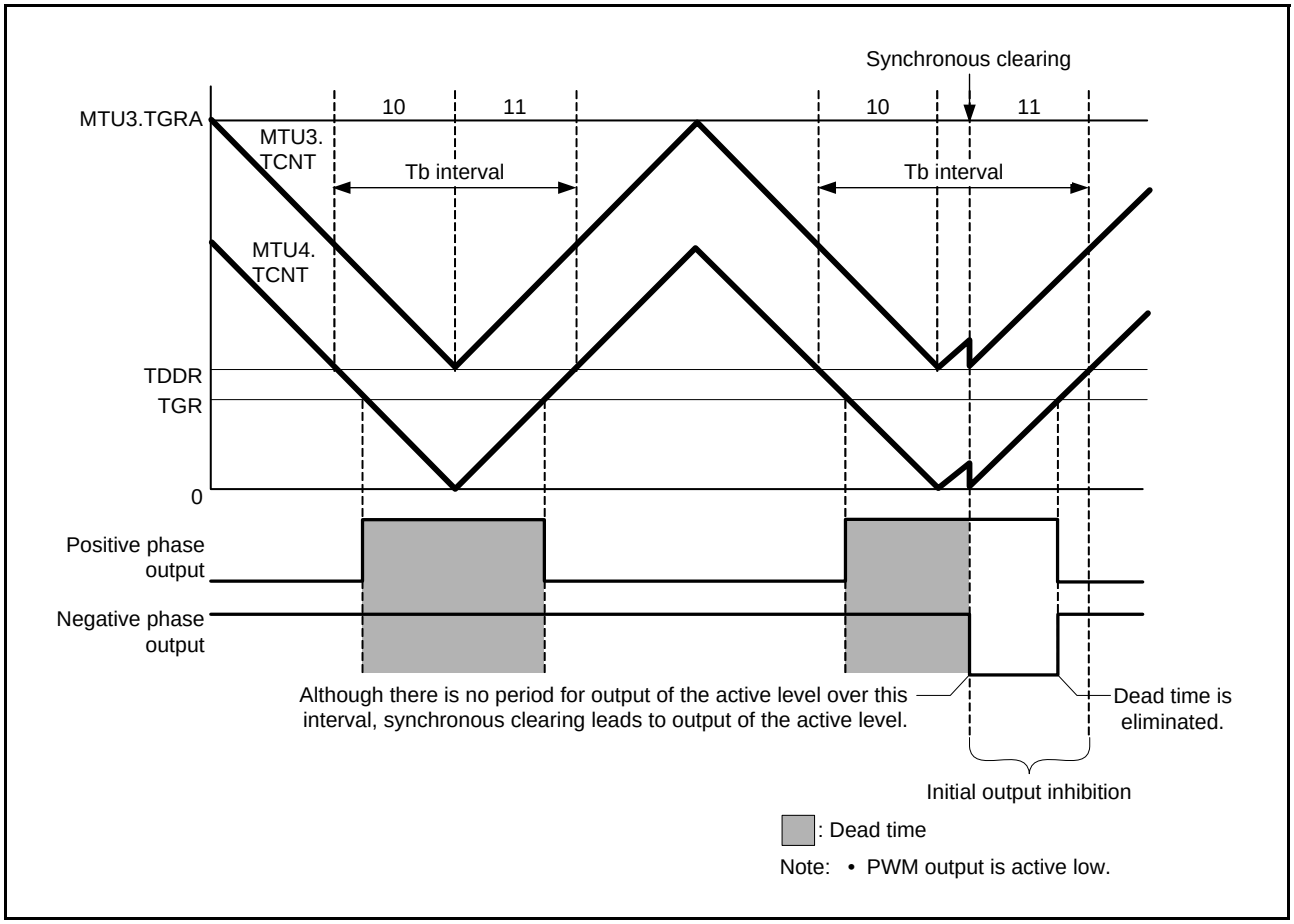


Figure 21.123 Example of Synchronous Clearing (when Condition 2 Applies)

21.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When TGR is set to 0000h, PCLK/1 is set as the counter clock, and compare match is set as the trigger for clearing of the counter 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 21.124 shows the timing for continuous output of the interrupt signal in response to a compare match.

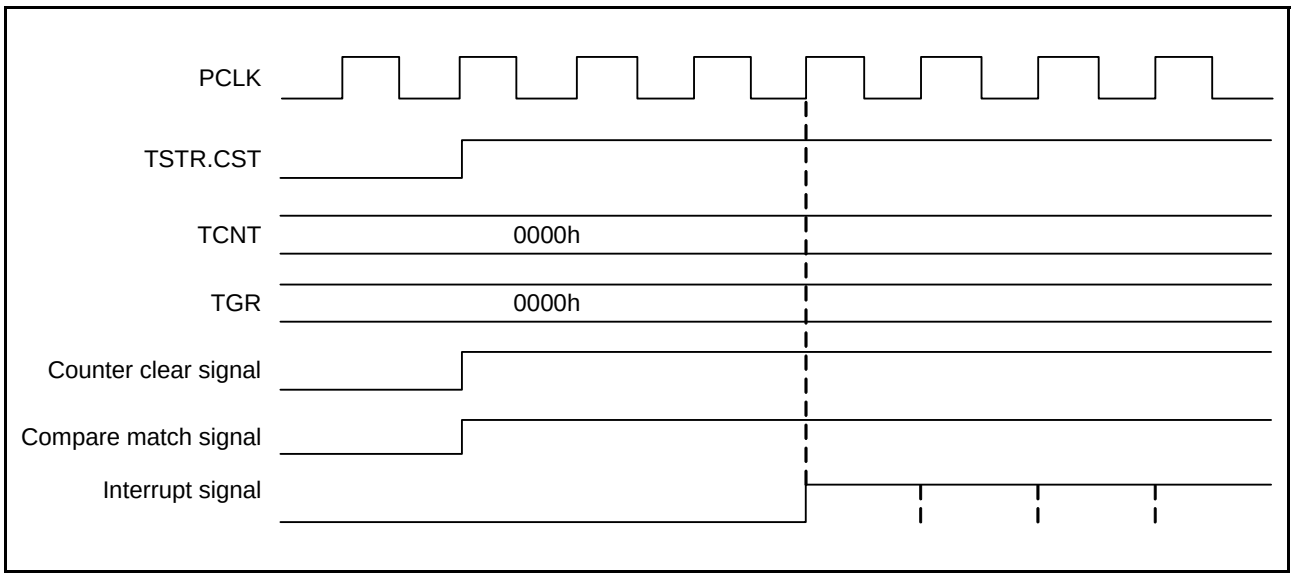


Figure 21.124 Continuous Output of Interrupt Signal in Response to a Compare Match

21.7 MTU Output Pin Initialization

21.7.1 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.

21.7.2 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. Cutting MTU output off is achieved by switching the pins to the output port function and placing the inverse of the active level on the pins. For motor driving pins, output can also be cut by hardware, using port output enable (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 21.59.

Table 21.59 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)

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

21.7.3 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, since a waveform is not output to the MTIOCnB (MTIOCnD) pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- 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, switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, and temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). 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 21.59. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 21.125 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

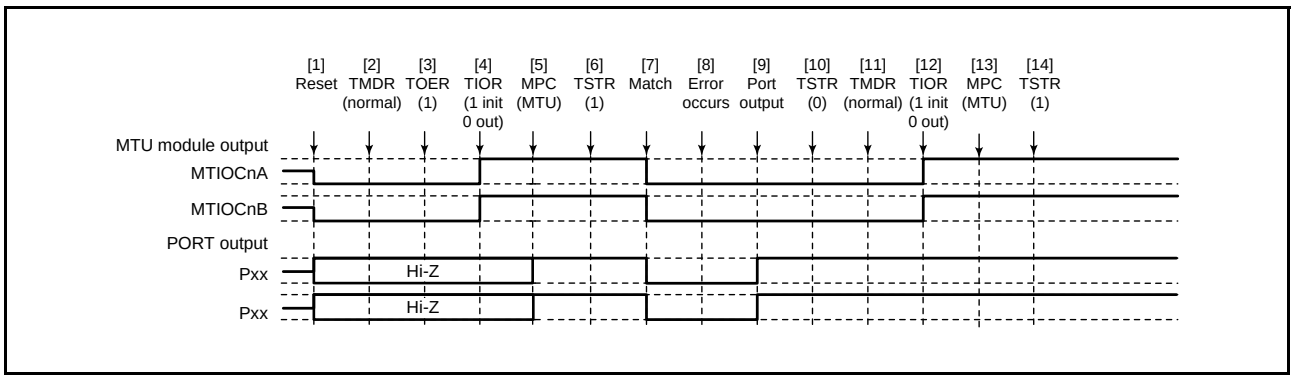


Figure 21.125 Error Occurrence in Normal Mode, Recovery in Normal Mode

[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 21.126 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

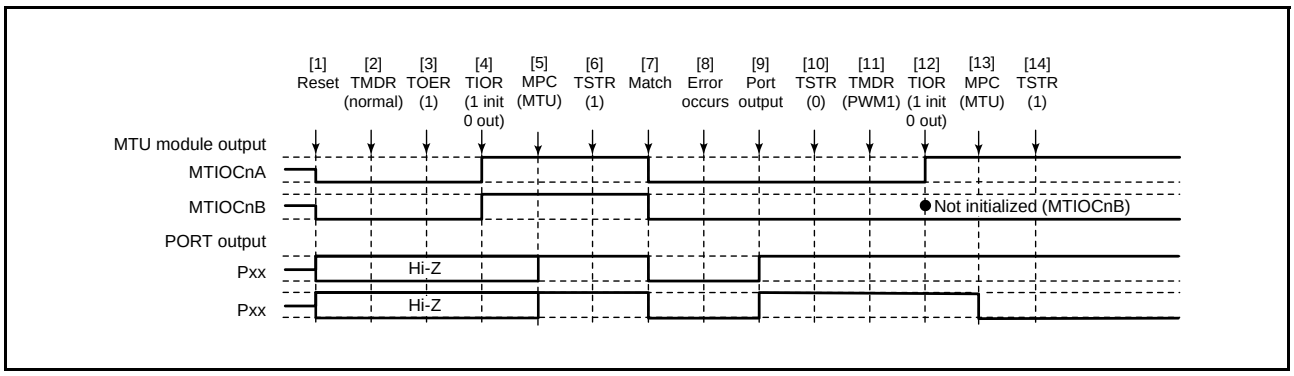


Figure 21.126 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 21.125.
[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 21.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

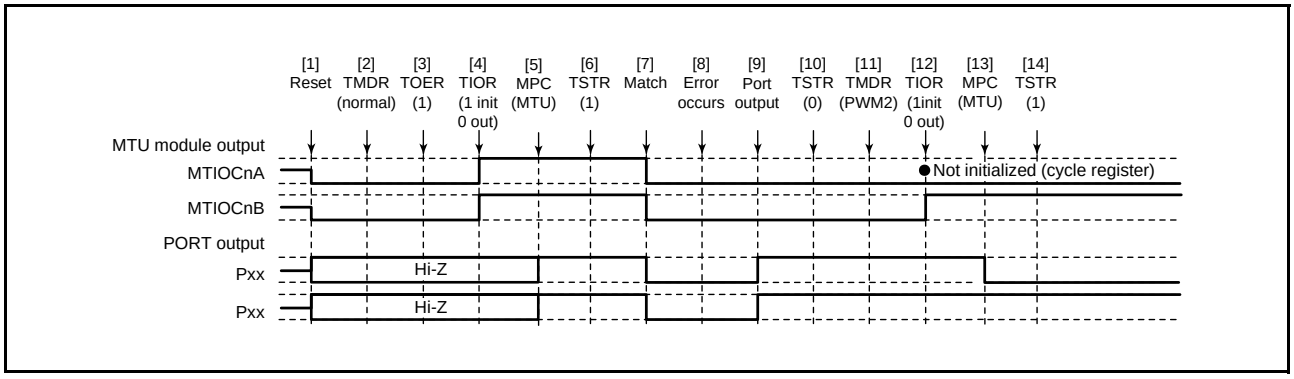


Figure 21.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 21.125.
[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 21.128 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

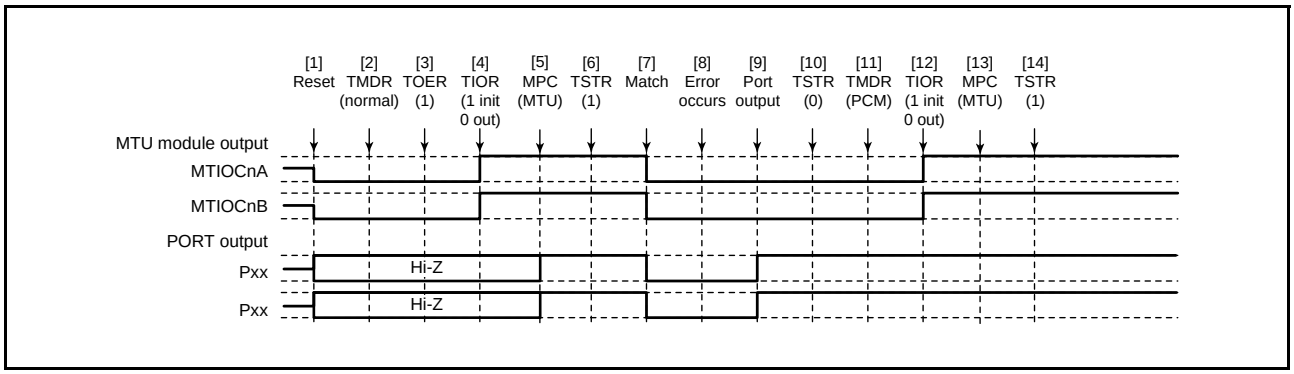


Figure 21.128 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 21.125.

[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 21.129 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

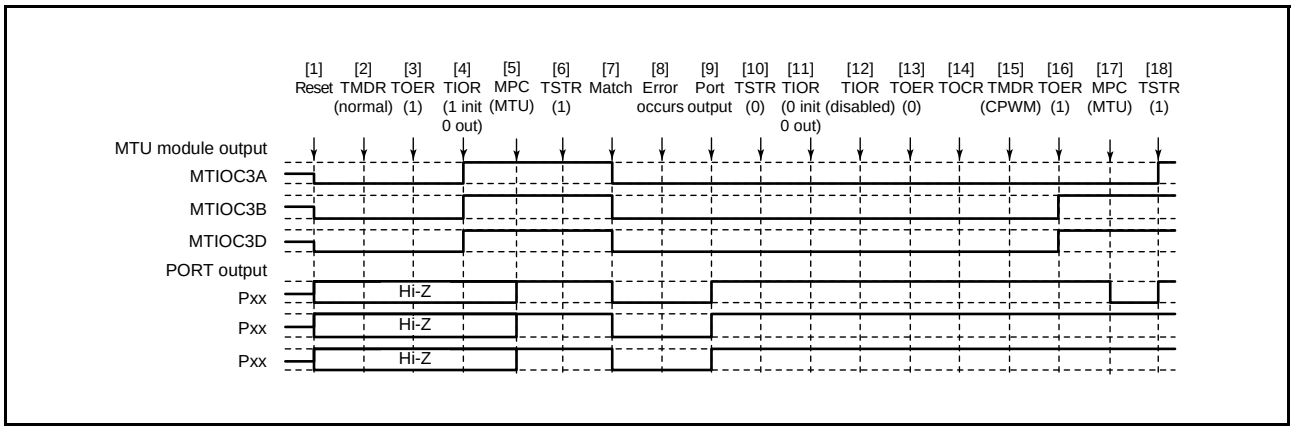


Figure 21.129 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 21.125.
[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 21.130 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

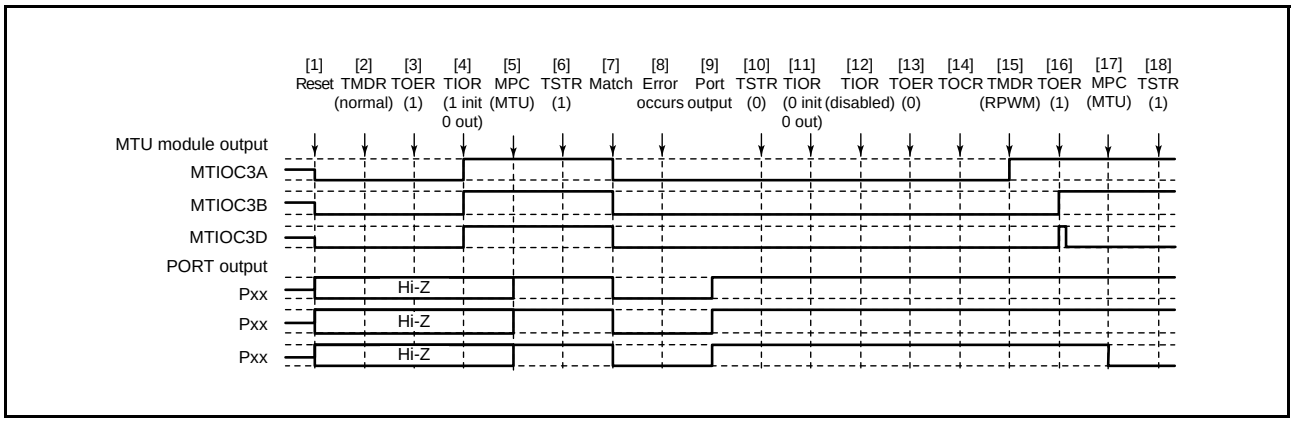


Figure 21.130 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [13] are the same as in Figure 21.125.
[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 21.131 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

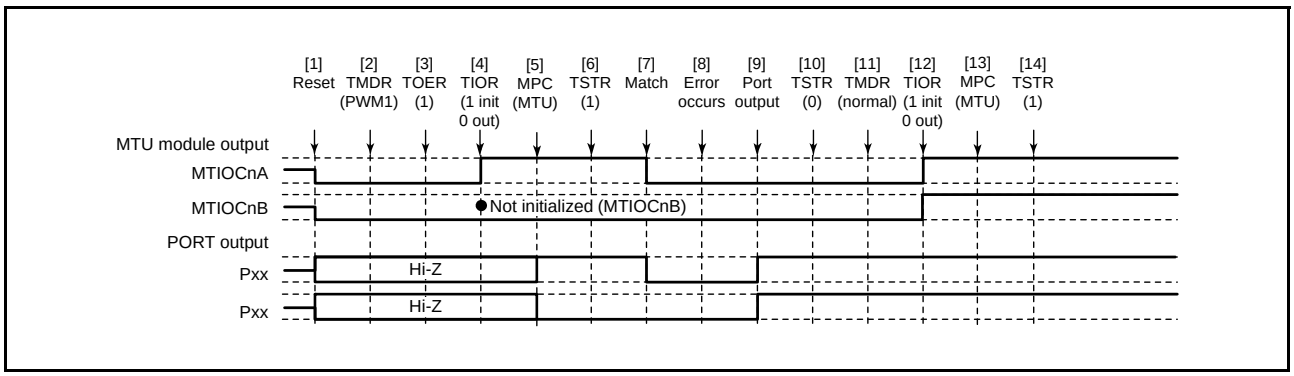


Figure 21.131 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

[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 21.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

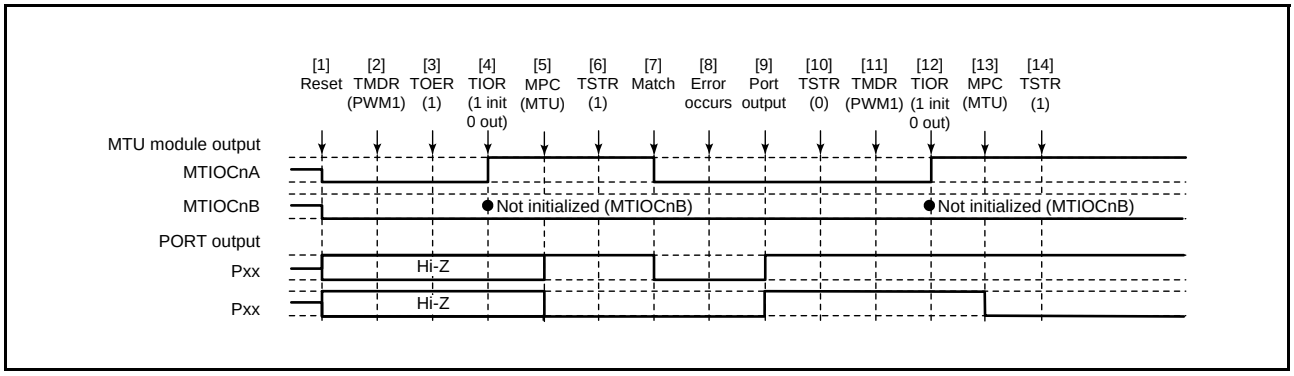


Figure 21.132 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 21.131.
[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 21.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

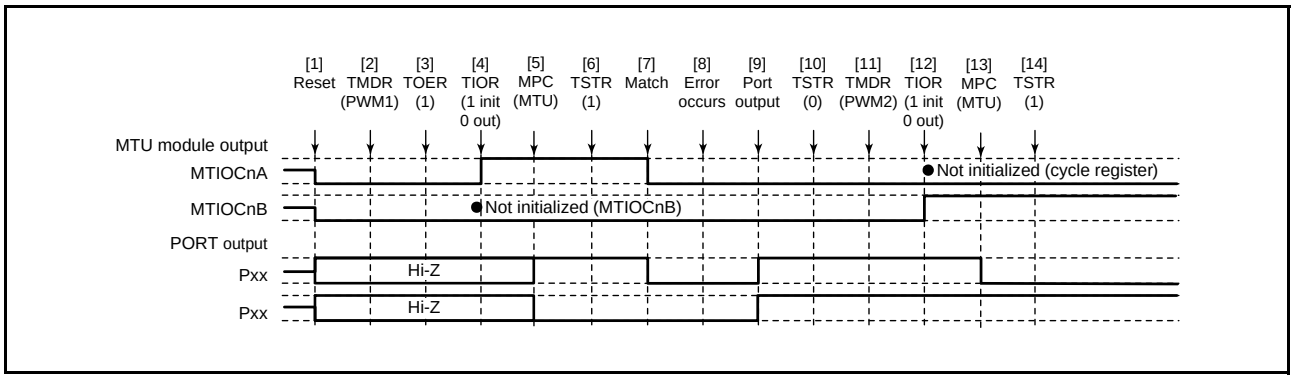


Figure 21.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 21.131.
[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 21.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

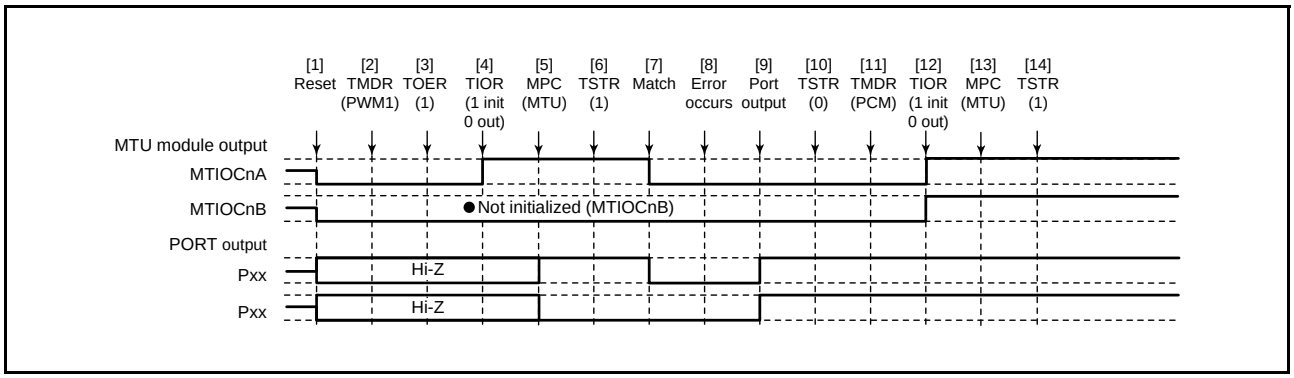


Figure 21.134 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 21.131.

[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 21.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

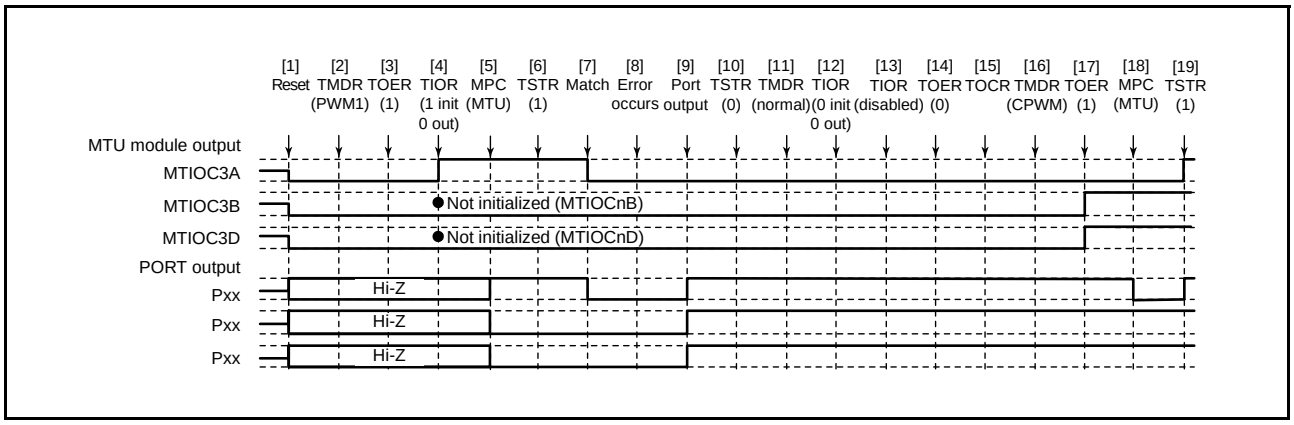


Figure 21.135 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 21.131.
[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 21.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

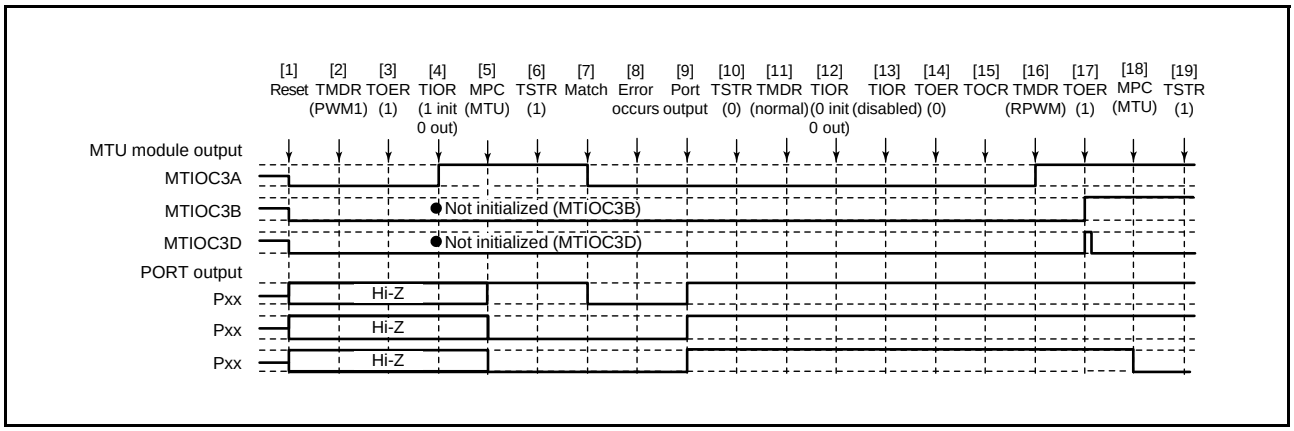


Figure 21.136 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

[1] to [14] are the same as in Figure 21.135.

[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 21.137 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

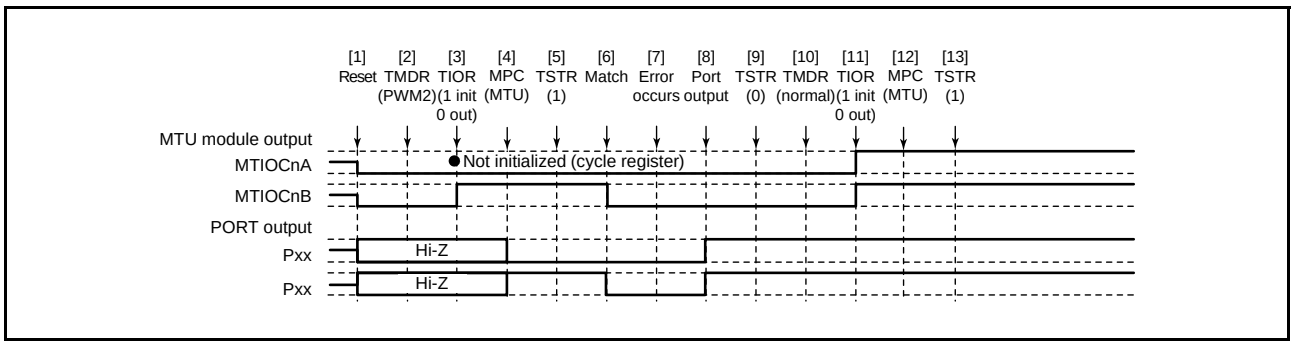


Figure 21.137 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

[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 21.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

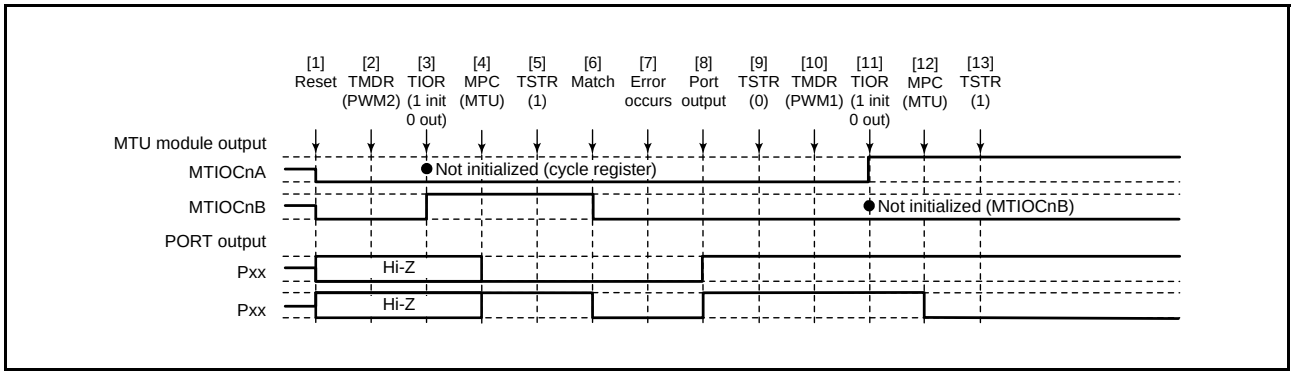


Figure 21.138 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 21.137.
[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 21.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

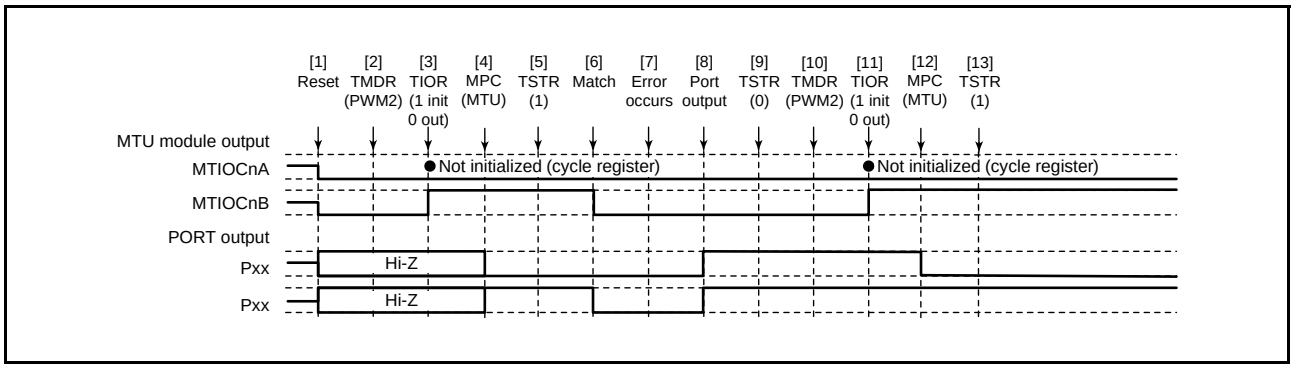


Figure 21.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 21.137.
[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 21.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

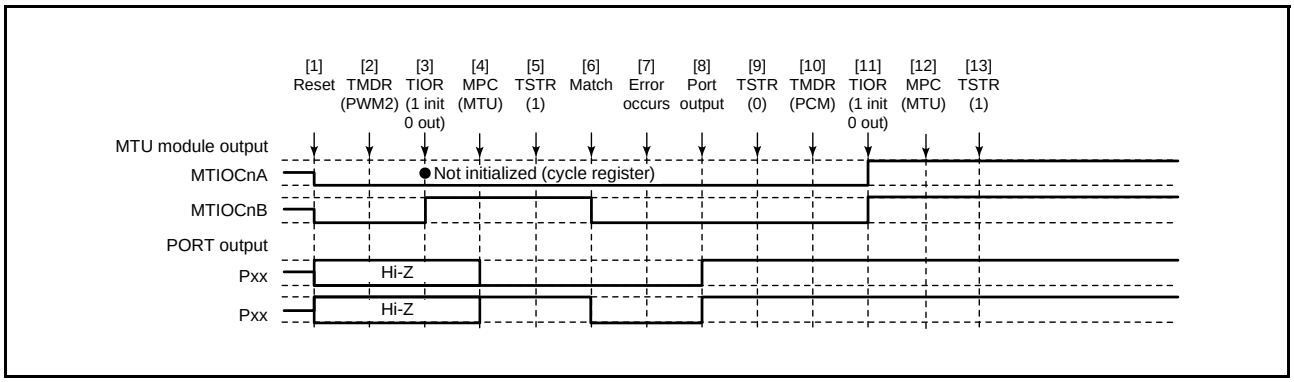


Figure 21.140 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 21.137.
[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 21.141 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

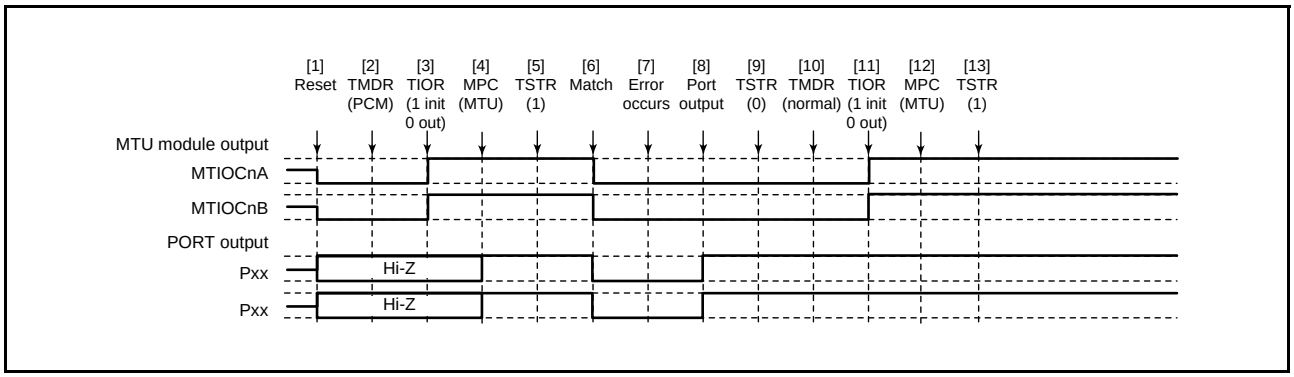


Figure 21.141 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

[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 21.142 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

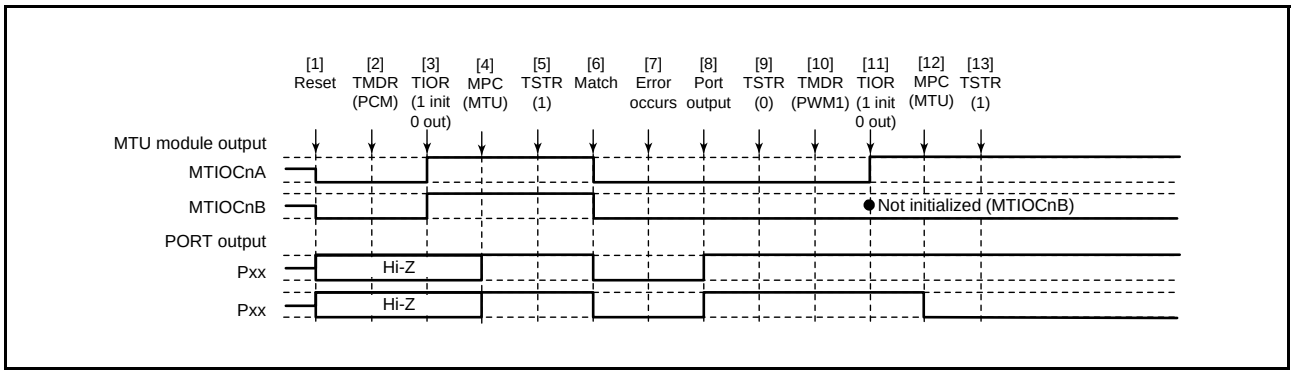


Figure 21.142 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 21.141.

[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 21.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

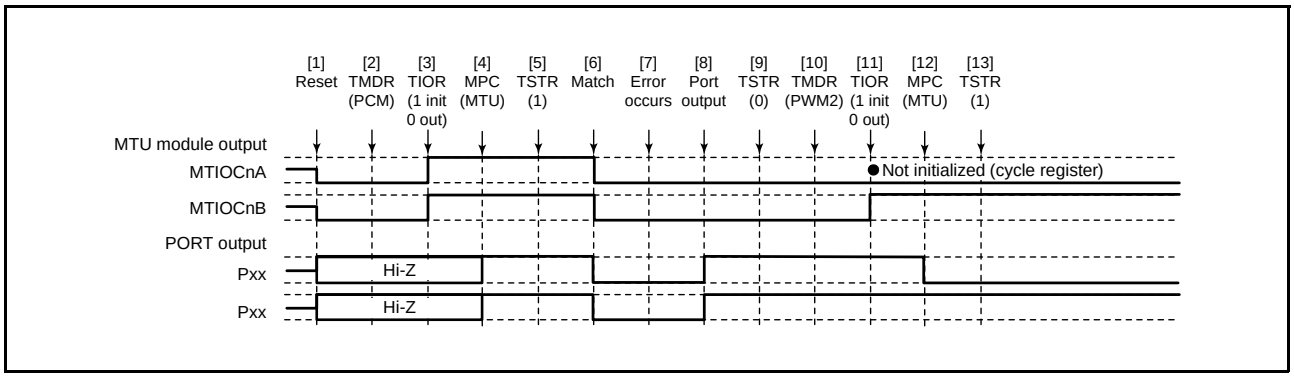


Figure 21.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 21.141.

[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 21.144 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

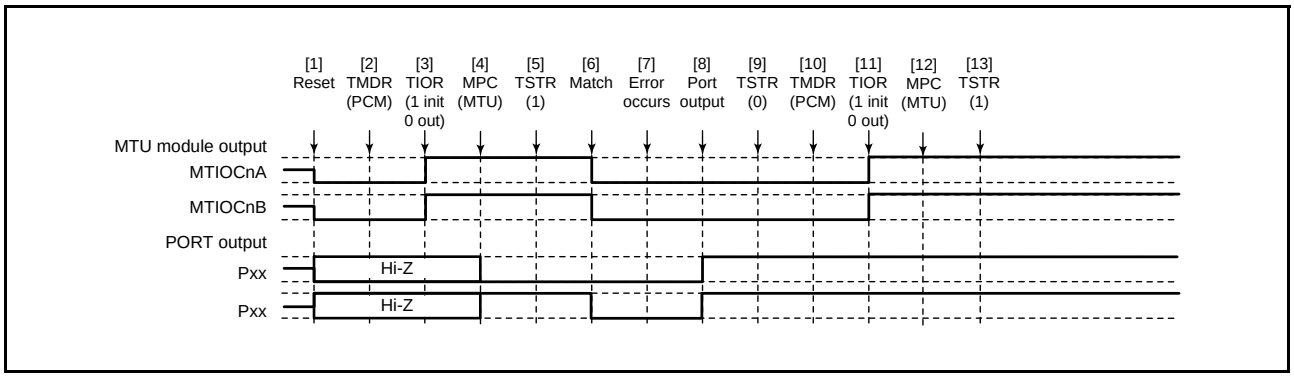


Figure 21.144 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 21.141.
[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 21.145 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

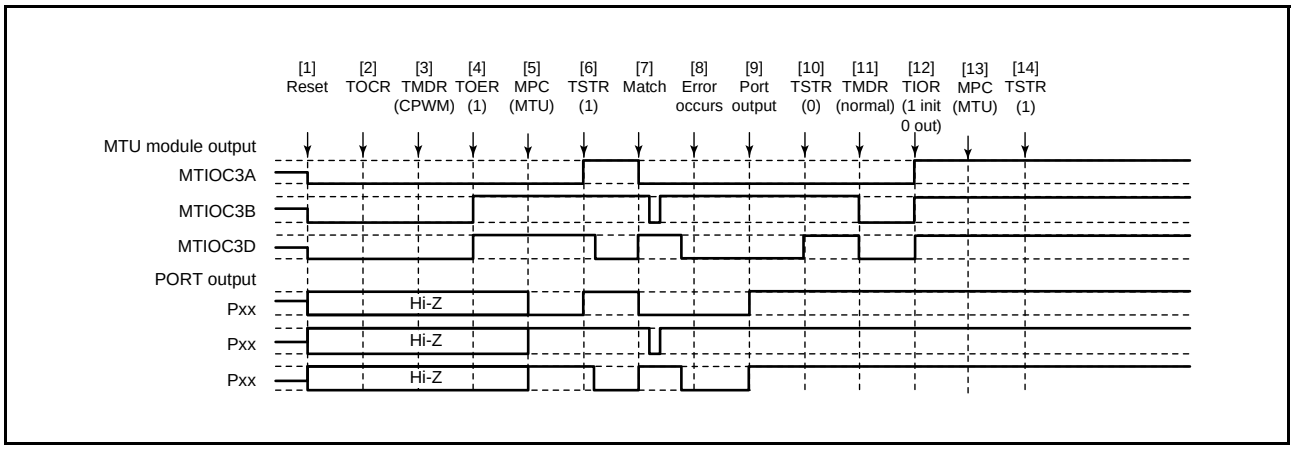


Figure 21.145 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

[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 21.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

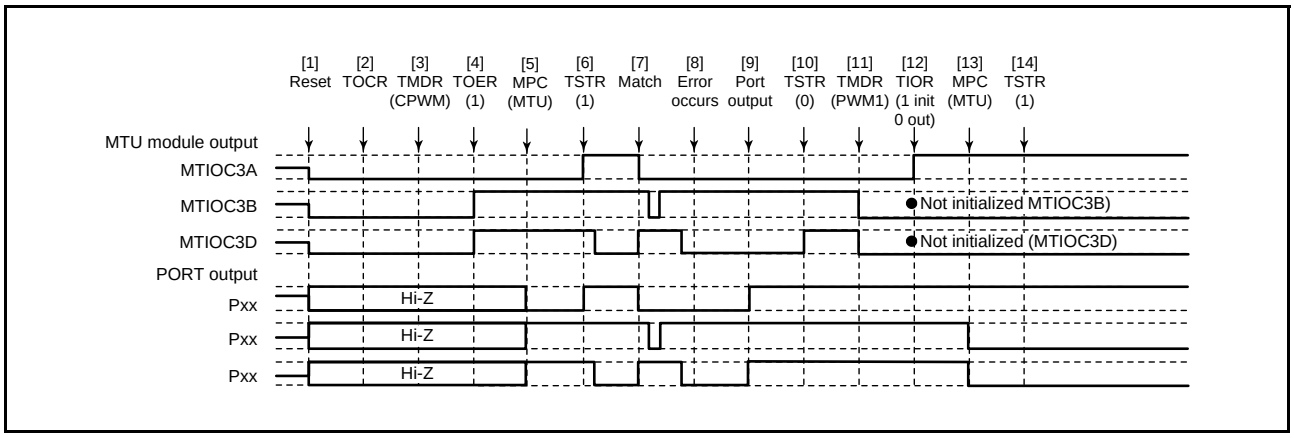


Figure 21.146 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 21.145.

[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 21.147 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).

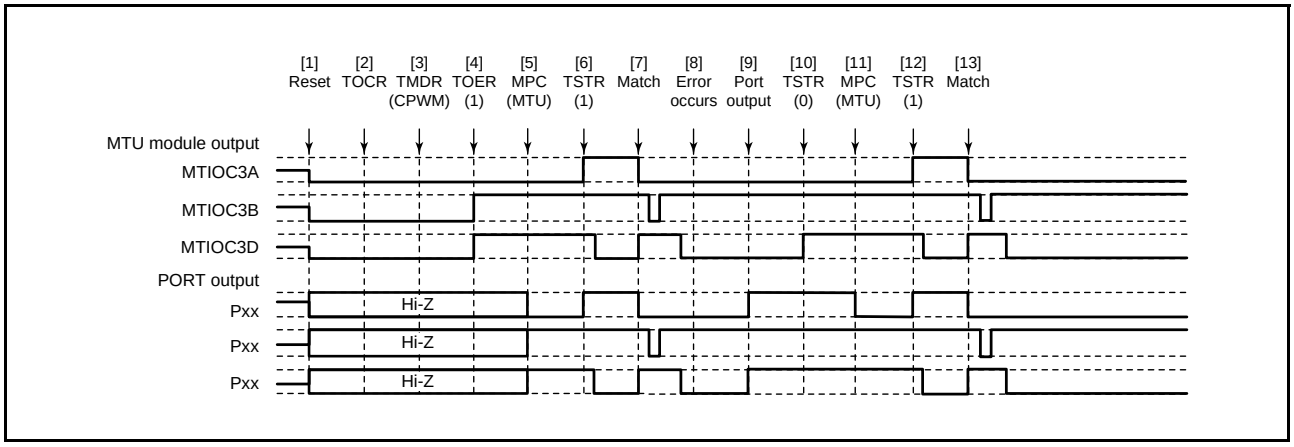


Figure 21.147 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 21.145.
[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 21.148 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).

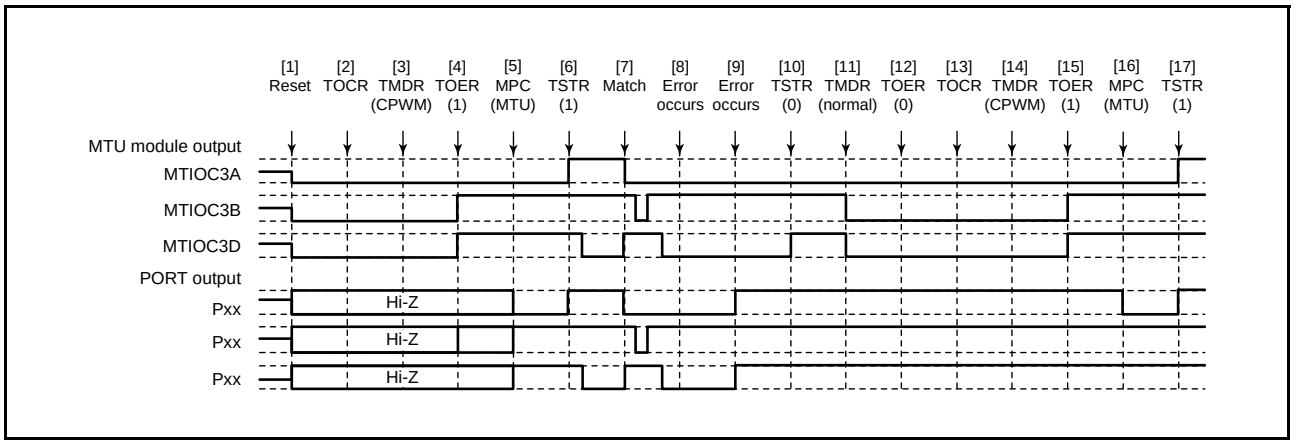


Figure 21.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 21.145.
[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 21.149 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

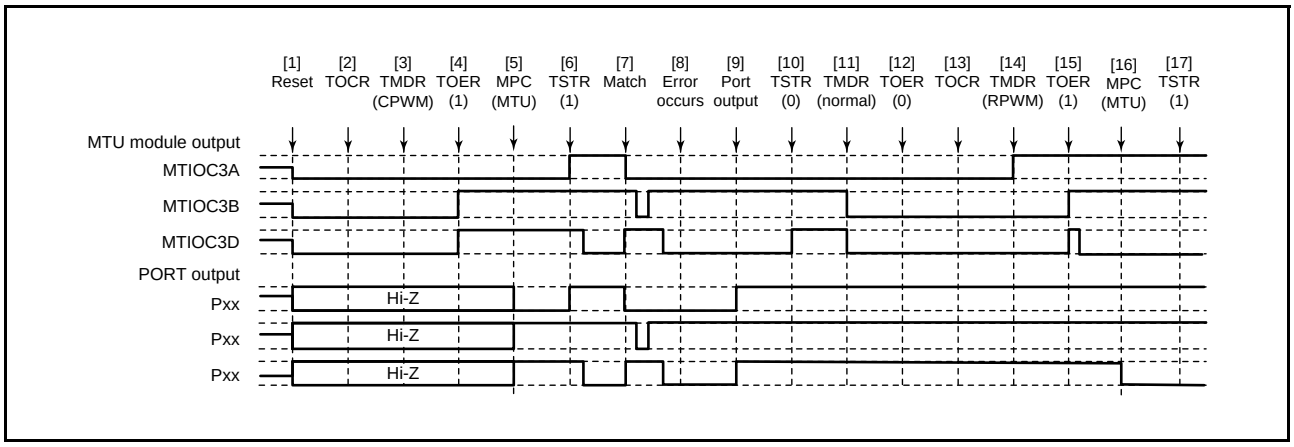


Figure 21.149 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [10] are the same as in Figure 21.145.
[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 21.150 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

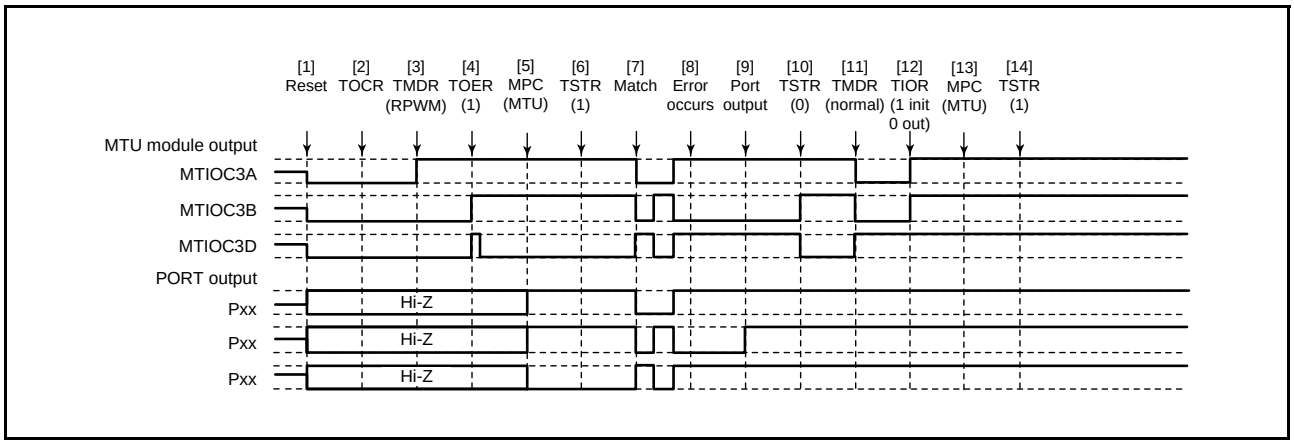


Figure 21.150 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

[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 21.151 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

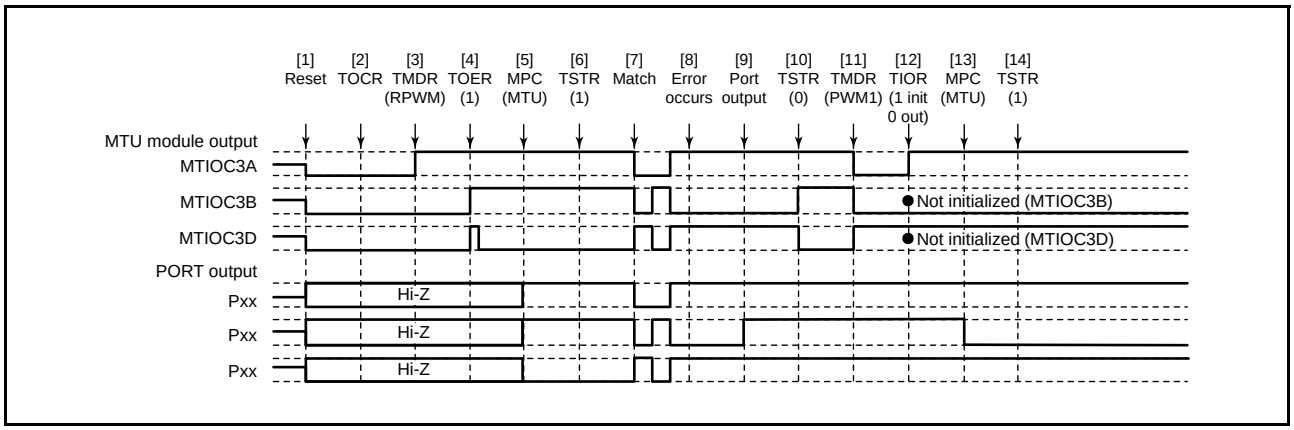


Figure 21.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 21.150.

[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 21.152 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

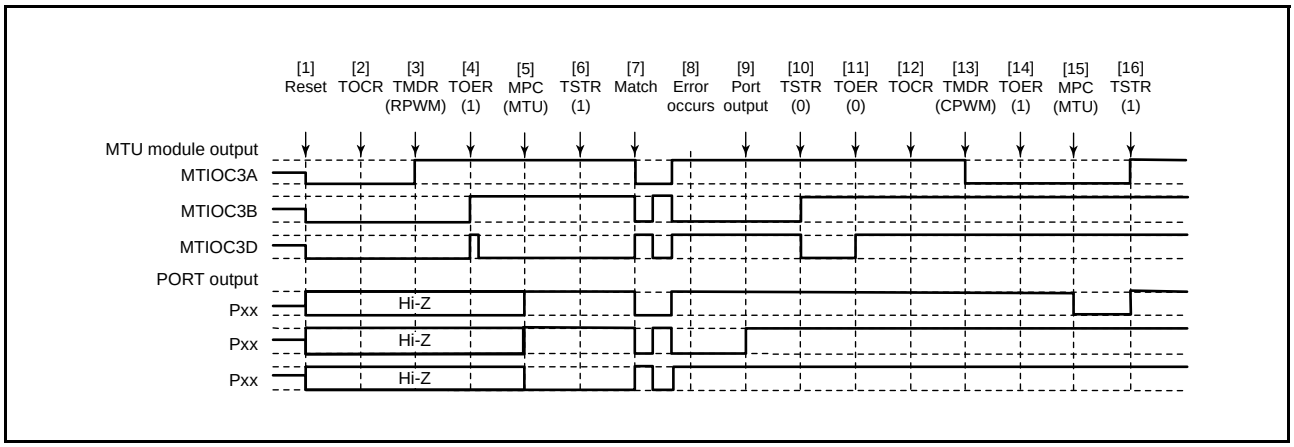


Figure 21.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 21.150.
[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 21.153 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.

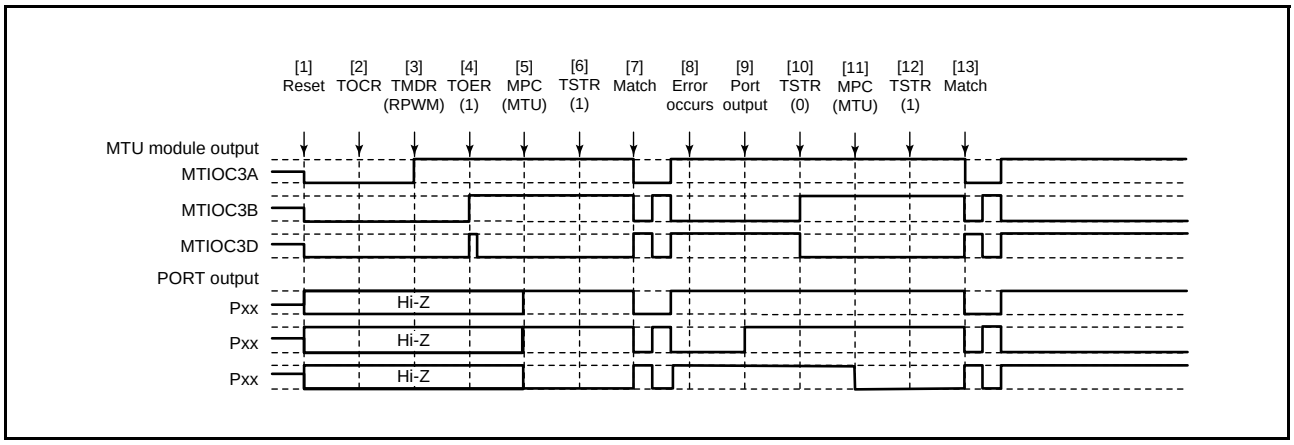


Figure 21.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

[1] to [10] are the same as in Figure 21.150.

[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.

21.8 Link Operation by ELC

21.8.1 Event Signal Output to ELC

MTU is capable of link operation for the preliminarily-setting module when interrupt request signal is used as an event signal by the event link controller (ELC).
The event signal is able to output regardless of the settings of the appropriate interrupt request enable bits.

21.8.2 MTU Operation by Event Signal Reception from ELC

MTU can be operated in the following by the preliminarily-setting event.

(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 21.60 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 21.60 lists the TSTR register bits used for each channel.
For details on the count start operation setting, see section 21.3.1, (1) Counter Operation.

Table 21.60 Linkage Operating TSTR Register by ELC

Channel No.	TSTR register
Channel 1	TSTR.CST1 bit
Channel 2	TSTR.CST2 bit
Channel 3	TSTR.CST3 bit
Channel 4	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 21.61 lists the timer general register and timer I/O control register used in the input capture operation by ELC.
For details on the input capture setting, see section 21.3.1, (3) Input Capture Function.

Table 21.61 Timer General Register and Timer I/O Control Register used in the Input Capture Operation by ELC

Channel No.	Register Name	Bit Name of TIOR Register
Channel 1	TGRA register	TIOR.IOA[3:0] bits
Channel 2	TGRA register	TIOR.IOA[3:0] bits
Channel 3	TGRA register	TIORH.IOA[3:0] bits
Channel 4	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 the initial value. When the CSTn bit in the TSTR register is 1, the count operation can be continued. For details on the TSTR.CSTn bit, see Table 21.60.

21.8.3 Usage Notes on MTU by Event Signal Reception from 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.

22. 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).
It can also generate simultaneous interrupt requests.

22.1 Overview

Table 22.1 lists the specifications of the POE, and Figure 22.1 shows a block diagram of the POE.

Table 22.1 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 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 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.
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-stopped detection signal from the clock generation circuit, and a high-impedance request/interrupt request generation circuit as shown in Figure 22.1.

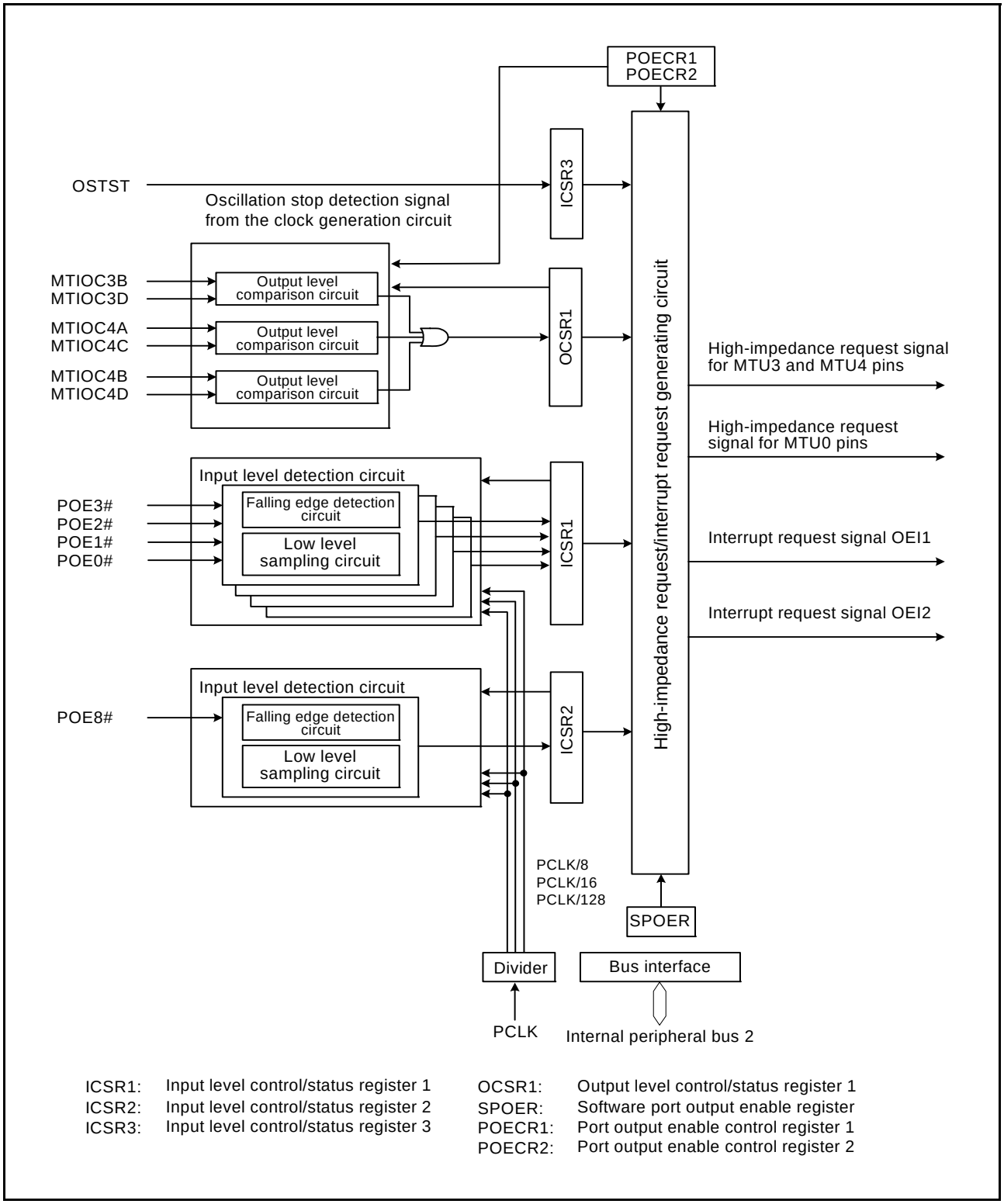


Figure 22.1 POE Block Diagram

Table 22.2 lists input/output pins to be used by the POE.

Table 22.2 POE Input/Output 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 signals to place the MTU0 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 22.3 lists output-level comparisons with pin combinations.

Table 22.3 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 cycles of PCLK. (When the MTU. TOCR1.TOCS bit = 0: Low level if the MTU.TOCR1.OLSP and OLSN bits are 0, and high level if the MTU.TOCR1.OLSP and OLSN bits are 1. When the MTU. TOCR1.TOCS bit = 1: Low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 0, and high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 1.)
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

22.2 Register Descriptions

22.2.1 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 request on the falling edge of POE0# input. 0 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a 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 request on the falling edge of POE1# input. 0 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a 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 request on the falling edge of POE2# input. 0 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a 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 request on the falling edge of POE3# input. 0 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a 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, see section 22.3.5, 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.
[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

POE1F Flag (POE1 Flag)

This flag indicates that a high impedance request has been input to the POE1# pin.
[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

POE2F Flag (POE2 Flag)

This flag indicates that a high impedance request has been input to the POE2# pin.
[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

POE3F Flag (POE3 Flag)

This flag indicates that a high impedance request has been input to the POE3# pin.
[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

22.2.2 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 22.3 has simultaneously become an active level. If the PnCZEA (n = 1, 2, 3) bits in POECR2 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.

[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, see section 22.3.5, Release from the High-Impedance.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

22.2.3 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 request on the falling edge of POE8# input 0 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/128 clock pulses 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.
[Clearing condition]
Writing 0 to POE8F after reading POE8F = 1
When writing 0 to the flag while low-level sampling is selected for the POE8M[1:0] bits, the POE8# pin input must be at the high level.
For details, see section 22.3.5, Release from the High-Impedance.
[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

22.2.4 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 specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

[Setting condition]

- By writing 1 to CH34HIZ

CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

[Setting condition]

- By writing 1 to CH0HIZ

22.2.5 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.

22.2.6 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.

22.2.7 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 cycles have elapsed after stopped oscillation was detected.

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

[Setting condition]

- Detection of the oscillation-stopped state

22.3 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 ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.

(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 ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.

(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 ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.

(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 ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE3ZE and ICSR3.OSTSTE set to 1.

(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 ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.

- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P1CZEA and ICSR3.OSTSTE set to 1.

(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 ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P2CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P2CZEA and ICSR3.OSTSTE set to 1.

(7) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

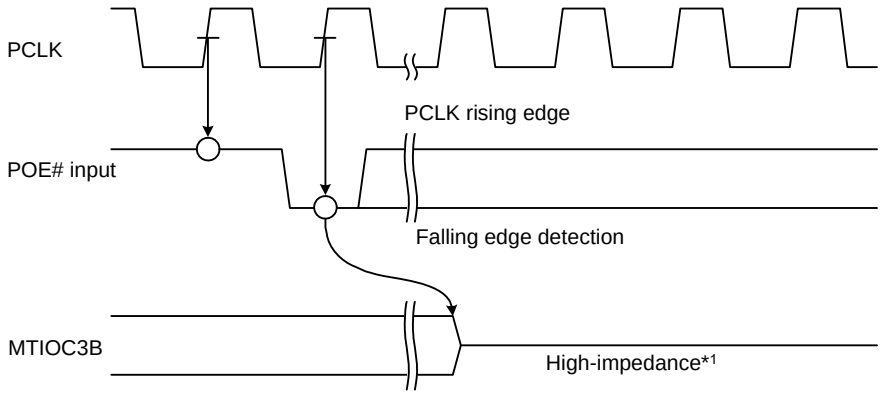
- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P3CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P3CZEA and ICSR3.OSTSTE set to 1.

22.3.1 Input Level Detection Operation

If the input conditions set by ICSR1 and ICSR2 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 full cycle of PCLK, whether the falling edge will or will not be detected cannot be guaranteed.
Figure 22.2 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.

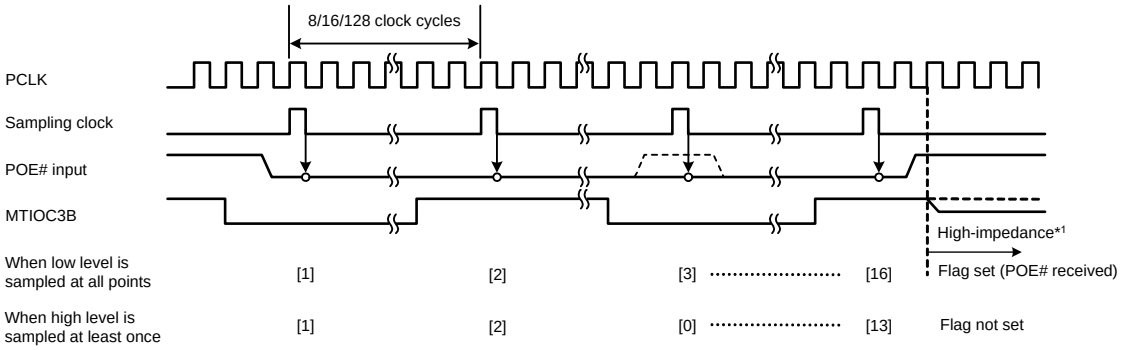


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 22.2 Falling Edge Detection

(2) Low-Level Detection

Figure 22.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 and ICSR2. If even one high level is detected during this interval, the low level is not accepted. 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.

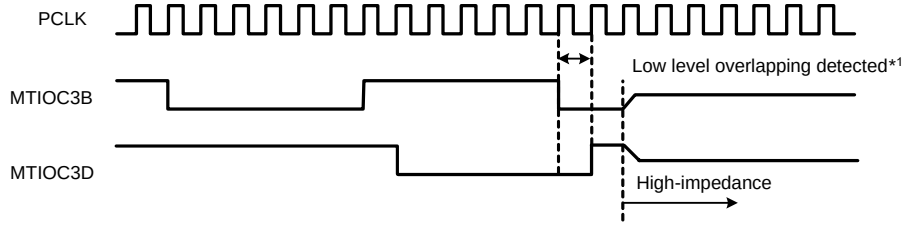


Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

Figure 22.3 Low-Level Detection Operation

22.3.2 Output-Level Compare Operation

Figure 22.4 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 22.4 Output-Level Compare Operation

22.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled using the software port output enable register (SPOER).
Setting the CH34HIZ bit in SPOER to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the port output enable control register 2 (POECR2) in the high-impedance.
Setting the CH0HIZ bit in SPOER to 1 places the MTU0 output pins specified by the port output enable control register 1 (POECR1) in the high-impedance.

22.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation-stop detection circuit in the clock pulse generator detects stopped oscillation, pins for complementary PWM output by POECR2 and the MTU0 output pins specified by POECR1 are placed in the high-impedance.

22.3.5 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 POE3F to POE0F flags in ICSR1 and the POE8F flag in ICSR2. Note, however, that when low-level sampling is selected by the POE3M[1:0] to POE0M[1:0] bits in ICSR1 and the POE8M[1:0] bits in ICSR2, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE#8 but has not yet been sampled, 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 OSF1 flag in OCSR1. 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 OSTSTF or OSTSTE bit in ICSR3 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 CH0HIZ) releases the pins from the high-impedance.

22.4 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 22.4 shows 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 22.4 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.

22.5 Usage Notes

22.5.1 Transitions to Software Standby Mode

When the POE is used, do not make a transition to software standby mode. In these modes, the POE stops and thus the high-impedance of pins cannot be controlled.

22.5.2 When POE is not Used

When the POE is not used, write 00h to the port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

22.5.3 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 port mode register (PMR). When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

23. 8-Bit Timer (TMR)

The RX220 Group 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 can generate a baud rate clock signal for the SCI and have the same functions.

23.1 Overview

Table 23.1 lists the specifications of the TMR.
Figure 23.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 23.2 shows that of the 8-bit timer module (unit 1).

Table 23.1 Specifications of TMR

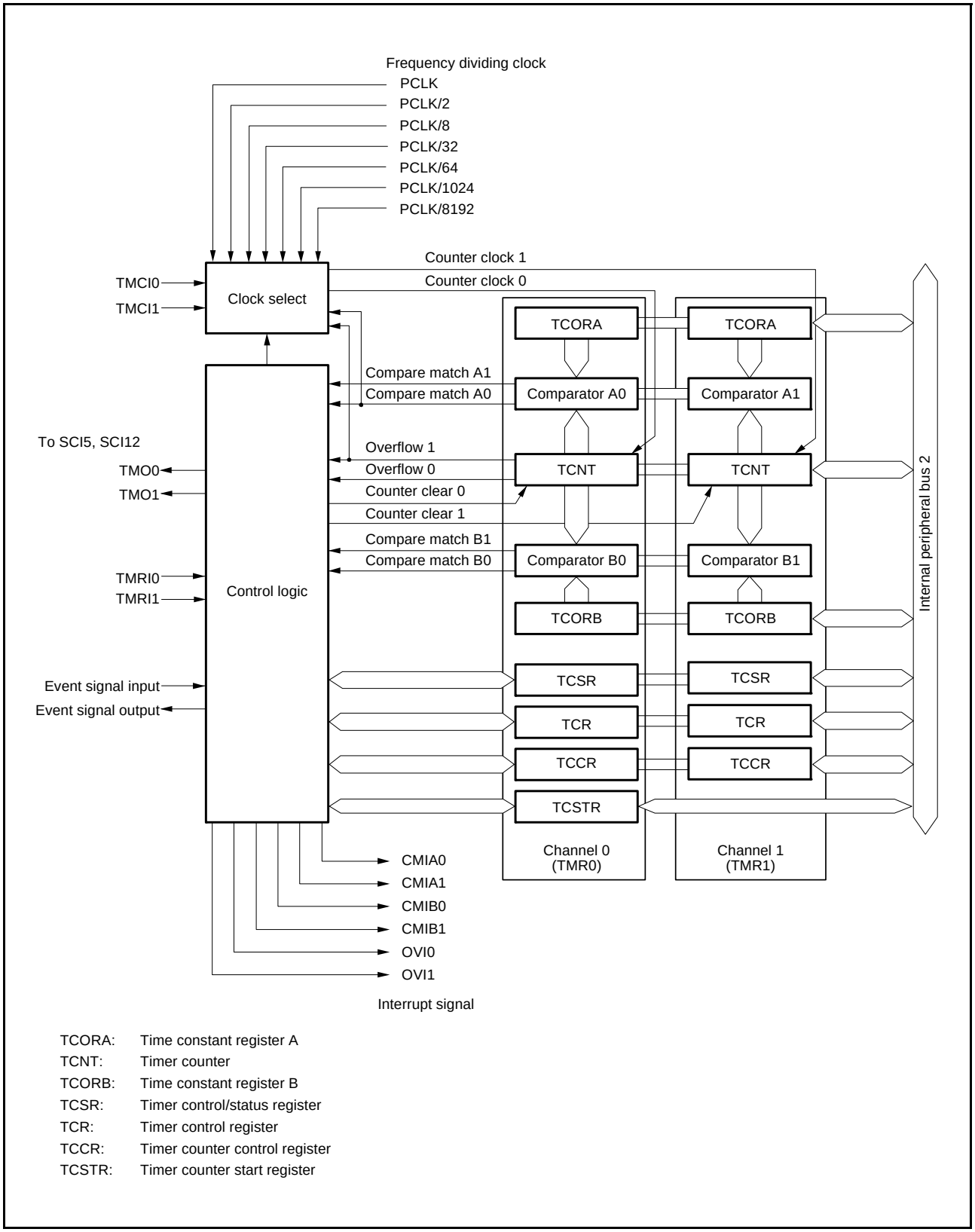
Item	Description
Count clock	• Frequency dividing clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter 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)	(1) Count start operation (TMR0, TMR2) (2) Event counter operation (TMR0, TMR2) (3) Count 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
Low power consumption facilities	Each unit can be placed in a module stop state

Note 1. For details, see section 27, Serial Communications Interface (SCIe, SCIf).

Table 23.2 Pin Configuration of TMR

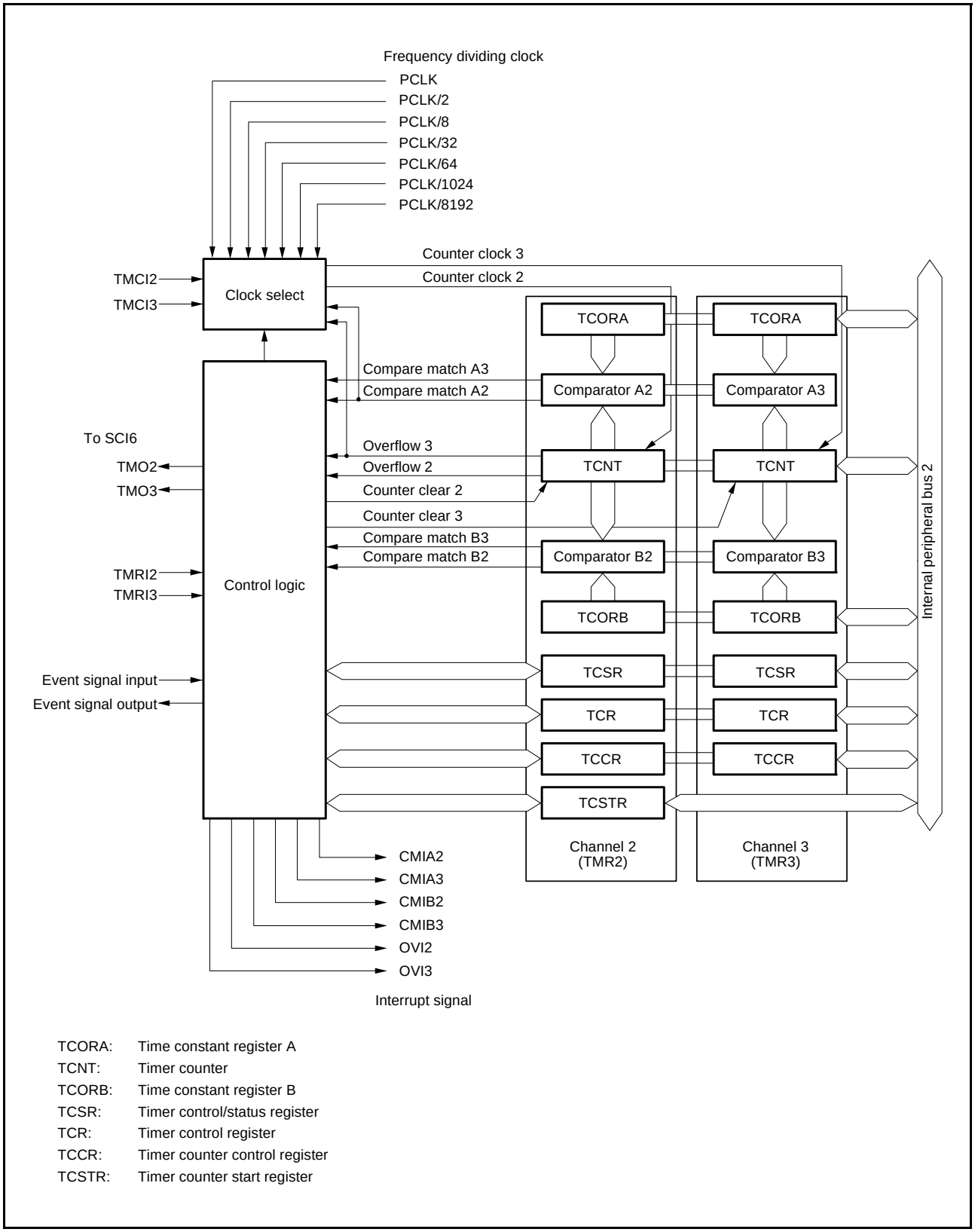
Item		Unit 0			Unit 1		
Counter mode		8 Bits		16 Bits	8 Bits		16 Bits
Channel		TMR0	TMR1	TMR0 + TMR1	TMR2	TMR3	TMR2 + TMR3
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 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		○		—	○		—
Module stop setting*2		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

○: Possible
—: Impossible
Note 1. For details, see section 27, Serial Communications Interface (SCIe, SCIf).
Note 2. For details, see section 11, Low Power Consumption.



TCORA: Time constant register A
TCNT: Timer counter
TCORB: Time constant register B
TCSR: Timer control/status register
TCR: Timer control register
TCCR: Timer counter control register
TCSTR: Timer counter start register

Figure 23.1 Block Diagram of TMR (Unit 0)



TCORA: Time constant register A
TCNT: Timer counter
TCORB: Time constant register B
TCSR: Timer control/status register
TCR: Timer control register
TCCR: Timer counter control register
TCSTR: Timer counter start register

Figure 23.2 Block Diagram of TMR (Unit 1)

Table 23.3 lists the input/output pins of the TMR.

Table 23.3 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

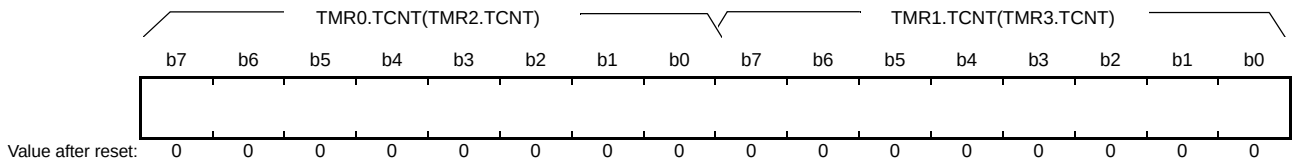
23.2 Register Descriptions

Table 23.4 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

23.2.1 Timer Counter (TCNT)

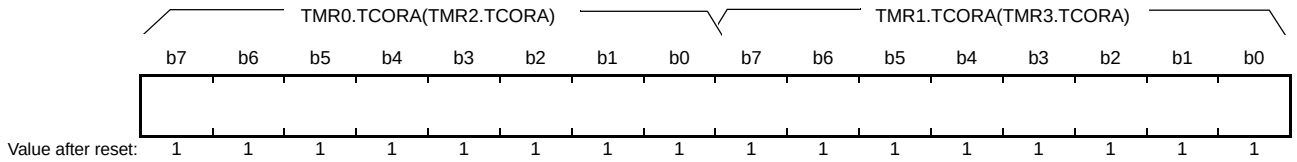
Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h



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 23.6, TMR Interrupt Sources.

23.2.2 Time Constant Register A (TCORA)

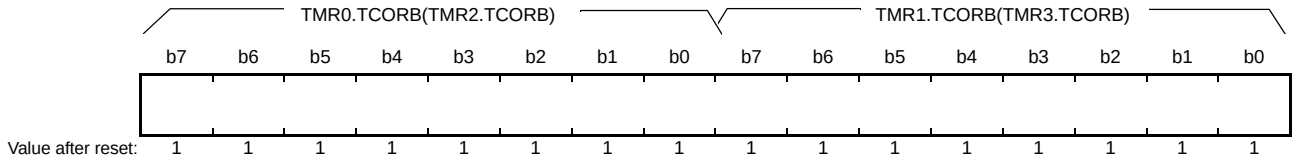
Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h



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.

23.2.3 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 TCORBn. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSRn.OSB[1:0] bits.

23.2.4 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 19, 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.

23.2.5 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 23.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 23.5.	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 19, 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 23.5.

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 23.5 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 19, 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.

23.2.6 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.

23.2.7 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 23.7, Link Operation by ELC, or section 18, Event Link Controller (ELC).

23.3 Operation

23.3.1 Pulse Output

Figure 23.3 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.

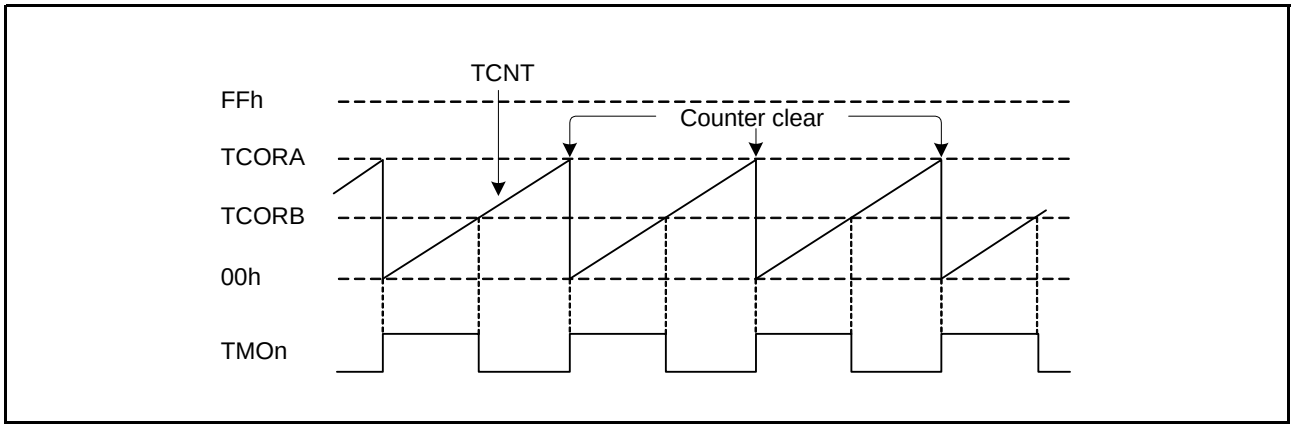


Figure 23.3 Example of Pulse Output (n = 0 to 3)

23.3.2 Reset Input

Figure 23.4 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.

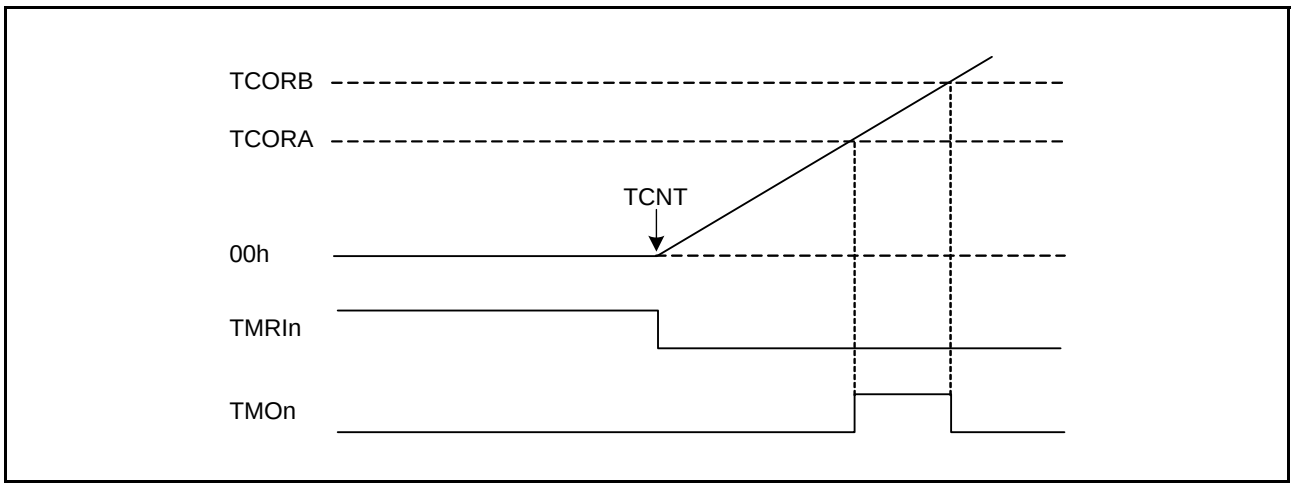


Figure 23.4 Example of Reset Input (n = 0 to 3)

23.4 Operation Timing

23.4.1 TCNT Count Timing

Figure 23.5 shows the count timing of TCNT for frequency dividing clock input. Figure 23.6 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.

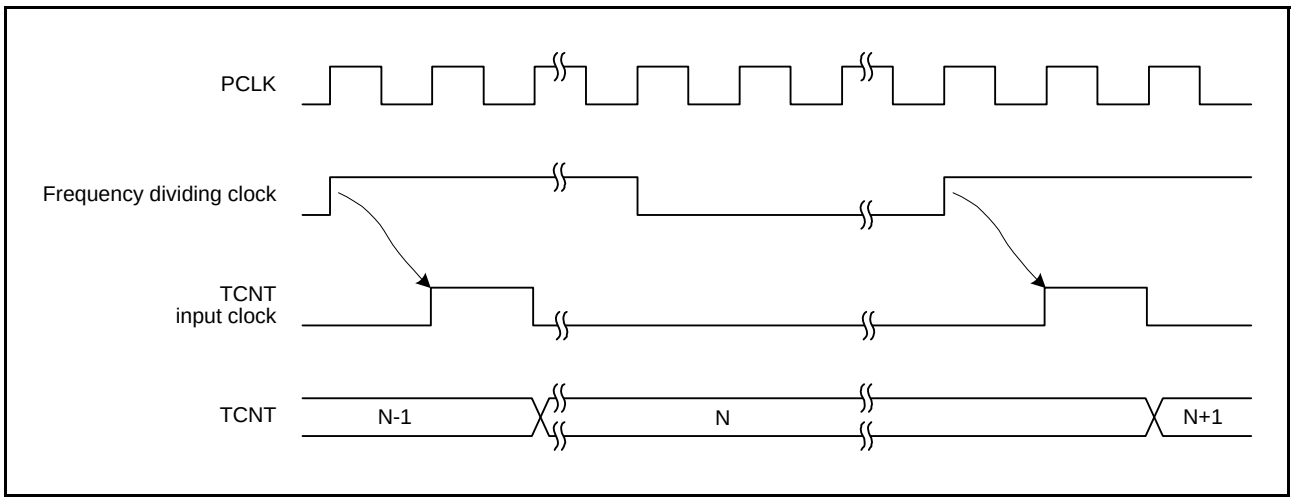


Figure 23.5 Count Timing for Frequency Dividing Clock Input

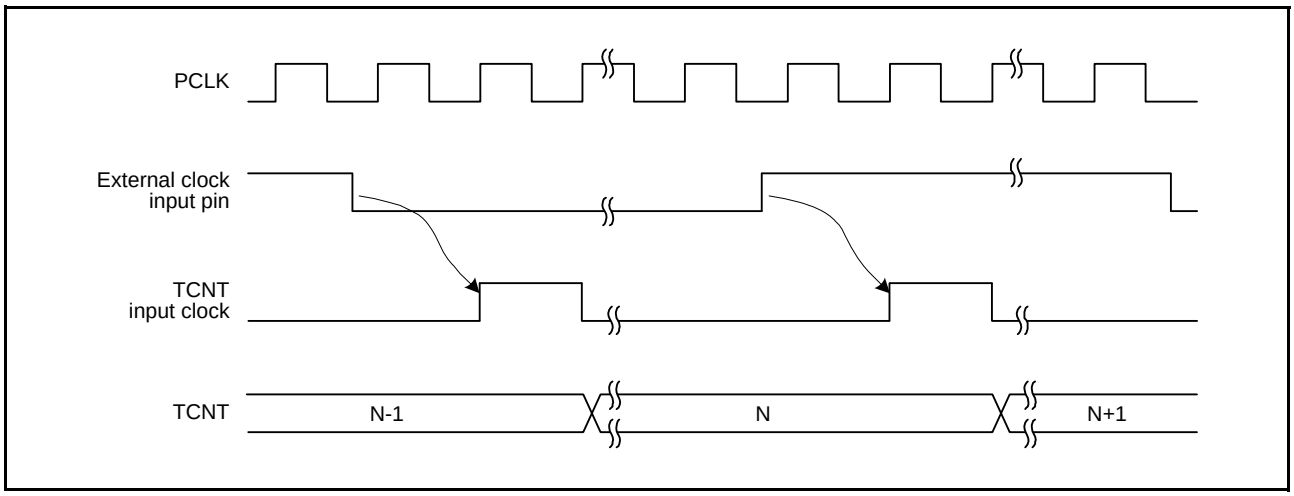


Figure 23.6 Count Timing for External Clock Input (at Both Edges)

23.4.2 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 23.7 shows the timing of output of the interrupt signal.

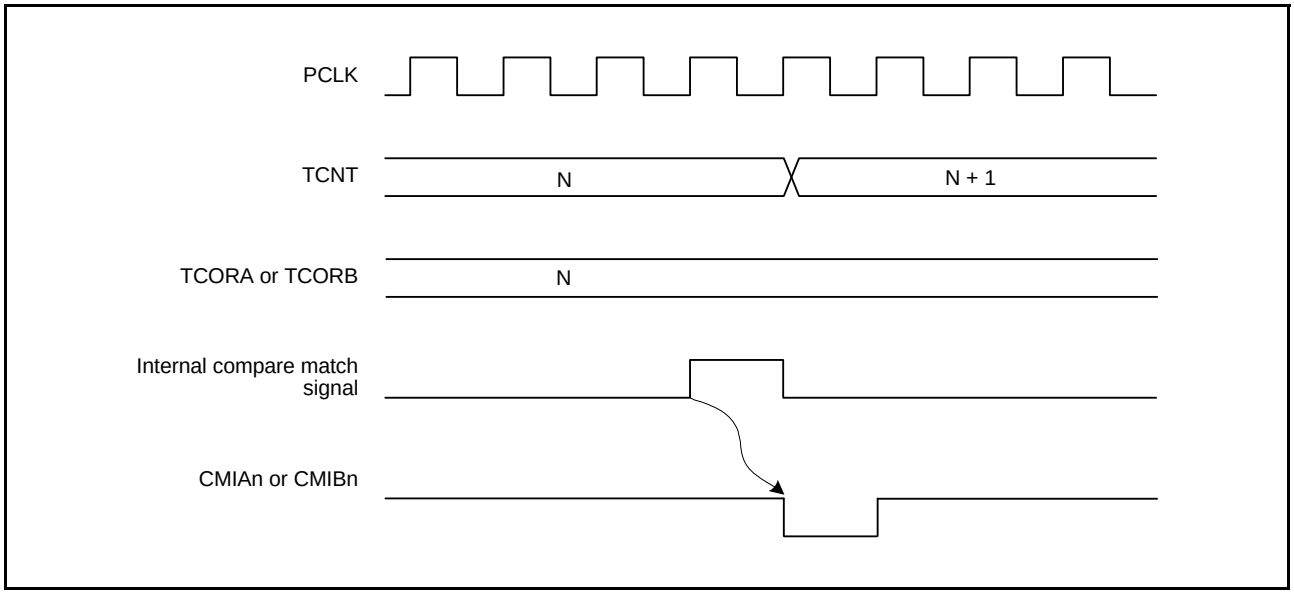


Figure 23.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

23.4.3 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 23.8 shows the timing when the timer output is toggled by the compare match A signal.

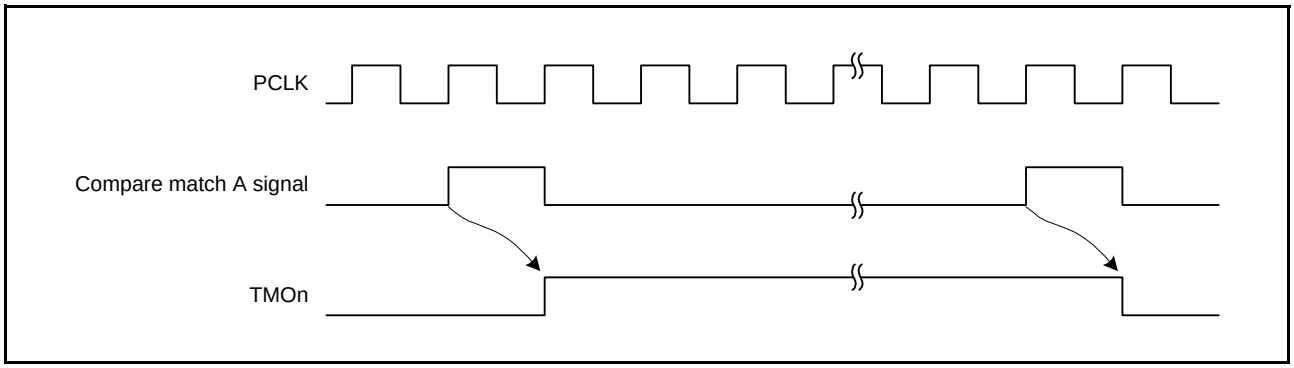


Figure 23.8 Timing of Timer Output at Compare Match A Signal (n = 0 to 3)

23.4.4 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 23.9 shows the timing of this operation.

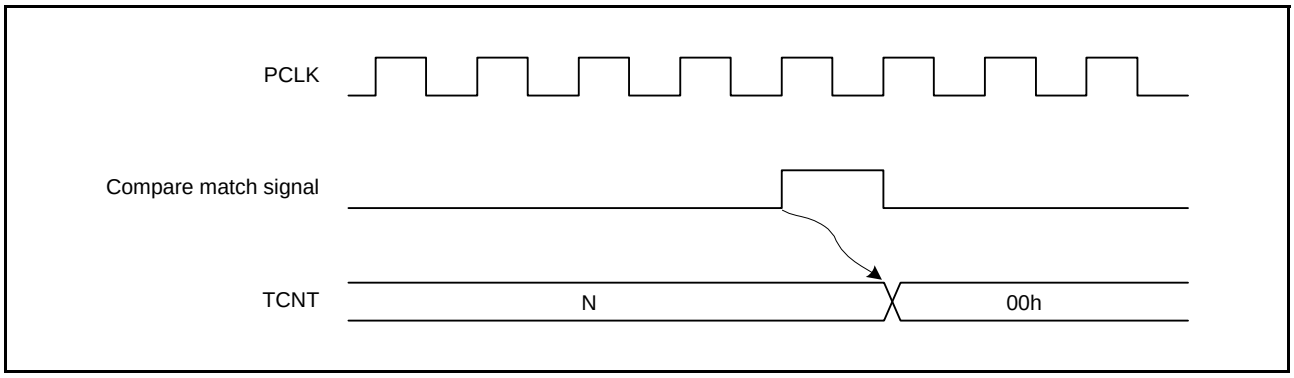


Figure 23.9 Timing of Counter Clear by Compare Match

23.4.5 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 23.10 and Figure 23.11 show the timing of this operation.

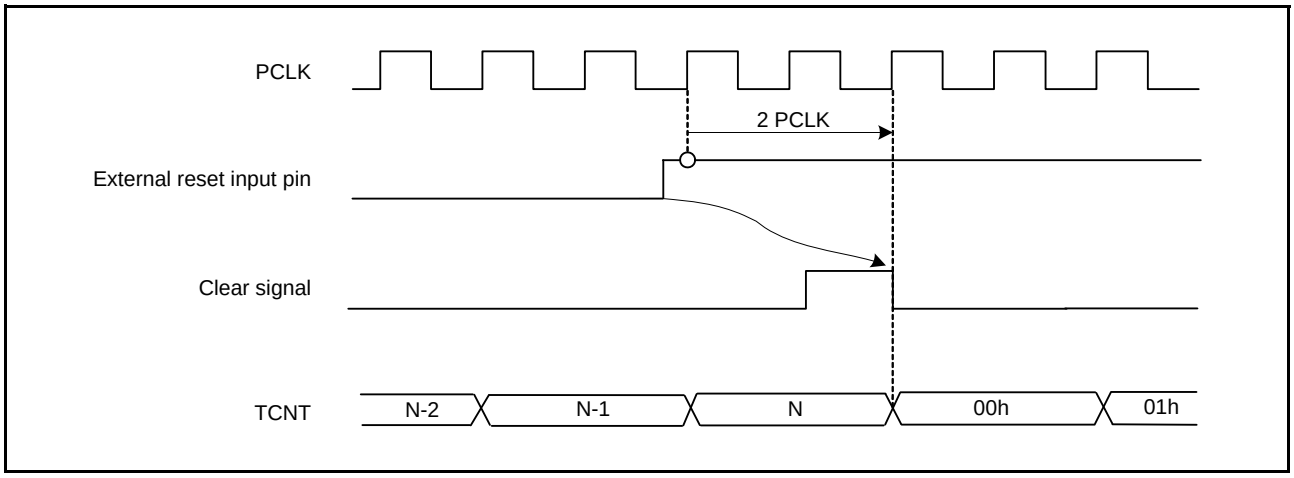


Figure 23.10 Timing of Clearance by External Reset (Rising Edge)

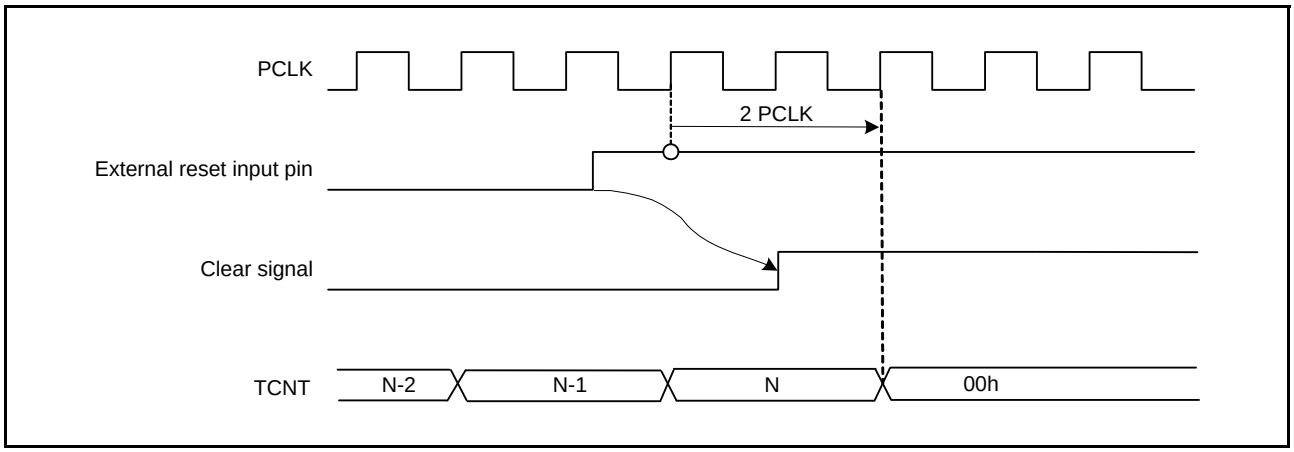


Figure 23.11 Timing of Clearance by External Reset (High Level)

23.4.6 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 23.12 shows the timing of output of the interrupt signal.

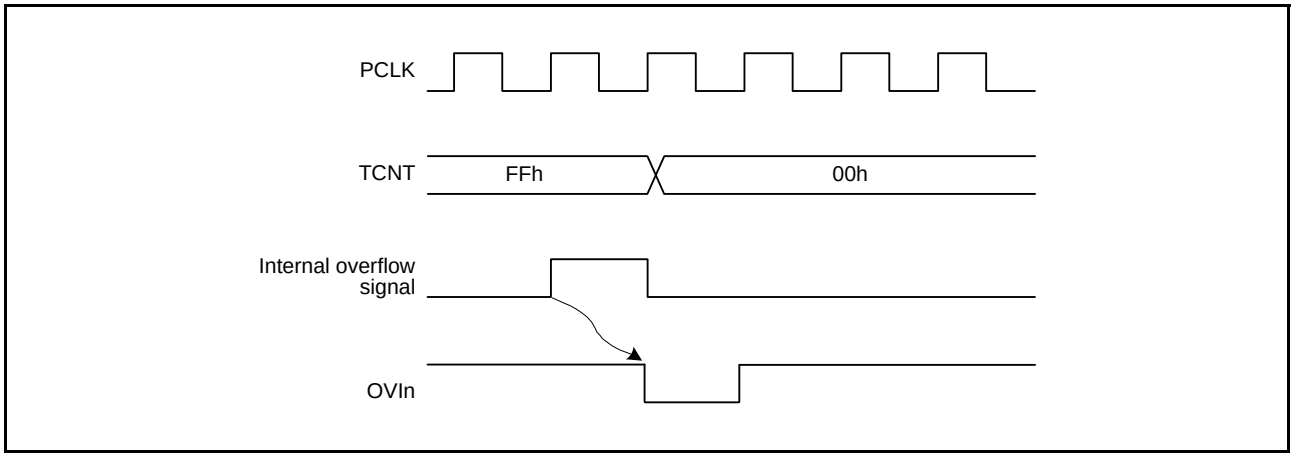


Figure 23.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

23.5 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.

23.5.1 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.

23.5.2 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.

23.6 Interrupt Sources

23.6.1 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 23.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts. The DMAC cannot be activated by the interrupt sources for TMRn.

Table 23.6 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

23.7 Link Operation by ELC

23.7.1 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 18, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

23.7.2 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.

23.7.3 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.

23.8 Usage Notes

23.8.1 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.

23.8.2 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).

$$f = PCLK / (N + 1)$$

23.8.3 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 23.13.

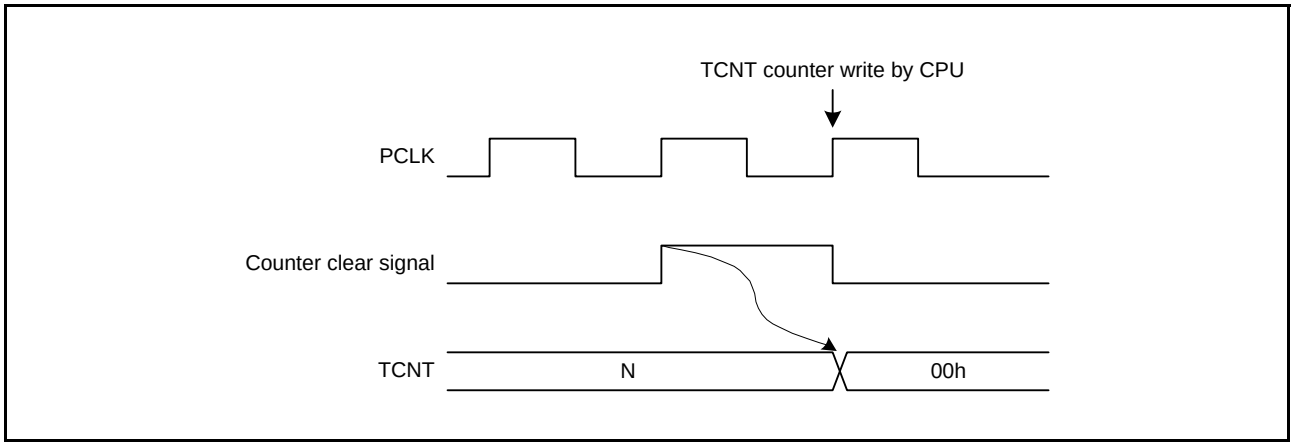


Figure 23.13 Conflict between TCNT Write and Counter Clear

23.8.4 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 23.14.

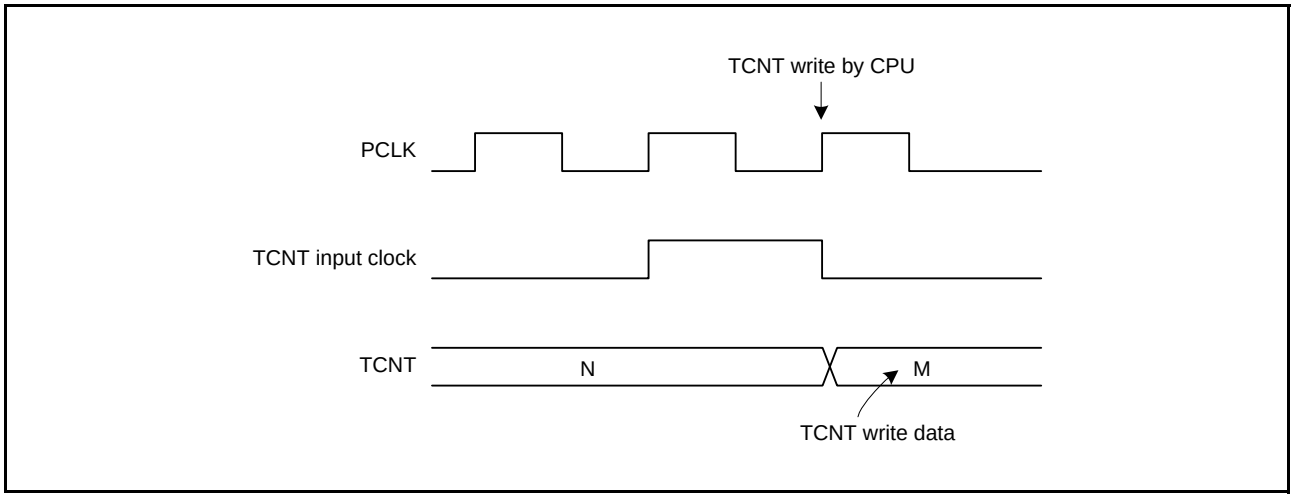


Figure 23.14 Conflict between TCNT Write and Increment

23.8.5 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, the write takes priority and the compare match signal is not generated.

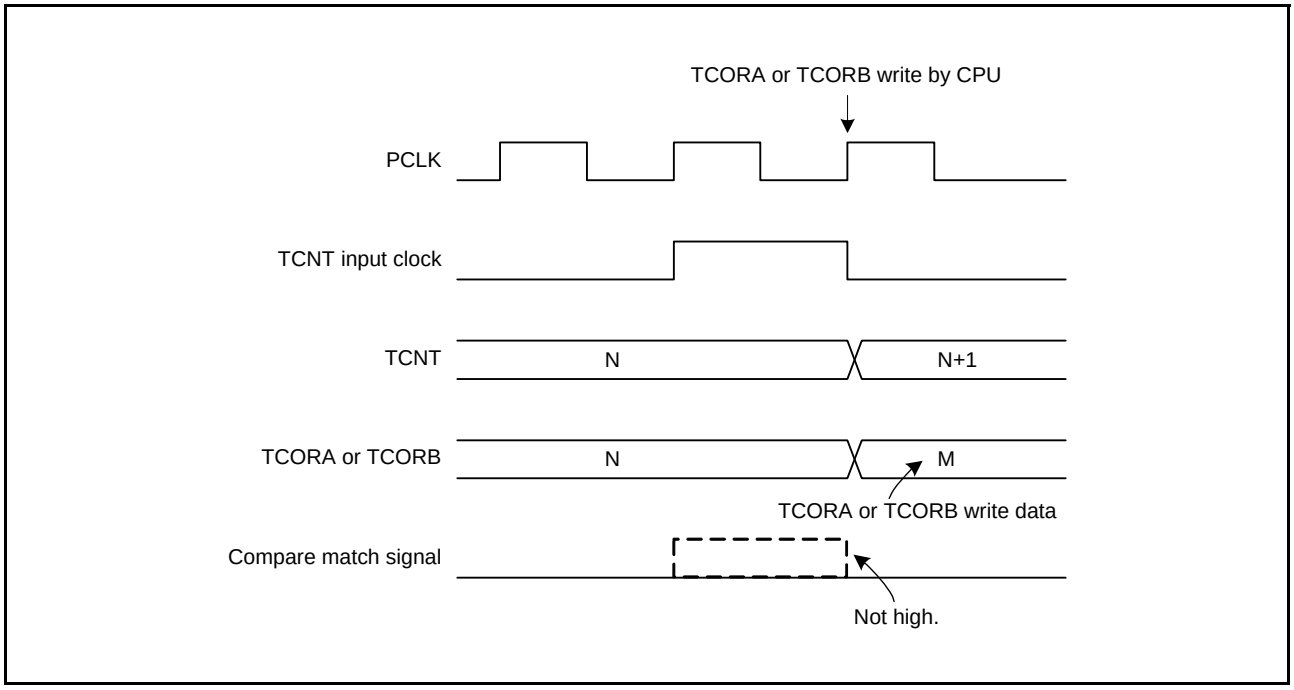


Figure 23.15 Conflict between TCORA or TCORB Write and Compare Match

23.8.6 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 23.7.

Table 23.7 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

23.8.7 Switching of Frequency Dividing Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the frequency dividing clock is switched. Table 23.8 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 23.8, 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 23.8 Switching of Frequency Dividing Clocks and TCNT Operation (1/2)

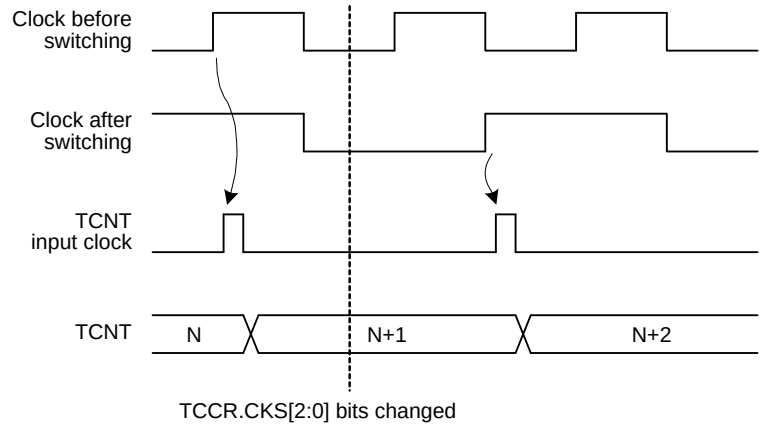
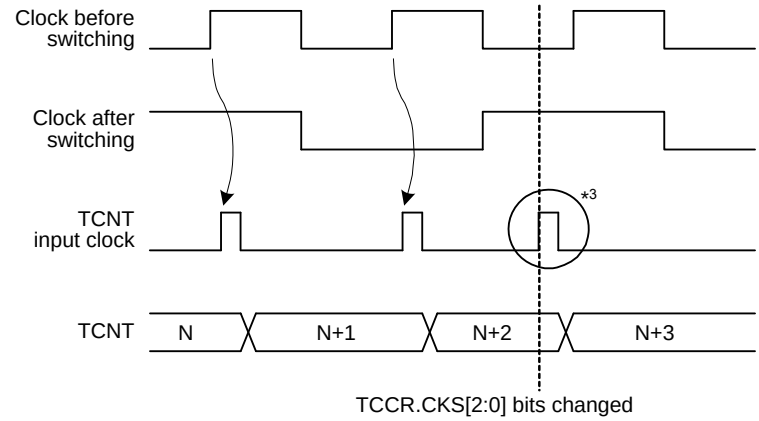
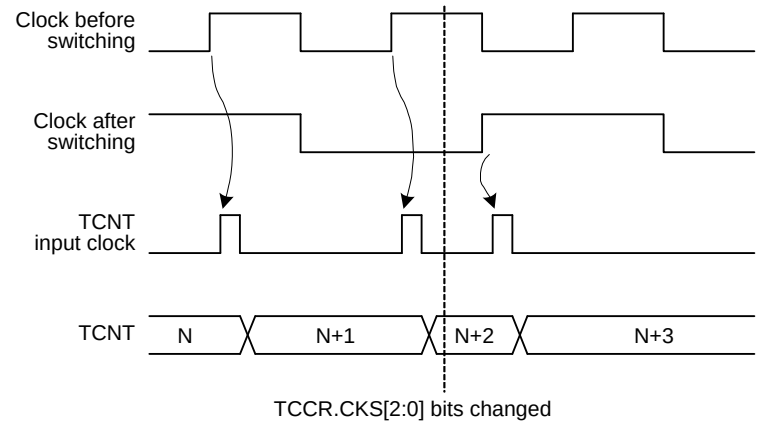
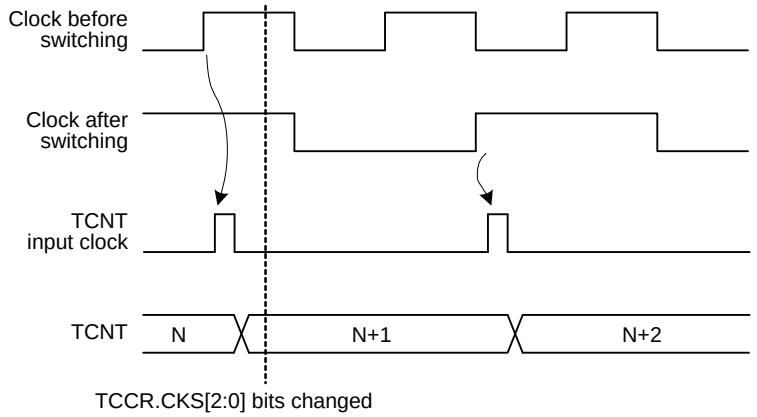
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	
3	Switching from high to low*4	

Table 23.8 Switching of Frequency Dividing Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
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.

23.8.8 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.

23.8.9 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 23.16 shows operation timing when the compare match interrupt signal is continuously output.

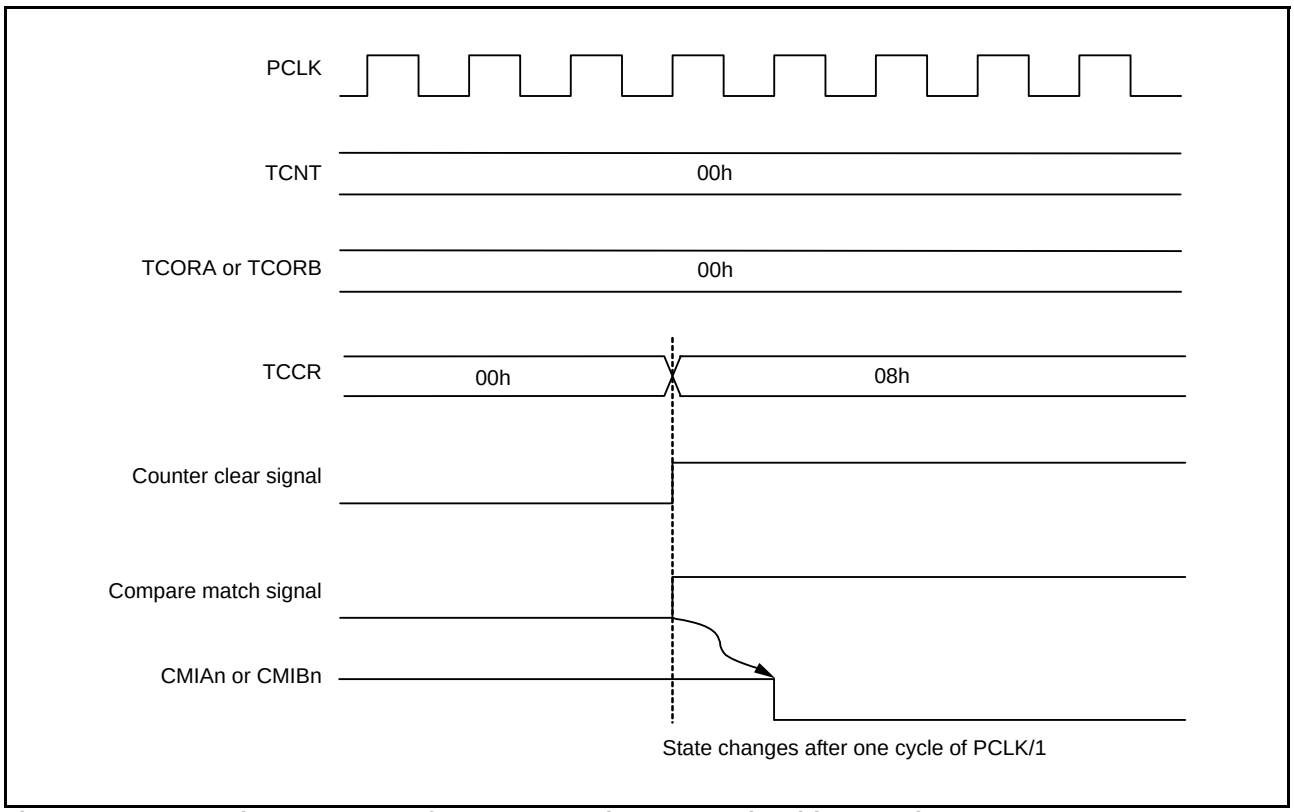


Figure 23.16 Continuous Output of Compare Match Interrupt Signal (n =0 to 3)

24. Compare Match Timer (CMT)

The RX220 Group 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.

24.1 Overview

Table 24.1 lists the specifications for the CMT.
Figure 24.1 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.

Table 24.1 Specifications of CMT

Item	Description
Count clock	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected individually for each channel.
Interrupt	A compare match interrupt can be requested individually for each channel.
Low power consumption facilities	Each unit can be placed in a module stop state.

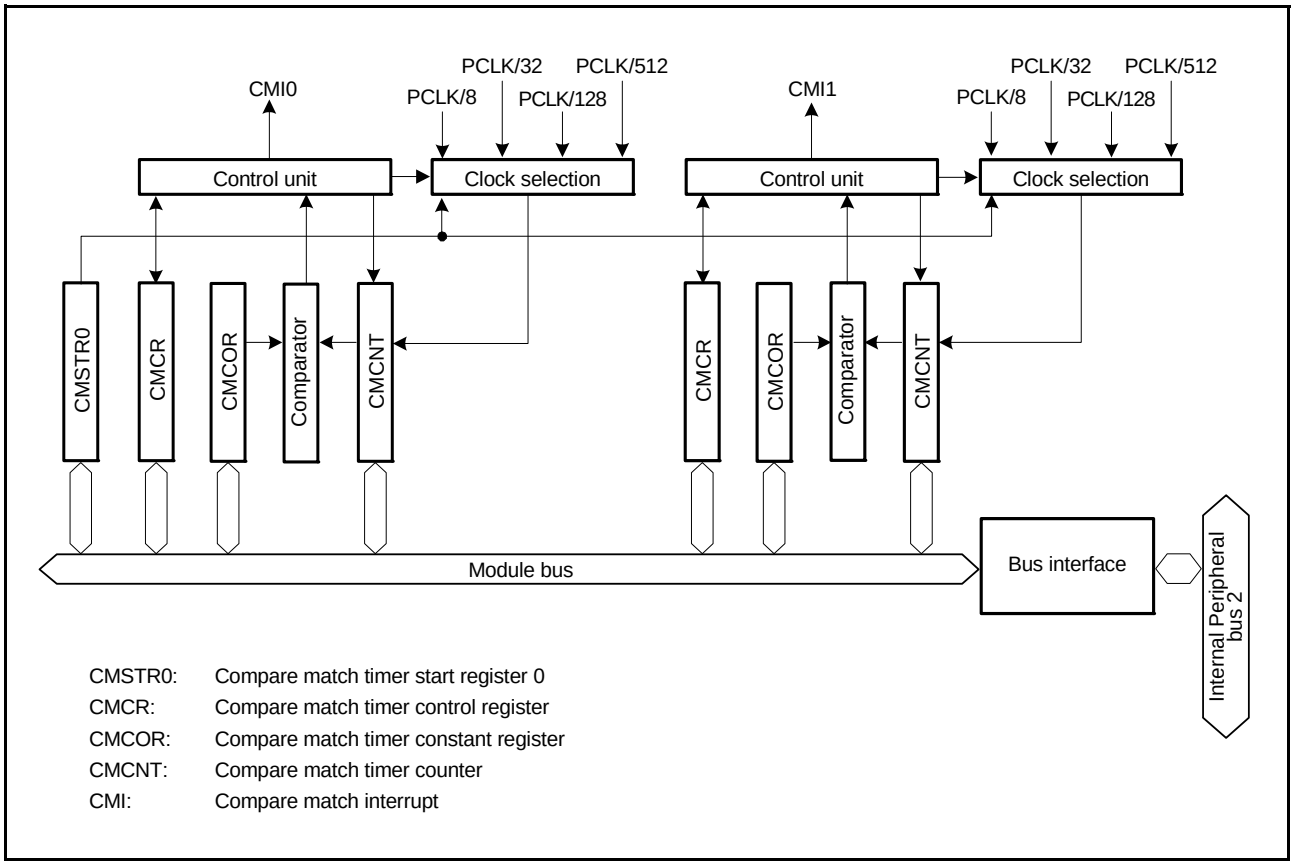


Figure 24.1 Block Diagram of CMT (Unit 0)

24.2 Register Descriptions

24.2.1 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

24.2.2 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

24.2.3 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	The read value is 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 clock to be input to CMCNT from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the STRn (n = 0 to 3) bit in CMSTRm (m = 0, 1) is set to 1, CMCNT starts counting up on the clock selected with bits CKS[1:0].

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when CMCNT and CMCOR values match.

24.2.4 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

CMCNT is a readable/writable up-counter to generate interrupt requests.
When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0, 1) is set to 1, CMCNT starts counting up using the selected clock.
When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

24.2.5 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

CMCOR is a readable/writable register to set a compare match cycle with CMCNT.

24.3 Operation

24.3.1 Periodic Count Operation

When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0, 1) is set to 1, CMCNT starts counting up using the selected clock.
When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated. CMCNT then starts counting up from 0000h. Figure 24.2 shows the operation of the CMCNT counter.

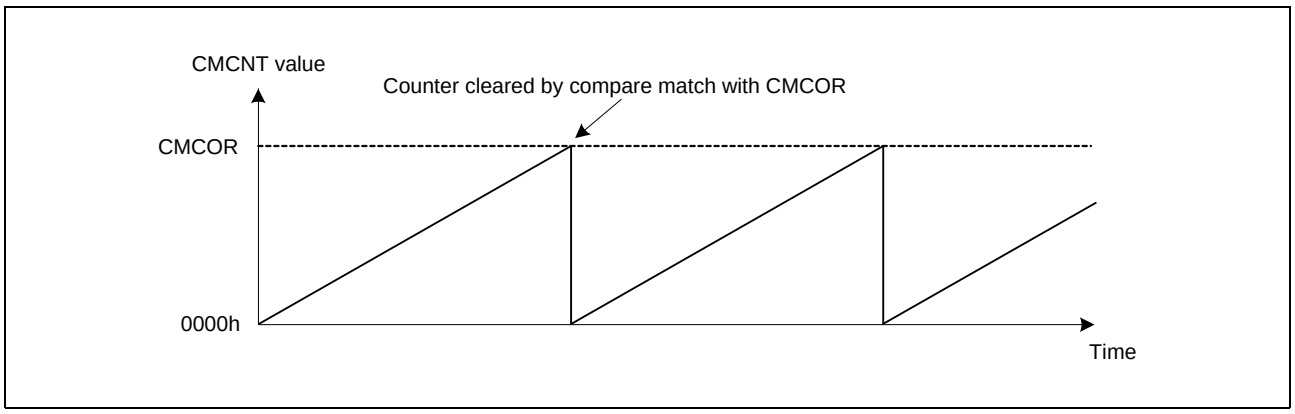


Figure 24.2 CMCNT Counter Operation

24.3.2 CMCNT Count Timing

As the count clock to be input to CMCNT, 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 CKS[1:0] bits in CMCR. Figure 24.3 shows the timing of CMCNT.

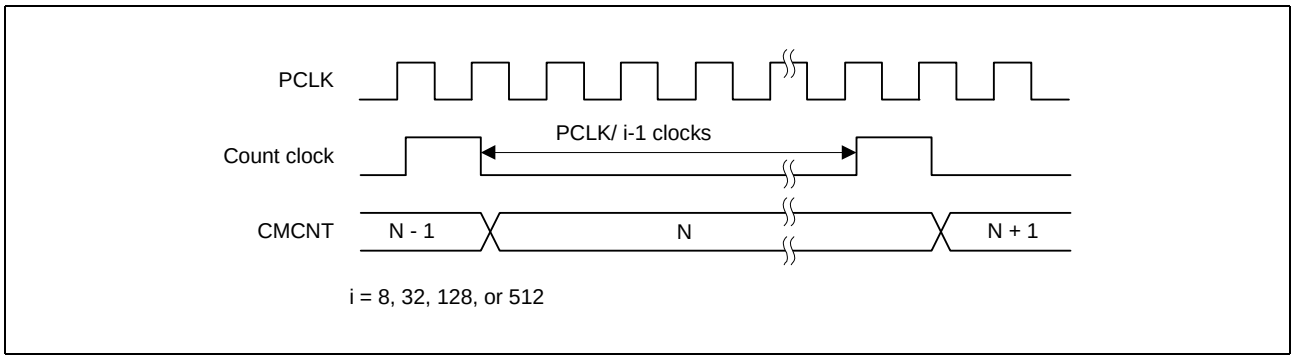


Figure 24.3 CMCNT Count Timing

24.4 Interrupts

24.4.1 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 is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUb).

Table 24.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match between CMT0.CMCNT and CMT0.CMCOR	Possible	Possible
CMI1	Compare match between CMT1.CMCNT and CMT1.CMCOR	Possible	Possible
CMI2	Compare match between CMT2.CMCNT and CMT2.CMCOR	Possible	Possible
CMI3	Compare match between CMT3.CMCNT and CMT3.CMCOR	Possible	Possible

24.4.2 Timing of Compare Match Interrupt Generation

When CMCNT and CMCOR 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 CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT input clock.
Figure 24.4 shows the timing of a compare match interrupt.

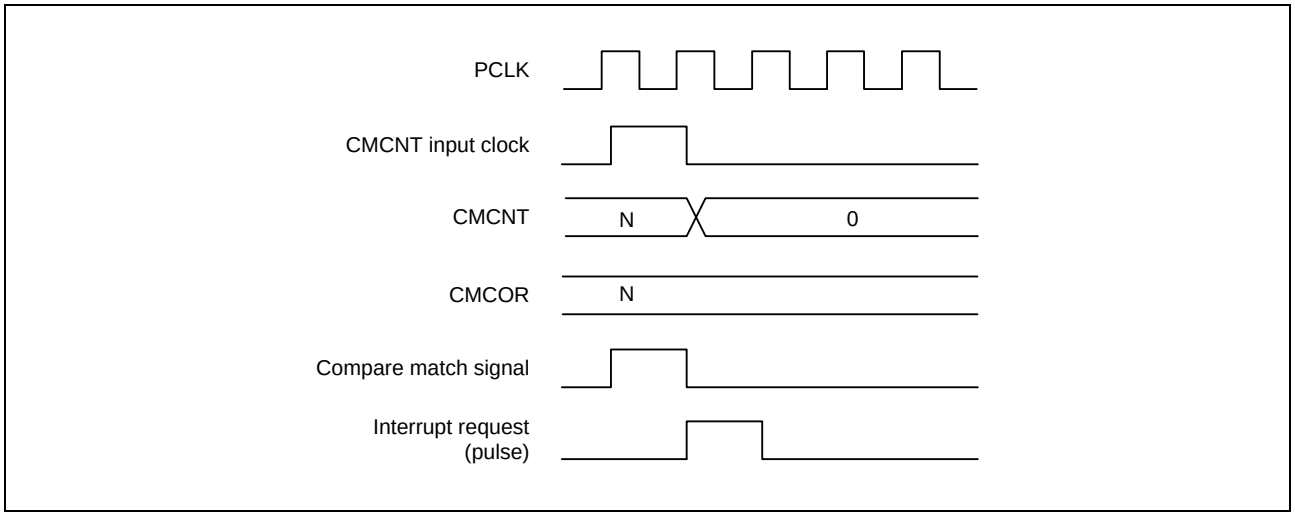


Figure 24.4 Timing of a Compare Match Interrupt

24.5 Usage Notes

24.5.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. The CMT is disabled by default. The registers can be accessed by canceling the module stop state. For details, see section 11, Low Power Consumption.

24.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 24.5 shows the timing to clear the CMCNT counter.

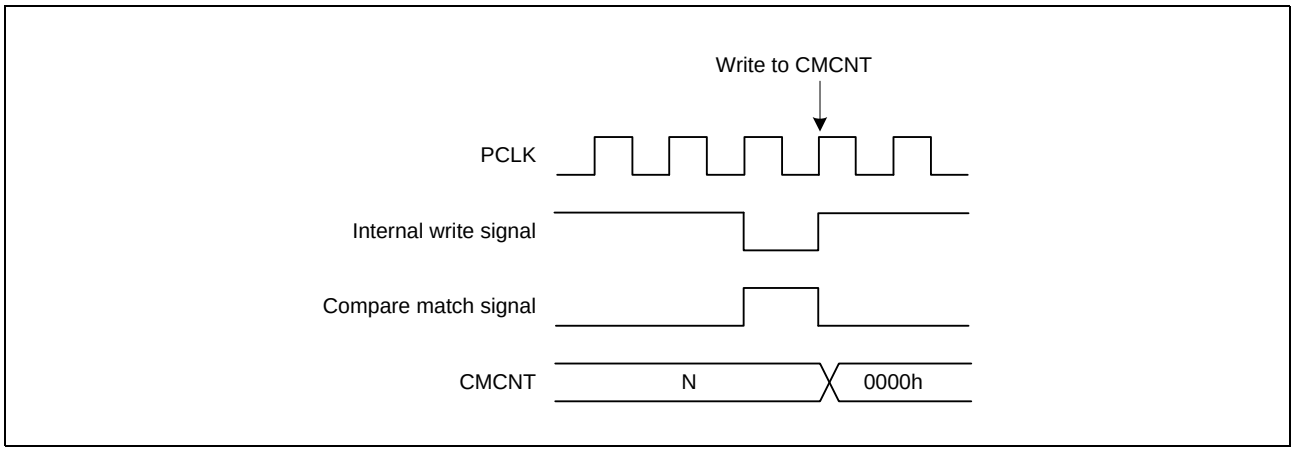


Figure 24.5 Conflict between Write and Compare Match Processes of CMCNT

24.5.3 Conflict between Write and Count-Up Processes of CMCNT

Even when the count-up occurs while writing to CMCNT, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 24.6 shows the timing to write the CMCNT counter.

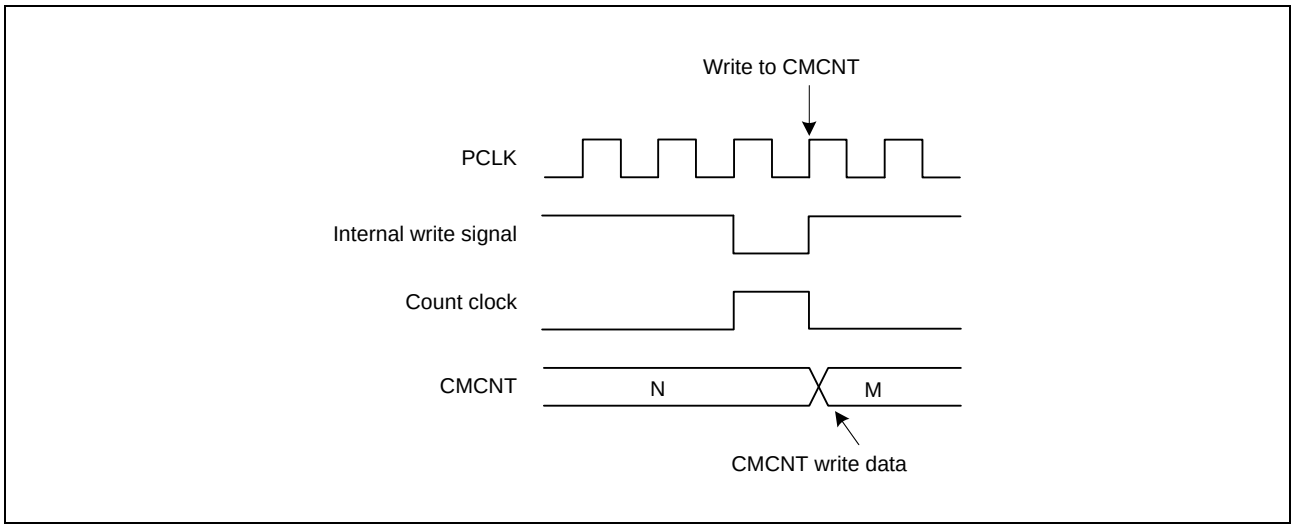


Figure 24.6 Conflict between Write and Count-Up Processes of CMCNT

25. Realtime Clock (RTCc)

25.1 Overview

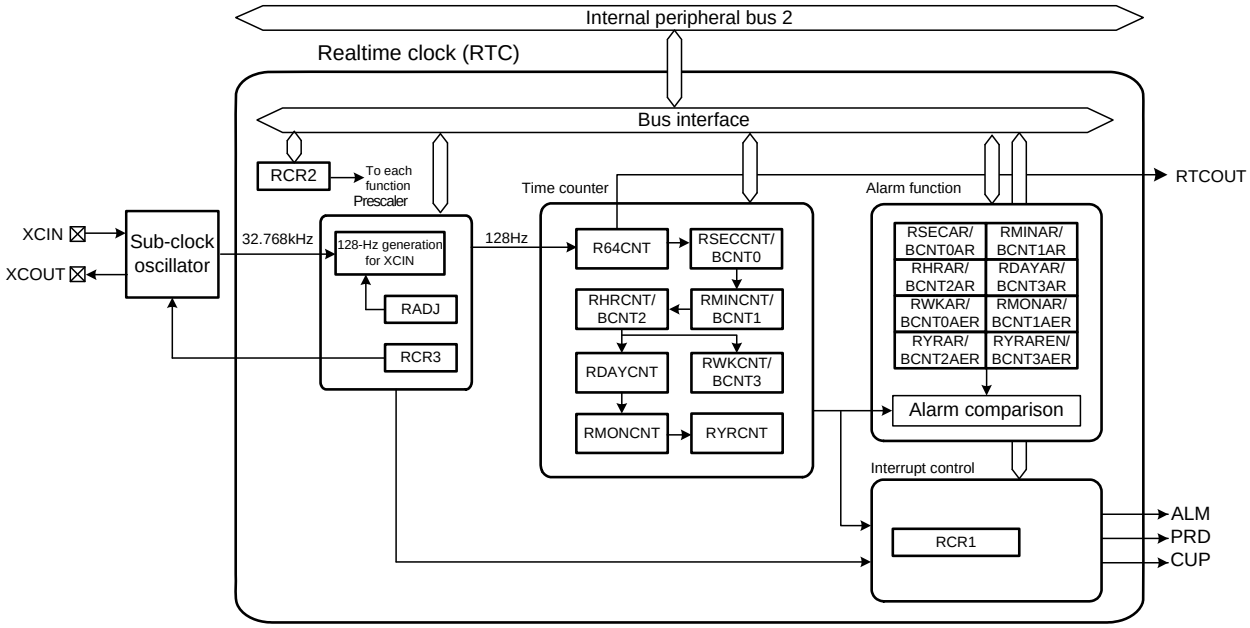
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
In calendar count mode, it operates as a clock counter that counts 100 years from year 00 to year 99. The order of hundreds and thousands are assumed as 20 and the leap years from the year 2000 to 2099 are automatically corrected.
In binary count mode, it operates as a 32-bit binary counter.
The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the basic clock. Year, month, date, day-of-week, am/pm (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

Table 25.1 lists the specifications of the RTC, Figure 25.1 shows a block diagram of the RTC, and Table 25.2 shows the pin configuration of the RTC.

Table 25.1 Specifications of RTC

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN)
Clock and calendar functions	• Calendar count mode Year, month, date, day-of-week, time, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The state of 1-Hz, 2-Hz, 4-Hz, 8-Hz, 16-Hz, 32-Hz, or 64-Hz is displayed in binary units Clock error correction function Clock (1-Hz/64-Hz) output
Interrupts	• Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) Indicates occurrence of a carry to the second counter/binary counter 0 or a carry to the 64-Hz counter during reading of the 64-Hz counter • Recovery from software standby mode can be performed by an alarm interrupt or periodic interrupt

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.



R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second counter/Binary counter 0
RMINCNT/BCNT1: Minute counter/Binary counter 1
RHRCNT/BCNT2: Hour counter/Binary counter 2
RWKCNT/BCNT3: Day-of-week counter/Binary counter 3
RDAYCNT: Date counter
RMONCNT: Month counter
RYRCNT: Year counter
RCR1: RTC control register 1
RCR2: RTC control register 2
RCR3: RTC control register 3
RADJ: Time error adjustment register

RSECAR/BCNT0AR: Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR: Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR: Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR: Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER: Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER: Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER: Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER: Year alarm enable register/Binary counter 3 alarm enable register

Figure 25.1 Block Diagram of RTC

Table 25.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal resonator for the RTC to these pins.
XCOUT	Output	
RTCOUT	Output	Output a 1-Hz/64-Hz clock

25.2 Register Descriptions

Furthermore, when writing to or reading from RTC registers, do so in accord with section 25.5.5, Points for Caution when Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. However, note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode just after setting any of these registers. For details, refer to section 25.5.4, Transitions to Low Power Consumption Modes after Setting Registers.

25.2.1 64-Hz Counter (R64CNT)

Address(es): 0008 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64Hz	Indicate the state between 1-Hz and 64-Hz.	R
b1	F32HZ	32Hz		R
b2	F16HZ	16Hz		R
b3	F8HZ	8Hz		R
b4	F4HZ	4Hz		R
b5	F2HZ	2Hz		R
b6	F1HZ	1Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock.

The state in the sub-seconds range can be confirmed by reading this counter.

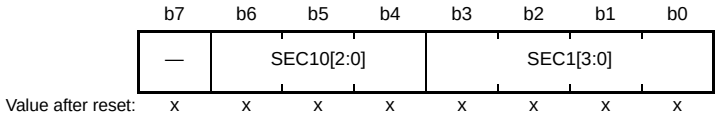
This counter is cleared to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): 0008 C402h



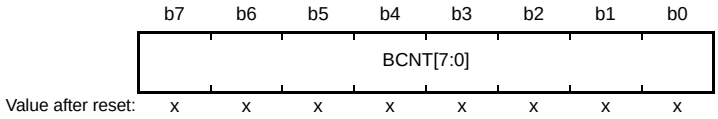
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	Ones Place of Seconds	Counts from 0 to 9 once per second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	Tens Place of Seconds	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RSECCNT is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C402h



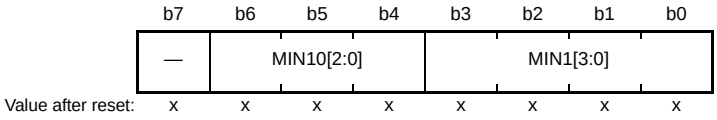
x: Undefined

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): 0008 C404h



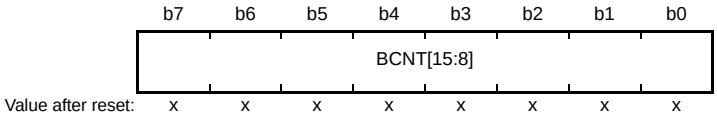
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	Ones Place of Minutes	Counts from 0 to 9 once per minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	Tens Place of Minutes	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RMINCNT is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C404h



x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	Ones Place of Hours	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	Tens Place of Hours	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RHRCNT is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (practically in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C406h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): 0008 C408h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as 0.	R/W

RWKCNT is used for setting and counting in the BCD-coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.6 Date Counter (RDAYCNT)

Address(es): 0008 C40Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	Ones Place of Days	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	Tens Place of Days	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode.
RDAYCNT is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.7 Month Counter (RMONCNT)

Address(es): 0008 C40Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	Ones Place of Months	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	Tens Place of Months	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode.
RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.8 Year Counter (RYRCNT)

Address(es): 0008 C40Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

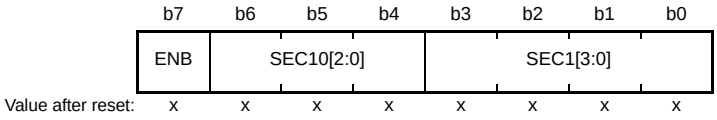
Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	Ones Place of Years	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	Tens Place of Years	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode.
RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.
A value from 00 through 99 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): 0008 C410h



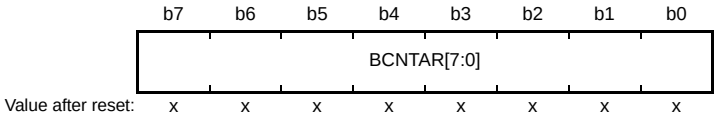
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RSECCNT value.	R/W

RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.
RSECAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C410h



x: Undefined

The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is cleared to 00h by an RTC software reset.

25.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMINCNT value.	R/W

RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.
RMINAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C412h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is cleared to 00h by an RTC software reset.

25.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RHRCNT value.	R/W

RHRAR is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.
The specifiable time differs according to the setting in the time mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (practically in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is cleared to 00h by an RTC software reset.

25.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RWKCNT value.	R/W

RWKAR is an alarm register corresponding to the BCD-coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RWKAR values from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is cleared to 00h by an RTC software reset.

25.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RDAYCNT value.	R/W

RDAYAR is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RDAYAR values from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the ICU IR92.IR flag becomes 1.

This register is cleared to 00h by an RTC software reset.

25.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMONCNT value.	R/W

RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.
RMONAR values from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the ICU IR92.IR flag becomes 1.
This register is cleared to 00h by an RTC software reset.

25.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the ICU IR92.IR flag becomes 1.
This register is cleared to 0000h by an RTC software reset.

25.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): 0008 C41Eh

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, the RYRAR value is compared with the RYRCNT value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the ICU IR92.IR flag becomes 1.
This register is cleared to 00h by an RTC software reset.

25.2.17 RTC Control Register 1 (RCR1)

Address(es): 0008 C422h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled 1: An alarm interrupt request is enabled	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1-Hz 1: RTCOUT outputs 64-Hz	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second. 0 1 1 1: A periodic interrupt is generated every 1/128 second. 1 0 0 0: A periodic interrupt is generated every 1/64 second. 1 0 0 1: A periodic interrupt is generated every 1/32 second. 1 0 1 0: A periodic interrupt is generated every 1/16 second. 1 0 1 1: A periodic interrupt is generated every 1/8 second. 1 1 0 0: A periodic interrupt is generated every 1/4 second. 1 1 0 1: A periodic interrupt is generated every 1/2 second. 1 1 1 0: A periodic interrupt is generated every 1 second. 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated in synchronization with the count source. When RCR1 is modified, check that all the bits have been updated without fail before continuing with further processing.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

CIE Bit (Carry Interrupt Enable)

This bit enables or disabled interrupt requests when a carry to the second counter/binary counter 0 occurred or a carry to the 64-Hz counter occurred during read access to the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.

RTCOS Bit (RTCOUT Output Select)

This bit is to select the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled and port control must also be enabled.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt (PRD) is generated with the period specified by these bits.

25.2.18 RTC Control Register 2 (RCR2)

(1) In calendar count mode:

Address(es): 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start	0: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler are stopped. 1: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and target registers are reset by RTC software reset. (R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP) • In reading 0: In normal time operation, or RTC software reset has completed. 1: During an RTC software reset	R/W
b2	ADJ30	30-Second Adjustment	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal time operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every one minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every ten seconds.	R/W
b6	HR24	Hours Mode	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select	0: The calendar count mode. 1: The binary count mode.	R/W

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated without fail, and then make next settings.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.
When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically cleared to 0.
When 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to one minute.
The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically cleared to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is cleared to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is cleared to 0 by an RTC software reset.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJE bit.
The AADJE bit is cleared to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJP bit.
The AADJP bit is cleared to 0 by an RTC software reset.

HR24 Bit (Hours Mode)

This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
After a new value is written to this bit, reflecting the value in the internal circuitry takes a certain amount of time. Read the bit to confirm whether a value written has actually been reflected.

(2) In binary count mode:

Address(es): 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start	0: The 32-bit binary counter, 64-Hz counter, and prescaler are stopped. 1: The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and registers to be reset by RTC software are reset. (R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP) • In reading 0: In normal time operation, or RTC software reset has completed. 1: During an RTC software reset	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select	0: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds 1: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every eight seconds	R/W
b6	—	Reserved	This bit is undefines. The write value should be 0.	R/W
b7	CNTMD	Count Mode Select	0: The calendar count mode. 1: The binary count mode.	R/W

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated without fail, and then make next settings.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically cleared to 0.

When 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write “0” to the START bit) and change the value of the RTCOE bit at the same time.
When a RTCOUT signal is to be output from an external pin, enable the port control as well as setting this bit.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJE bit.
The AADJE bit is cleared to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Correction period can be selected from 32 second units or 8 second units in binary count mode.
Set the plus–minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJP bit.
The AADJP bit is cleared to 0 by an RTC software reset.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode. The CNTMD bit is updated synchronously with the count source.

25.2.19 RTC Control Register 3 (RCR3)

Address(es): 0008 C426h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RTCDV[2:0]			RTCEN
Value after reset:	0	0	0	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-clock Oscillator Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is operating.	R/W
b3 to b1	RTCDV[2:0]	Sub-clock Oscillator Drive Ability Control	b3 b1 0 0 0: Setting prohibited 0 0 1: Drive ability for low CL 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Drive ability for standard CL 1 1 1: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RTCEN Bit (Sub-clock Oscillator Control)

Running and stopping of the sub-clock oscillator is controlled by the RTCEN bit and a clock generation circuit register. When using the sub-clock as the count source to the RTC, set up the sub-clock oscillator using the RTCEN bit.

RTCDV[2:0] Bits (Sub-clock Oscillator Drive Ability Control)

These bits control the drive ability of the sub-clock oscillator. When connecting a standard CL crystal unit, set these bits to 110b (drive ability for standard CL). When connecting a low CL crystal unit, set these bits to 001b (drive ability for low CL). Set the RTCDV[2:0] bits while the SOSCCR.SOSTP bit is 1 and the RCR3.RTCEN bit is 0.

(1) Notes on using a low CL crystal unit

When the RCR3.RTCDV[2:0] bits are 001b (drive ability for low CL), the oscillator is susceptible to noise. Especially when the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1187EJ) to reduce the influence from the noise.

The following are examples that may significantly affect oscillation accuracy:

- When connecting an on-chip debugging emulator to the FINED pin

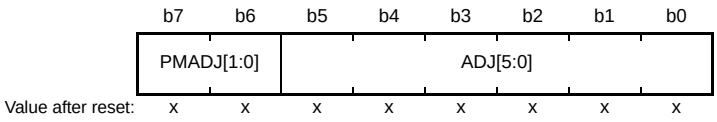
Since the FINED pin (FINE interface pin) is near the XCIN and XCOUT pins, the oscillation accuracy of the sub-clock oscillator is affected when using the FINED pin during debugging. When using the FINED pin during debugging, keep using the low CL crystal unit and set the RCR3.RTCDV[2:0] bits to 110b (drive ability for standard CL).However, this measure may affect the reliability of the crystal unit. Therefore, use this measure only when using an on-chip debugging emulator. Set the RCR3.RTCDV[2:0] bits to 001b (drive ability for low CL) in mass production programs.

- When supplying an external clock to the main clock oscillator

When inputting an external clock to the EXTAL pin, the oscillation accuracy of the sub-clock oscillator may be affected.

25.2.20 Time Error Adjustment Register (RADJ)

Address(es): 0008 C42Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

The RADJ register is used in both calendar count mode and in binary count mode.
Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated without fail before continuing with further processing.
This register is cleared to 00h by an RTC software reset.

ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

25.3 Operation

25.3.1 Outline of Initial Settings of Registers after Power-On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, and interrupt should be performed.

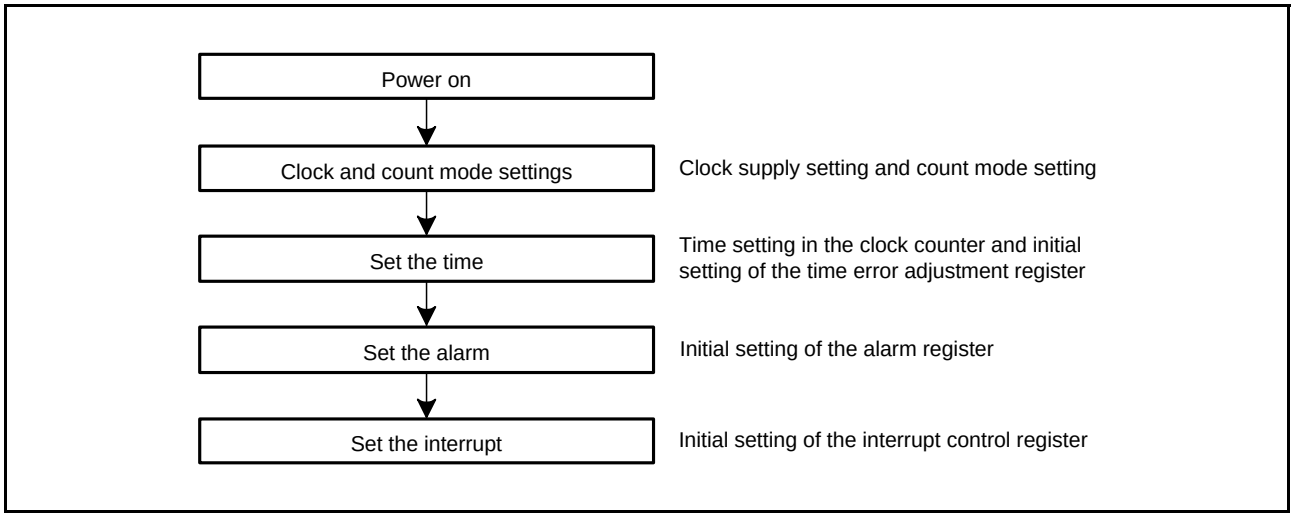
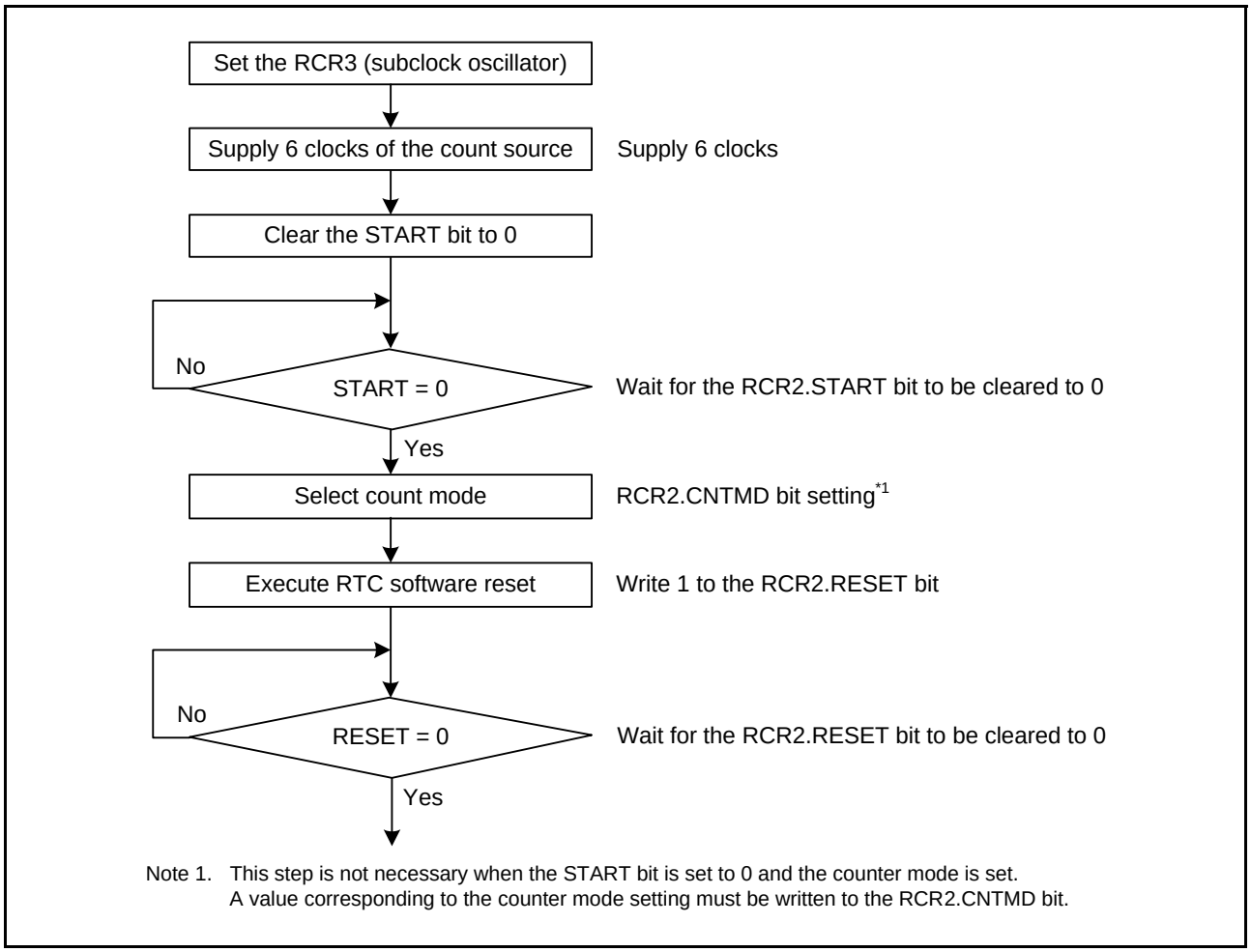


Figure 25.2 Outline of Initial Settings after Power-On

25.3.2 Clock and count mode Setting Procedure

Figure 25.3 shows how to set the clock and the count mode.



Note 1. This step is not necessary when the START bit is set to 0 and the counter mode is set.
A value corresponding to the counter mode setting must be written to the RCR2.CNTMD bit.

Figure 25.3 Clock and the count mode Setting Procedure

25.3.3 Setting the Time

Figure 25.4 shows how to set the time.

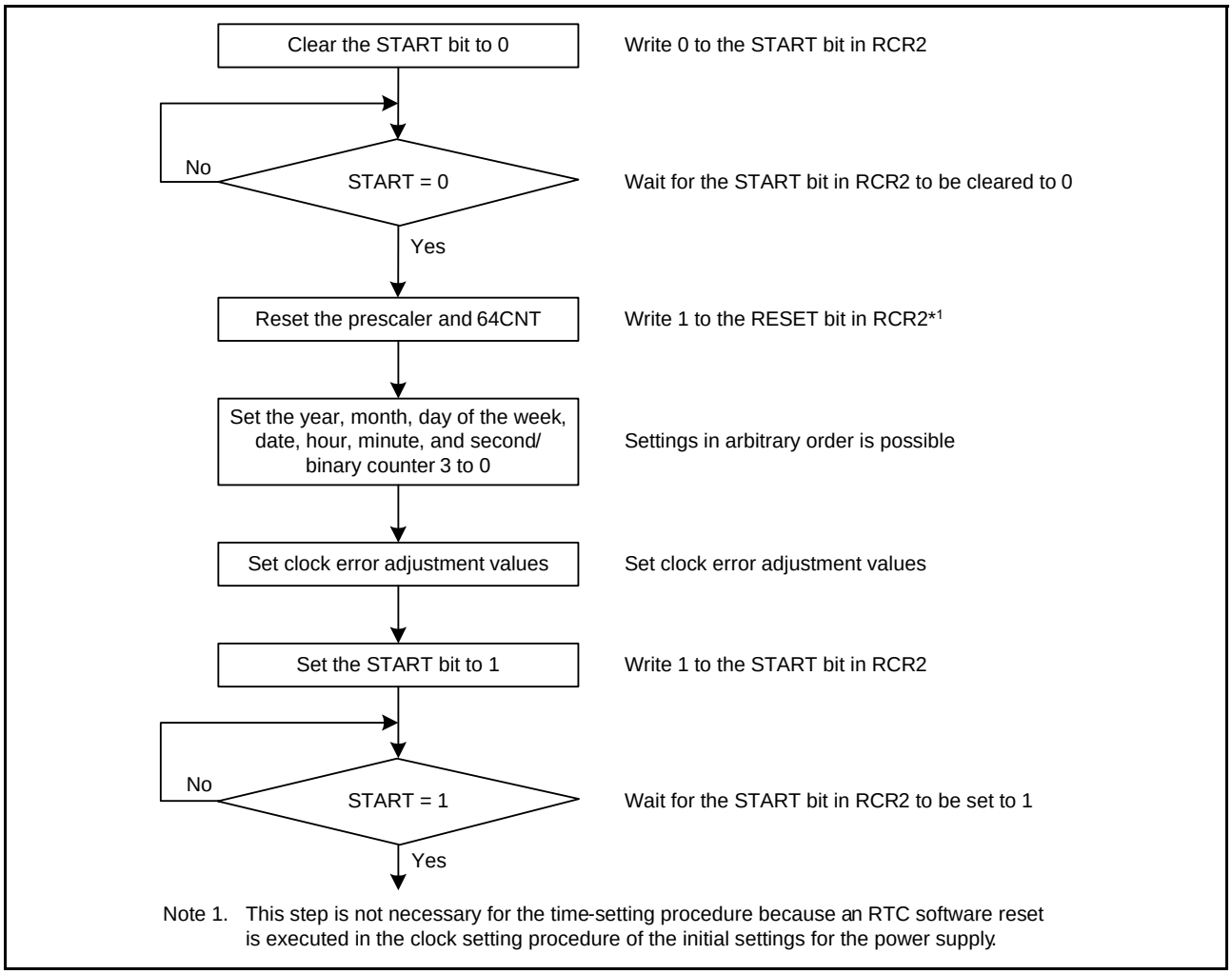


Figure 25.4 Setting the Time

25.3.4 30-Second Adjustment

Figure 25.5 shows how to execute 30-second adjustment. The 30-second adjustment function is only usable in calendar count mode.

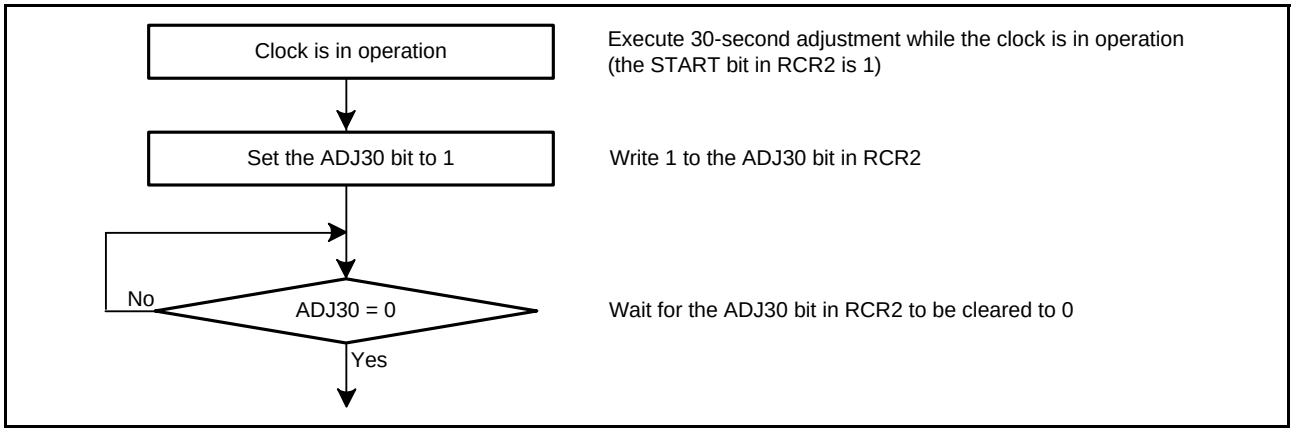


Figure 25.5 30-Second Adjustment

25.3.5 Reading 64-Hz Counter and Time

Figure 25.6 shows how to read the 64-Hz counter and time.

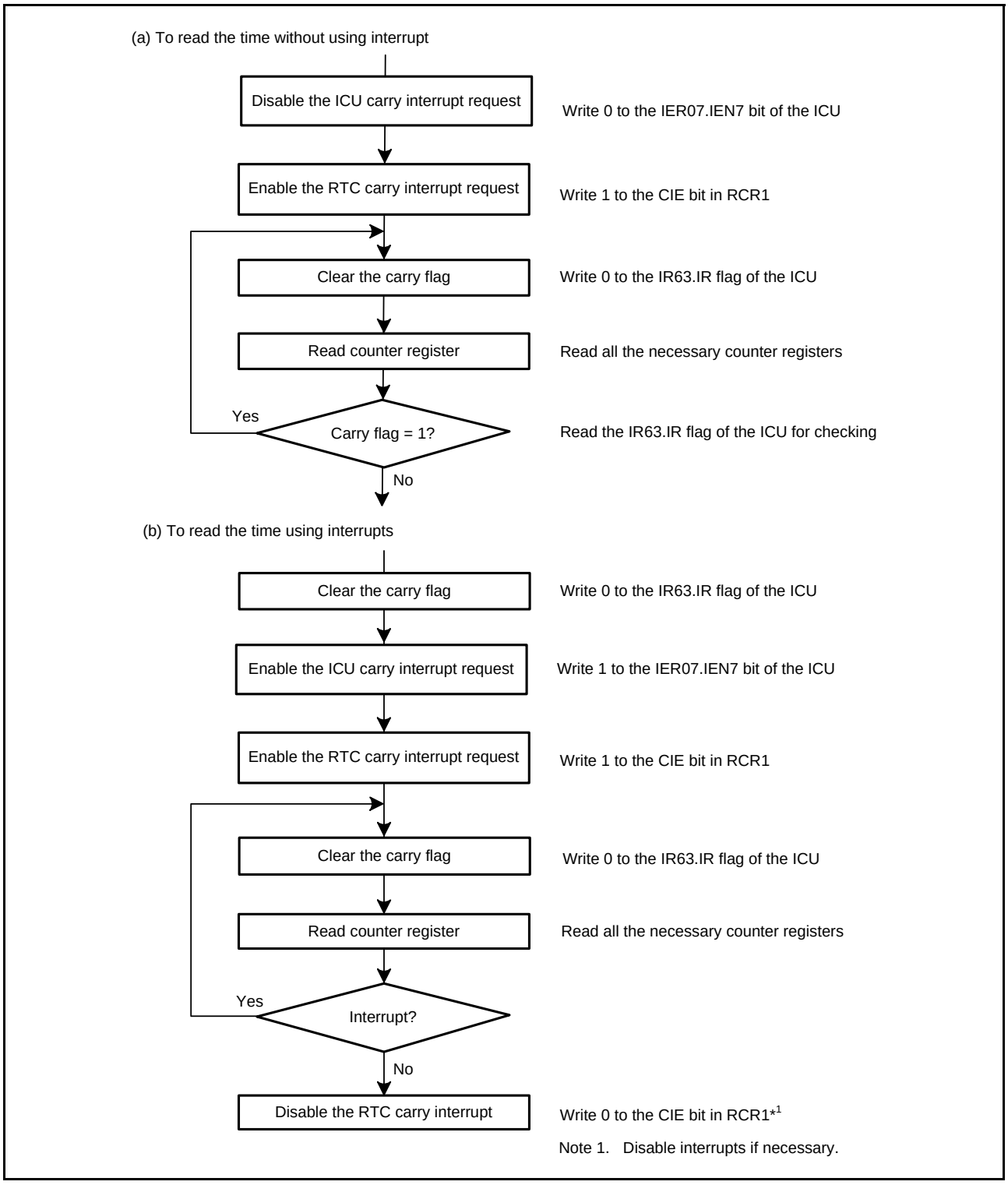


Figure 25.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 25.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

25.3.6 Alarm Function

Figure 25.7 shows how to use the alarm function.

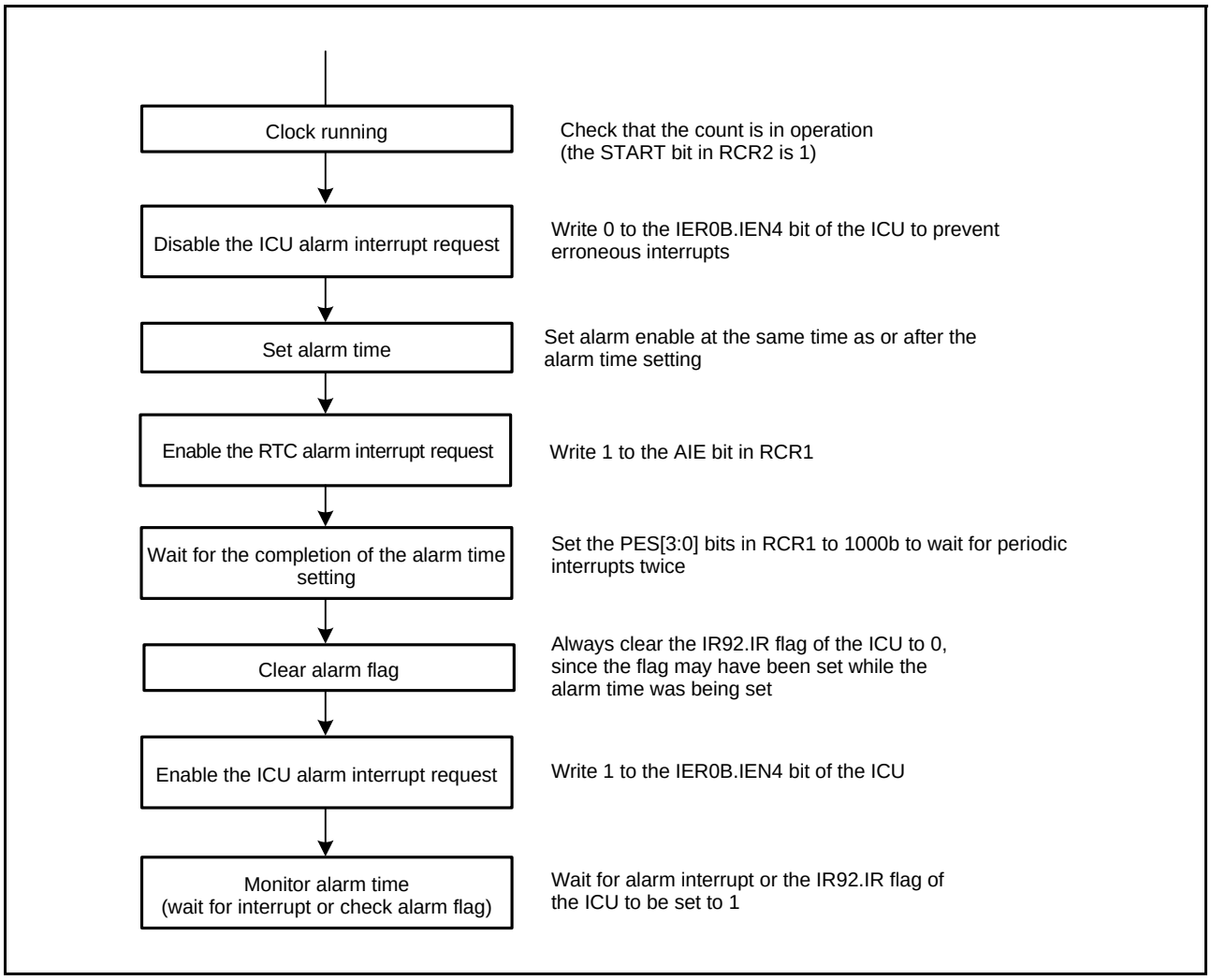


Figure 25.7 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR92.IR flag of the ICU is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the IER0B.IEN4 bit of the ICU, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 clears the IR92.IR flag of the ICU.

When the counter and the alarm time match in a low power consumption state, this LSI returns from the low power consumption state.

25.3.7 Procedure for Disabling Alarm Interrupt

Figure 25.8 shows the procedure for disabling the enabled alarm interrupt request.

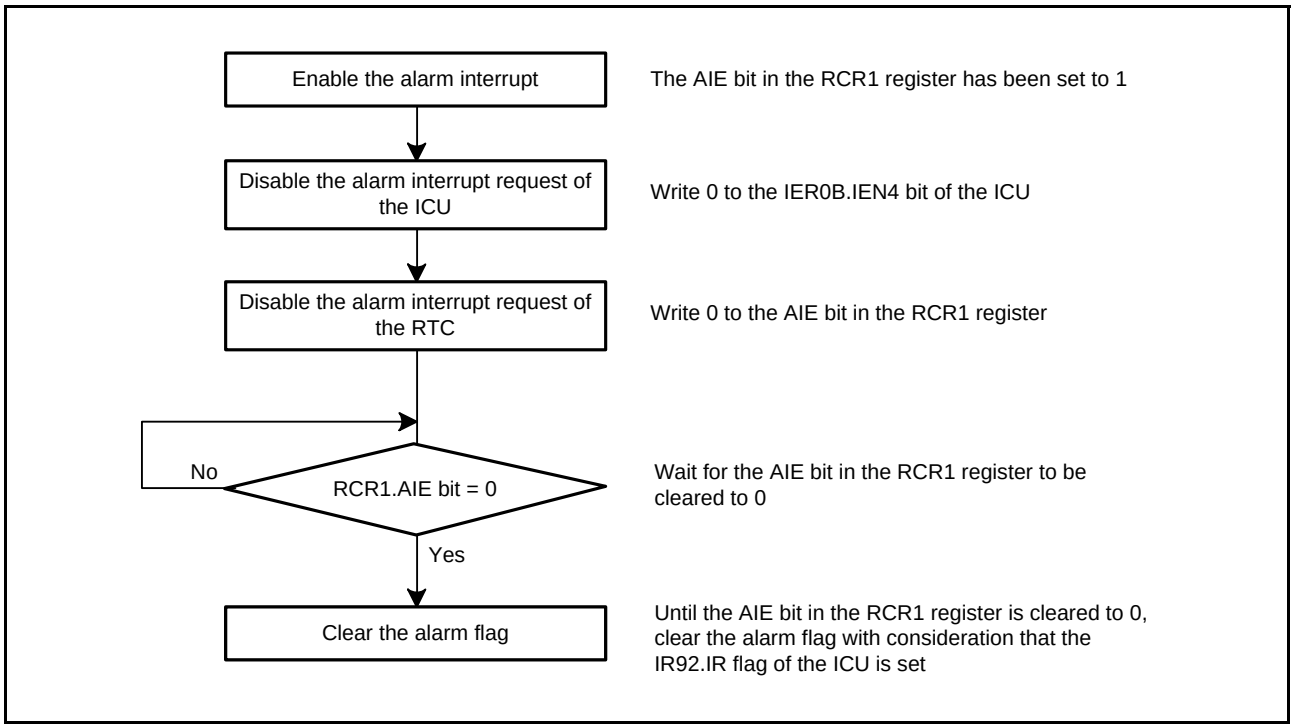


Figure 25.8 Procedure for Disabling Alarm Interrupt Request

25.3.8 Time Error Adjustment Function

The time-error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute one second of operation, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time-error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

25.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the AADJE bit in RCR2 to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the AADJP bit in RCR2 elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, one second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings:(when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (subtraction)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, one second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every ten seconds, so adjustment can take the form of setting the clock forward by 20 cycles every ten seconds.

Register settings:(when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (addition)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as one second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles per 8 seconds.

Register settings: Register settings: (when RCR2.CNTMD = 1)

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (addition)
- RADJ. ADJ[5:0] = 32 (20h)

25.3.8.2 Adjustment by Software

Enable adjustment by software by setting the AADJE bit in RCR2 to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, one second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (subtraction)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

25.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after clearing the PMADJ[1:0] bits in RADJ to 00b (no adjustment).

Changing from adjustment by software to automatic adjustment:

(1) Clear the PMADJ[1:0] bits in RADJ to 00b (no adjustment).
(2) Set the AADJE bit in RCR2 to 1 (enabling automatic adjustment).
(3) Use the AADJP bit in RCR2 to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time-error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Clear the PMADJ[1:0] bits in RADJ to 00b (no adjustment).
(2) Set the AADJE bit in RCR2 to 0 (enabling adjustment by software).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time-error adjustment at the desired time. After that, the time is adjusted every time a value is written to RADJ.

25.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by clearing the PMADJ[1:0] bits in RADJ to 00b (no adjustment).

25.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 25.3 lists interrupt sources for the RTC.

Table 25.3 RTC Interrupt Sources

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to the description of each alarm register).

Since there is a possibility of the interrupt flag being set when the settings of the alarm registers match the clock counters, clear the IR flag in IR92 for the interrupt after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been cleared and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

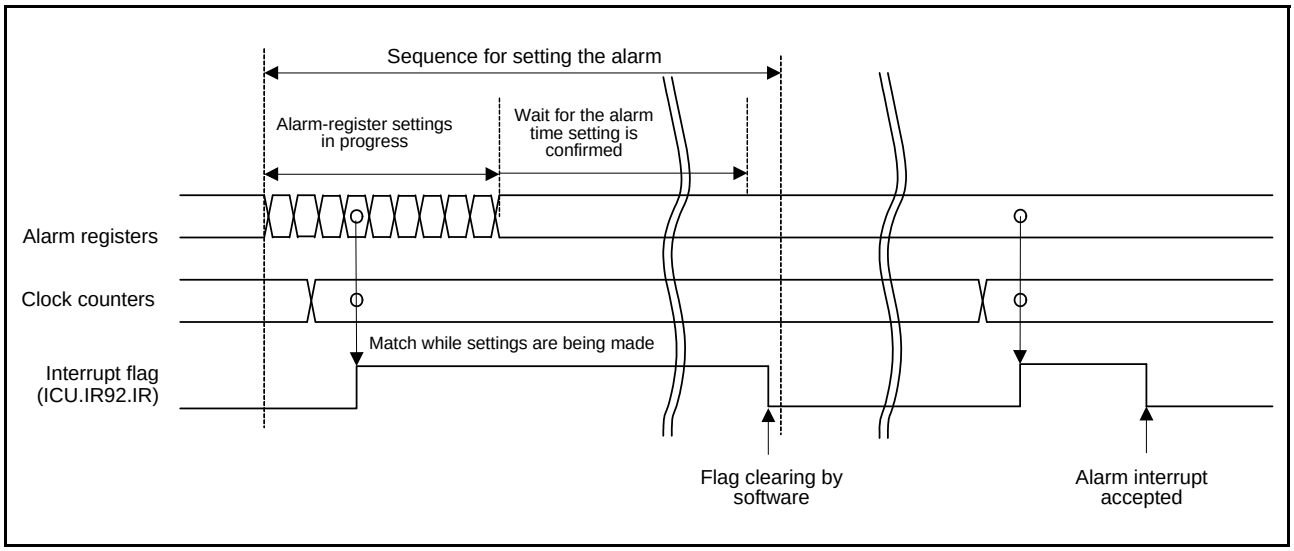


Figure 25.9 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

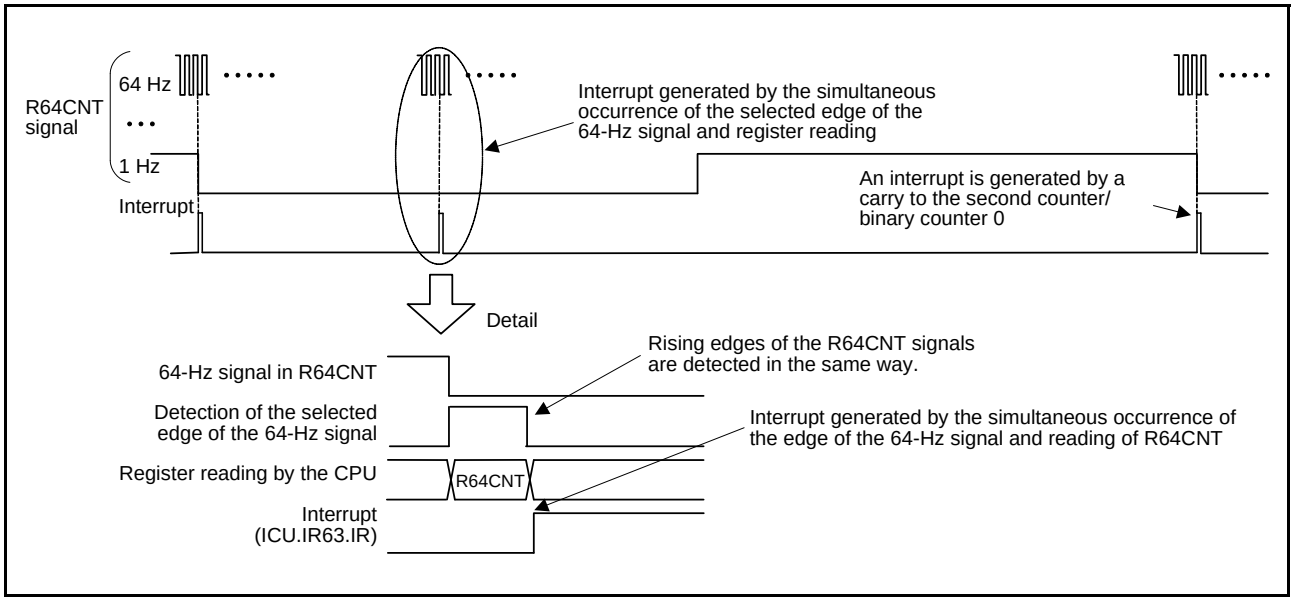


Figure 25.10 Carry Interrupt (CUP) Timing Chart

25.5 Usage Notes

25.5.1 Register Writing during Counting

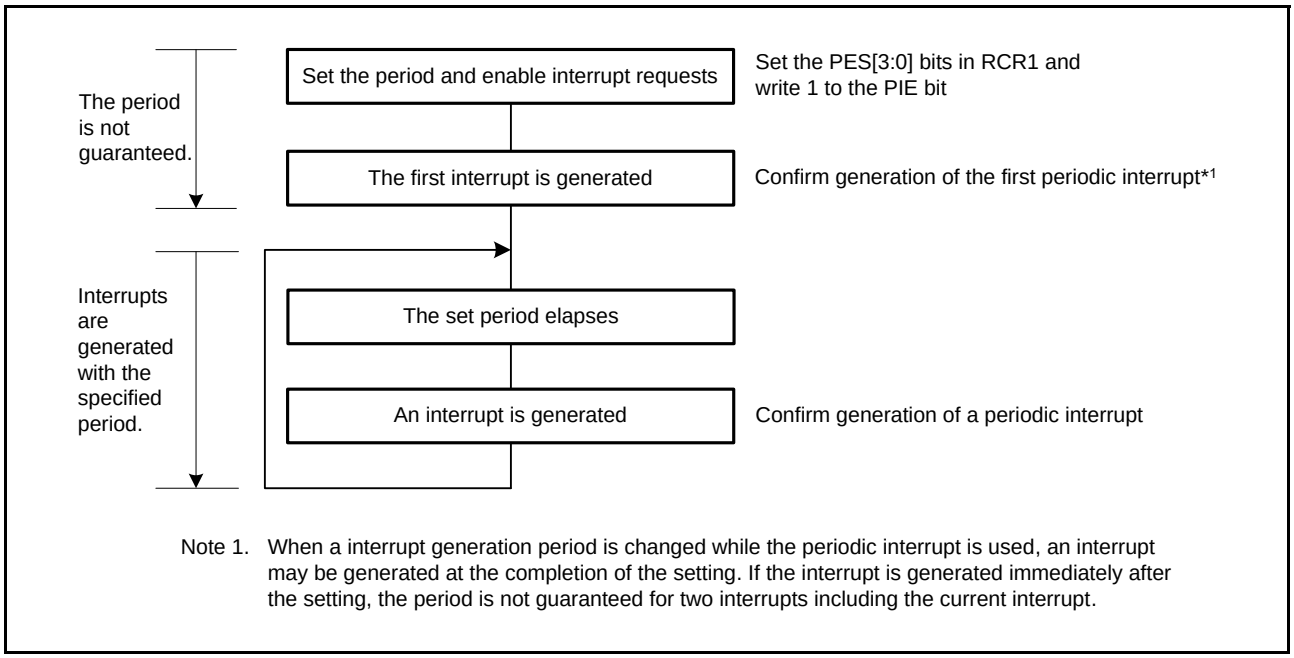
The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

25.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 25.11.
The generation and period of the periodic interrupt can be changed by the setting of the PES[3:0] bits in RCR1. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the PES[3:0] bits in RCR1.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time-error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.



Note 1. When a interrupt generation period is changed while the periodic interrupt is used, an interrupt may be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

Figure 25.11 Using Periodic Interrupt Function

25.5.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time-error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

25.5.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

25.5.5 Points for Caution when Writing to and Reading from Registers

- When reading a counter register such as the second counter/binary counter 0 after having written to the counter register, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.
- When reading the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, or RCR3 register after having written to the registers, perform three dummy read operations to ensure that the written value has been reflected in the register.
- When reading the RCR1.CIE, RTCOS, and RCR2.RTCOE bits after having written to the register, the written value can be read.
- To read the value from the timer counter after return from a reset, or period in software standby mode, wait for 1/128 second while the clock is operating (the RCR2.START bit = 1).

25.5.6 Changing Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section section 25.3.1, Outline of Initial Settings of Registers after Power-On.

26. Independent Watchdog Timer (IWDTa)

The independent watchdog timer (IWDT) detects programs being out of control.
The IWDT has a 14-bit down-counter, and can be set up so that the chip is reset by a reset output when counting down from the initial value causes an underflow of the counter. Alternatively, generation of an interrupt request is selectable when the counter underflows. The initial value for counting can be restored to the down-counter by refreshing its value. The interval over which refreshing is possible can also be selected. Refreshing the counter during this interval will restore its initial value for counting, while attempting to refresh the counter beyond this interval leads to the output of a reset or interrupt request. The refresh interval can be adjusted and used to detect the program entering runaway conditions. The IWDT stops counting after an underflow or an attempt at refreshing the counter beyond the allowed interval. Counting is restarted by refreshing the counter when the IWDT is in register start mode. When the IWDT is in auto-start mode, counting is restarted automatically after output of the reset or interrupt request.

26.1 Overview

The IWDT 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 the IWDT (writing to the register).
In auto-start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in option function select register 0 (OFS0) before release from the reset state.
In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the respective registers before the counter is started by refreshing after release from the reset state.
Set the IWDT start mode select bit (OFS0.IWDTSTRT) to select auto-start mode or register start mode.
When auto-start mode is selected (OFS0.IWDTSTRT = 0), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) settings are disabled and option function select register 0 (OFS0) settings are enabled.
When register start mode is selected (OFS0.IWDTSTRT = 1), option function select register 0 (OFS0) settings are ignored and the IWDTCR, IWDTRCR, and IWDTCSTPR settings take effect.
Specifications of the IWDT are listed in Table 26.1.

Table 26.1 Specifications of IWDT

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 by refreshing the IWDTRR register (writing 00h and then FFh) (register start mode)
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 Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset-output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Interrupt request output sources	• A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter • 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.
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 time-out period of the watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency division ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the time-out period of the watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or all-module clock stop mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

To use the IWDT, two clocks (peripheral clock (PCLK) and IWDT-dedicated clock (IWDTCLK)) should be supplied so that the IWDT works while the peripheral clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit down-counter and control circuits operate with IWDTCLK.
Signal lines between the blocks operating with the peripheral clock and IWDT-dedicated clock are connected through synchronization circuits.
Figure 26.1 is a block diagram of the IWDT.

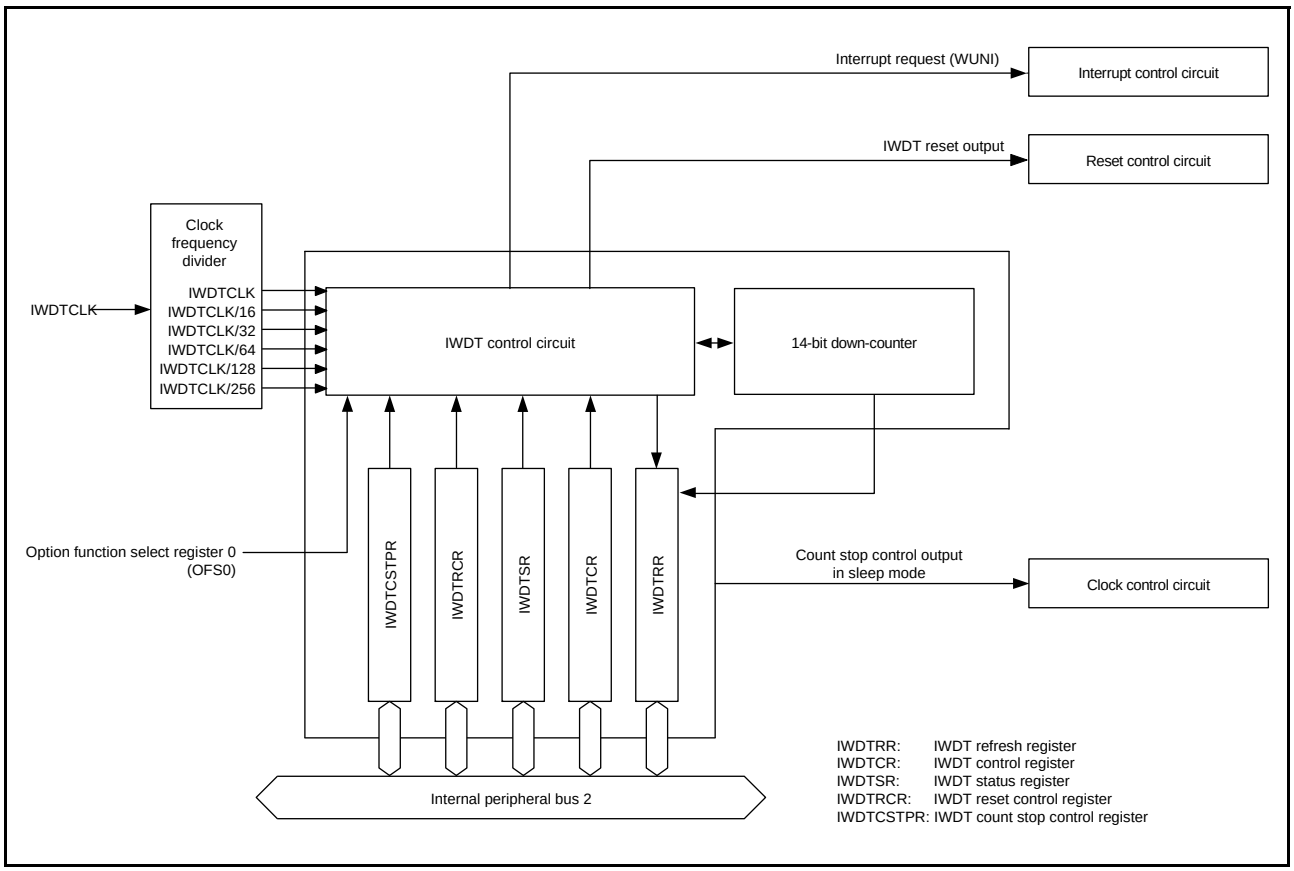
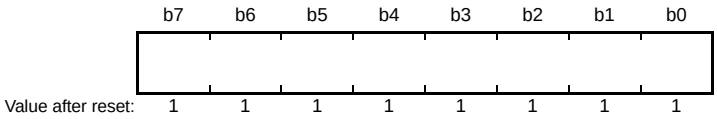


Figure 26.1 Block Diagram of IWDT

26.2 Register Descriptions

26.2.1 IWDT Refresh Register (IWDTRR)

Address(es): 0008 8030h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

IWDTRR refreshes the down-counter of the IWDT.
The down-counter of the IWDT is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by the IWDT time-out 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 time-out period selection (TOPS[1:0]) bits 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 26.3.3, Refresh Operation.

26.2.2 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]	Time-Out 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. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	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 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. Writing to these bits has no effect.	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. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 26.3.2, 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 26.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Time-Out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) from among 1,024, 4,096, 8,192, or 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 IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the time-out period, and the number of IWDTCLK cycles are listed in Table 26.2.

Table 26.2 Settings and Time-Out Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Division Ratio	Time-Out 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 Selection)

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 1,024 and 4,194, 304 cycles of the IWDTCLK clock can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Selection)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-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 26.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 26.3 Relationship between Time-Out Period and Window Start and End Counter Values

TOPS[1:0] Bits		Time-Out 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 Selection)

These bits select 100%, 75%, 50%, and 25% of the count period for the window start position for the down-counter (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 26.2 shows the relationship between 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%.

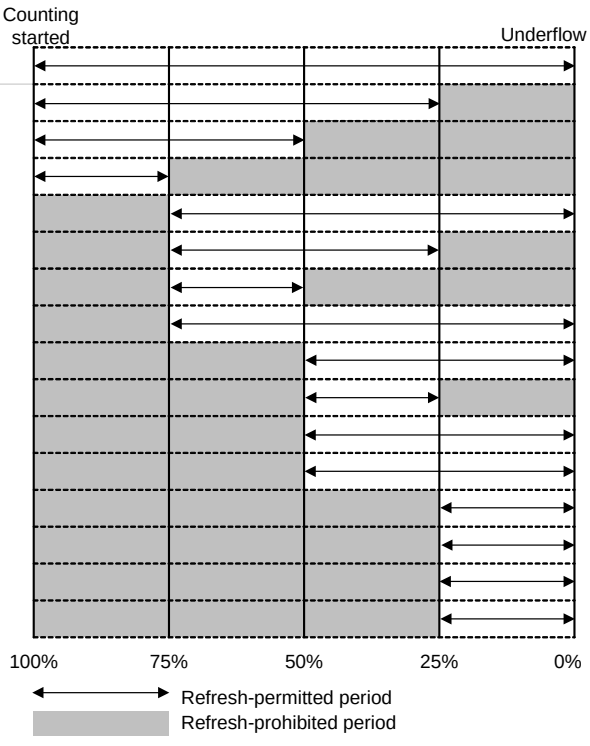


Figure 26.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

26.2.3 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]	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.

IWDTSR is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

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 bit can be read 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 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.

26.2.4 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 Selection	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 26.3.2, 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) enabled. The bit setting mode to the IWDTRCR register can also be made in option function select register 0. For details, refer to section 26.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

26.2.5 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, or all-module clock stop mode	R/W

IWDTCSTPR controls whether to stop the IWDT down-counter at transitions to low power consumption modes. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 26.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the settings in the IWDTCSTPR register are ignored, and the settings in option function select register 0 (OFS0) take effect. The bit setting mode to the IWDTCSTPR register can also be made in option function select register 0 (OFS0). For details, refer to section 26.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or all-module clock stop mode.

26.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 26.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

26.3 Operation

26.3.1 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.

26.3.1.1 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 canceling from the reset, set the clock division ratio, window start and end positions, and time-out period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the down-counter stop control at transitions to low power consumption modes in the IWDTCSTPR register. Then refresh the down-counter to start counting down from the value selected by setting the time-out period selection (IWDTCR.TOPS[1:0]) bits.
There after, 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 down-counter underflows because of 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 IWDT outputs a reset signal or a non-maskable interrupt request (WUNI).
Select reset output or interrupt request output by setting the reset interrupt request selection (IWDTRCR.RSTIRQS) bit.

Figure 26.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

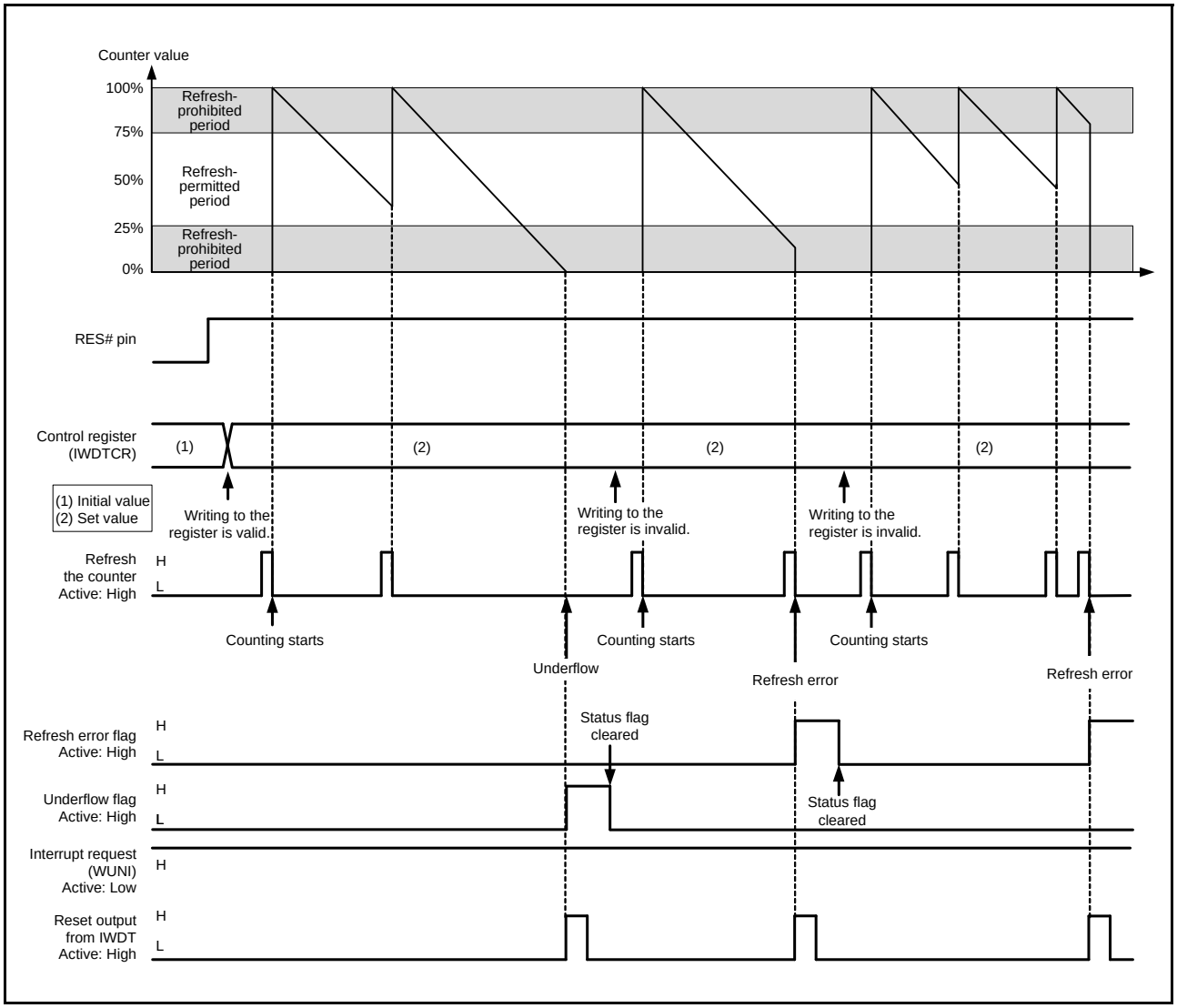


Figure 26.3 Operation Example in Register Start Mode

26.3.1.2 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, time-out period, reset output or interrupt request output, and down-counter stop control at transitions to low power consumption modes should be specified in option function select register 0 (OFS0). When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the IWDT time-out period select (OFS0.IWDTTOPS[1:0]) bits.
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set when the counter is refreshed and counting down continues. The IWDT 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 IWDT outputs the reset signal or non-maskable interrupt request (WUNI). The reset output or interrupt request output can be selected through the IWDT reset interrupt request select (OFS0.IWDTRSTIRQS) bit.

Figure 26.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

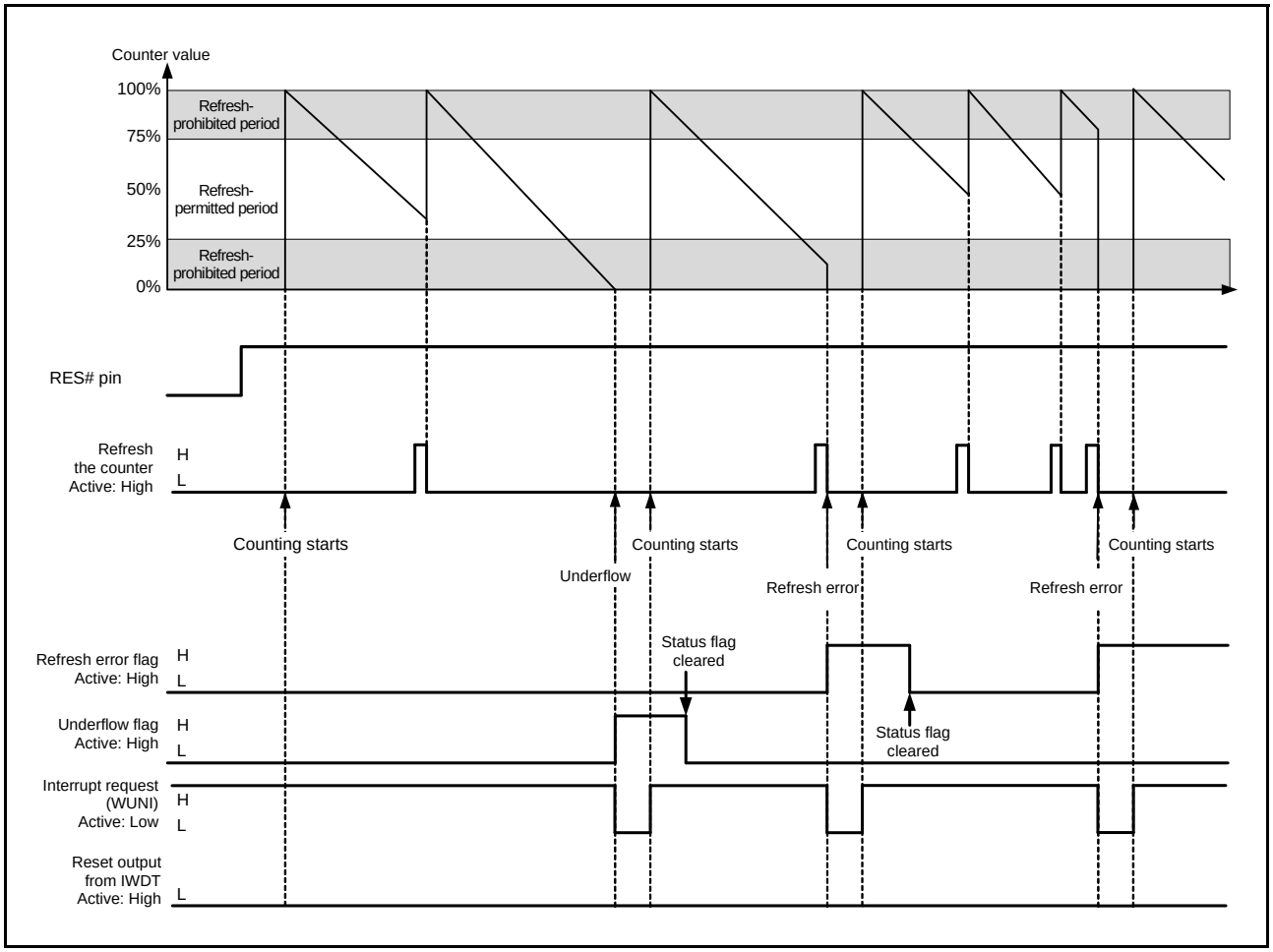


Figure 26.4 Operation Example in Auto-Start Mode

26.3.2 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 26.5 sows control waveforms produced in response to writing to the IWDTCR.

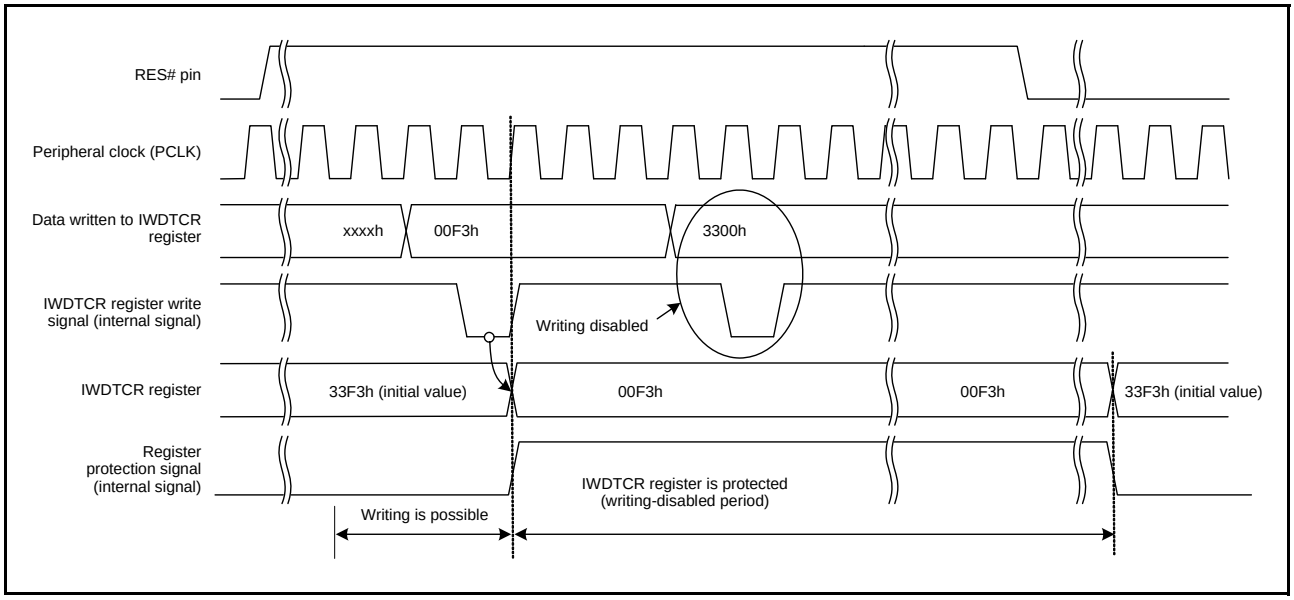


Figure 26.5 Control Waveforms Produced in Response to Writing to the IWDTCR

26.3.3 Refresh Operation

The down-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 down-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 (n-th 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 (n-th 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 down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (the clock division ratio selection (IWDTCR.CKS[3:0]) bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-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 26.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

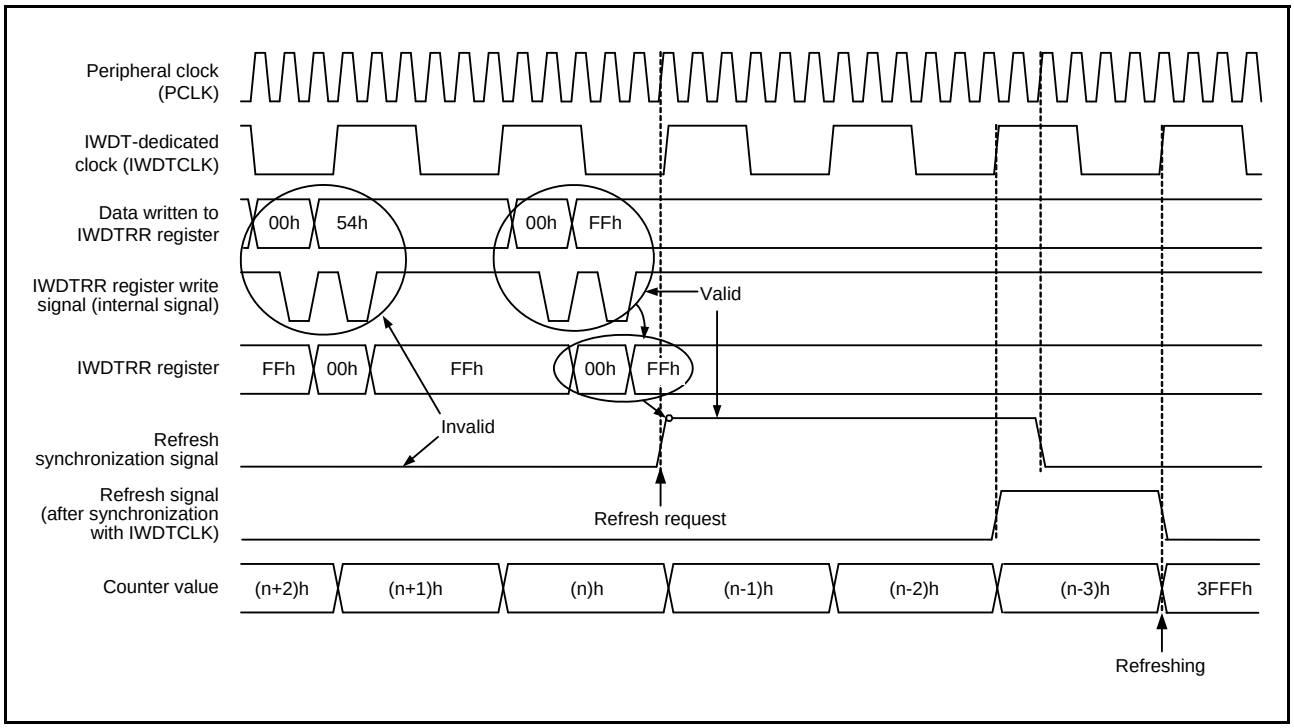


Figure 26.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

26.3.4 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.

26.3.5 Reset Output

When the reset interrupt request selection (IWDTRCR.RSTIRQS) bit is set to 1 in register start mode or when the IWDT reset interrupt request select (OFS0.IWDTRSTIRQS) bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits cleared to 0) and kept in that state after assertion of the reset signal. After the reset is canceled 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.

26.3.6 Interrupt Source

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the IWDT reset interrupt request select (OFS0.IWDTRSTIRQS) bit in option function select register 0 (OFS0) is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

Table 26.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

26.3.7 Reading the Down-Counter Value

As the down-counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLK) and stores it in the down-counter value (IWDTSR.CNTVAL[13:0]) bits of the IWDT status register. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the down-counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual down-counter value by a value of one count.

Figure 26.7 shows the processing for reading the IWDT down-counter value when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

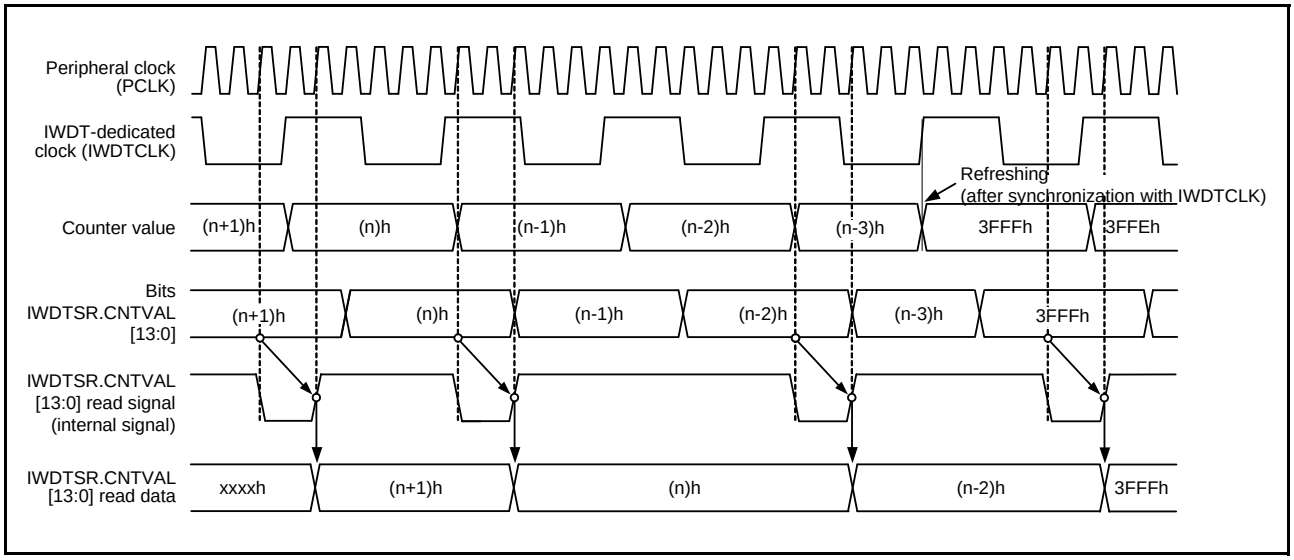


Figure 26.7 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

26.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 26.5 lists the correspondence between option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function. Control can be switched between option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 (OFS0).
Note that option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.
For details on option function select register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

Table 26.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Effective in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Effective in Register Start Mode) OFS0.IWDTSTRT = 1
Down-counter	Time-out 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

26.4 Usage Notes

26.4.1 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.

27. Serial Communications Interface (SCIe, SCIf)

The RX220 Group has five independent serial communications interface (SCI) channels. The SCI is configured as SCIe module (SCI1 5, 6 and 9) and SCIf module (SCI12).
The SCIe (SCI1, 5, 6 and 9) 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). Single-master operation as an simple I^2C bus interface and simple SPI interfaces are also supported.
As well as the functions of the SCIe module, the SCIf module (SCI12) supports an extended serial protocol with a structure formed from Start Frames and Information Frames.

27.1 Overview

Table 27.1 lists the specifications of the SCIe module, Table 27.2 lists the specifications of the SCIf module, and Table 27.3 lists the specifications of the individual SCI channels.
Figure 27.1 is a block diagram depicting SCI1 and 9. Figure 27.2 is a block diagram depicting SCI5 and SCI6. Figure 27.3 is a block diagram depicting SCI12 (the SCIf module).

Table 27.1 Specifications of SCIe (1/2)

Item		Description
Serial communications modes		• Asynchronous operation • Clock synchronous operation • Smart card interface • Simple I^2C bus • Simple SPI bus
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
I/O pins		See Table 27.4 to Table 27.6
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, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Start bit detection	Low level or falling edge is selectable.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR (SCI5 and 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 errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.

Table 27.1 Specifications of SCIe (2/2)

Item		Description
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I2C mode	Transfer format	I2C bus format (MSB-first transfer only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 384 kbps
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	Eight bits
	Detection of errors	Overrun errors
	SS input pin function	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense are selectable.
Event link function (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 I2C mode, only MSB-first is available.

Table 27.2 Specifications of SCIf (1/2)

Item		Description
Serial communications modes		• Asynchronous operation • Clock synchronous operation • Smart card interface • Simple I2C bus • Simple SPI bus
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
Input/output pins		See Table 27.4 to Table 27.7.
Data transfer		Selectable as LSB-first or MSB-first transfer*1
Interrupt sources		Transmit-end, transmit-data-empty, receive-data-full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
Power consumption reduction function		Module stop state can be set.
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Start bit detection	Low level or falling edge is selectable.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.

Table 27.2 Specifications of SCIf (2/2)

Item		Description
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I2C mode	Transfer format	I2C bus format (MSB-first transfer only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 384 kbps
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun errors
	SS input pin function	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense 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 RXDX12 • 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 I2C mode, only MSB-first is available.

Table 27.3 List of Functions of SCI Channels

Item	SCI1 and 9	SCI5	SCI6	SCI12
Asynchronous mode	○	○	○	○
Clock synchronous mode	○	○	○	○
Smart card interface mode	○	○	○	○
Simple I2C mode	○	○	○	○
Simple SPI bus	○	○	○	○
Extended serial mode	—	—	—	○
TMR clock input	—	○	○	○
Event link function	—	○	—	—

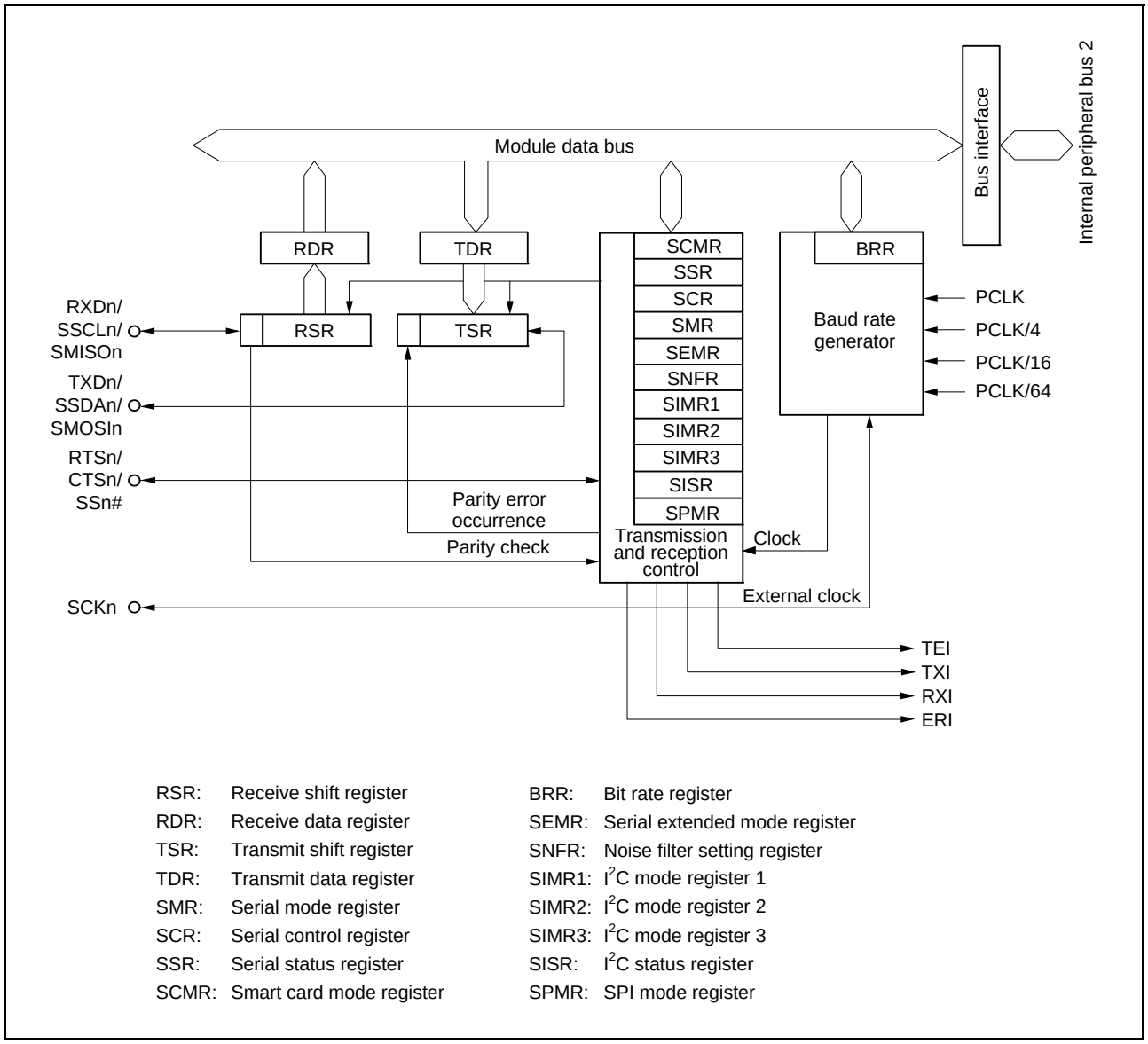
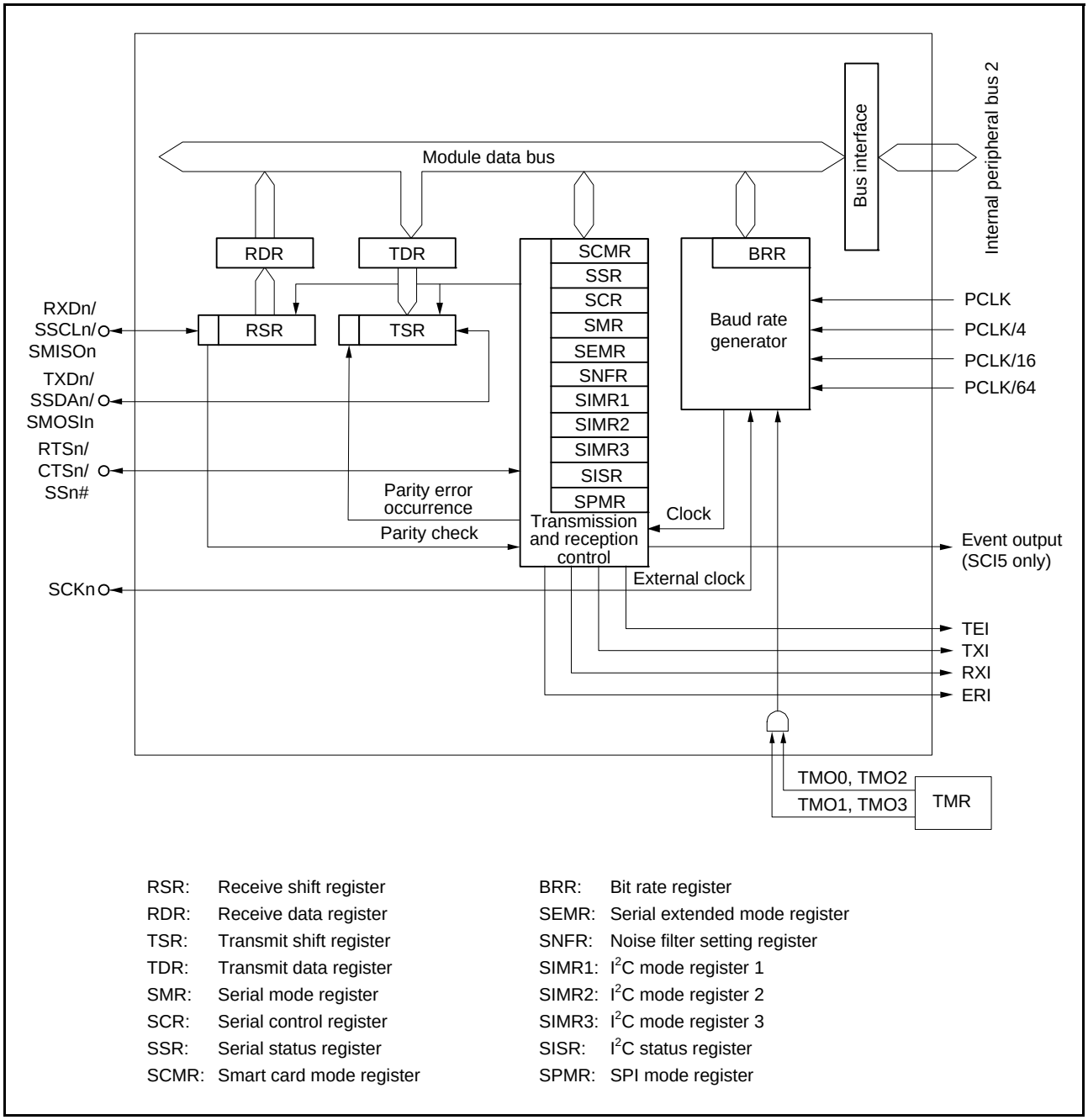


Figure 27.1 Block Diagram of SCI1 and 9



RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR: Serial control register
SSR: Serial status register
SCMR: Smart card mode register
BRR: Bit rate register
SEMR: Serial extended mode register
SNFR: Noise filter setting register
SIMR1: I^2C mode register 1
SIMR2: I^2C mode register 2
SIMR3: I^2C mode register 3
SISR: I^2C status register
SPMR: SPI mode register

Figure 27.2 Block Diagram of SCI5 and SCI6

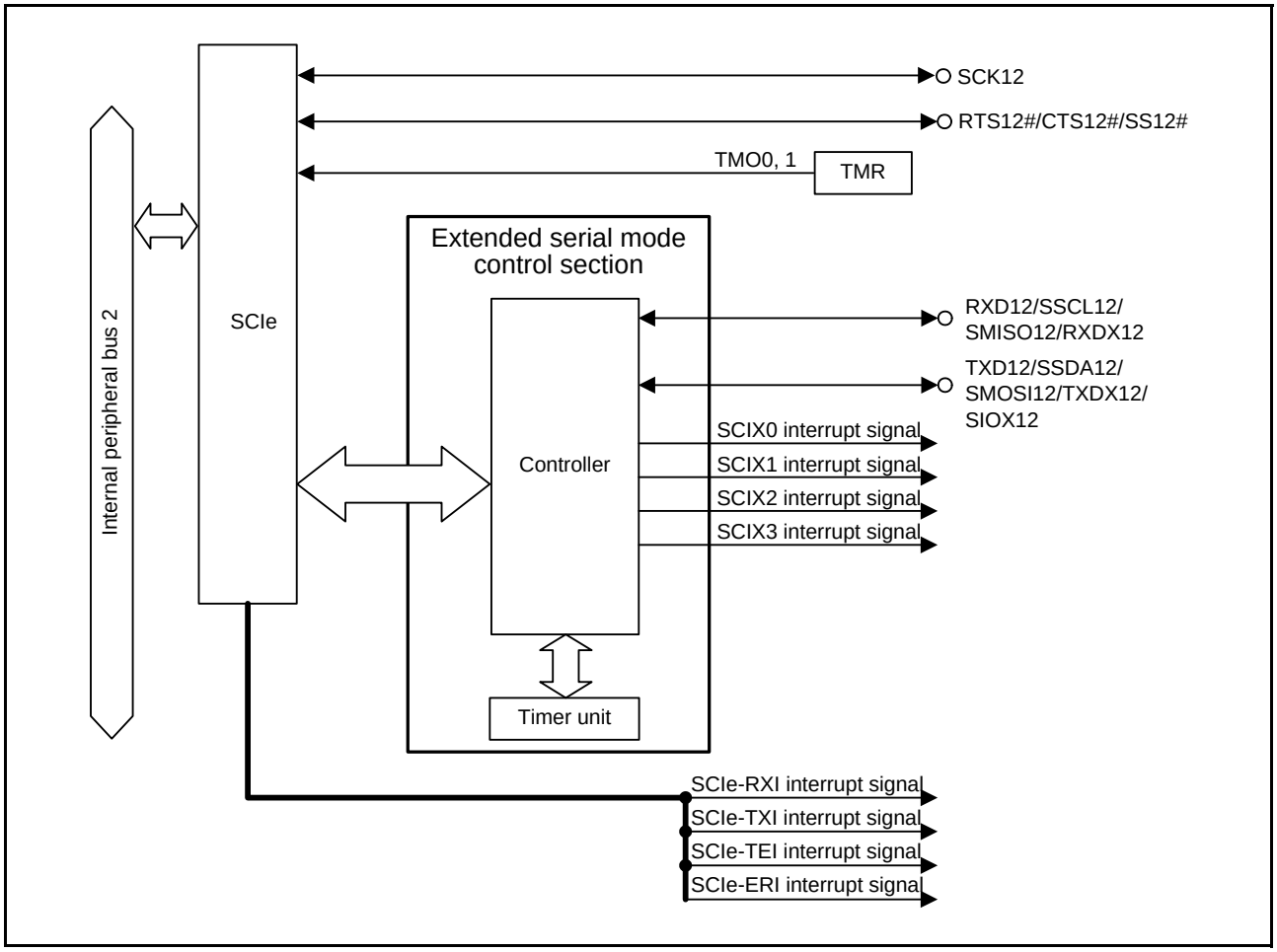


Figure 27.3 Block Diagram of SCI12 (SCIf)

Table 27.4 to Table 27.7 list the pin configuration of the SCIs for the individual modes.

Table 27.4 Input and Output Pins of the SCIs (Asynchronous/Clock Synchronous Modes)

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6	Input	SCI6 receive data input
	TXD6	Output	SCI6 transmit data output
	CTS6#/RTS6#	I/O	SCI6 transfer start control input/output
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
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 27.5 Pin Configuration of SCI (Simple I^2C Mode)

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output
SCI6	SSCL6	I/O	SCI6 I^2C clock input/output
	SSDA6	I/O	SCI6 I^2C data input/output
SCI9	SSCL9	I/O	SCI9 I^2C clock input/output
	SSDA9	I/O	SCI9 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 27.6 Pin Configuration of SCI (Simple SPI Mode)

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI6	SCK6	I/O	SCI6 clock input/output
	SMISO6	I/O	SCI6 slave transmit data input/output
	SMOSI6	I/O	SCI6 master transmit data input/output
	SS6#	Input	SCI6 chip select input
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
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 27.7 Pin Configuration of SCI (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

27.2 Register Descriptions

27.2.1 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 transferred to RDR automatically.
RSR cannot be directly accessed by the CPU.

27.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI9.RDR 0008 A125h, 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 the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. This allows RSR to 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.

27.2.3 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, SCI9.TDR 0008 A123h, 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 data for transmission to TDR once after each instance of the transmit data empty interrupt (TXI).

27.2.4 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.

27.2.5 Serial Mode Register (SMR)

Note: • Some bits in SMR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI9.SMR 0008 A120h, 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 in asynchronous mode) 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	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (see section 27.2.9, Bit Rate Register (BRR)).
Note 2. In clock synchronous 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, see section 27.2.9, 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 clock synchronous mode, a fixed data length of 8 bits is used.

CM Bit (Communications Mode)

Selects asynchronous or clock synchronous mode.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI9.SMR 0008 A120h, 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 BCP2 bit in SCMR. Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR: BCP2 b3 b2 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 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 data for transmission, 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 (see section 27.2.9, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (see section 27.2.9, 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, see section 27.2.9, 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 BCP2 bit in SCMR.

For details, see section 27.6.4, Receive Data Sampling Timing and Reception Margin.

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, see section 27.6.2, 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, see section 27.6.3, 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, see section 27.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 27.6.8, Clock Output Control.

27.2.6 Serial Control Register (SCR)

Note: • Some bits in SCR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI9.SCR 0008 A122h, 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 SCI1and 9 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock 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 in accord with the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock • When an external clock is used, 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. (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. A 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 in 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 clearing the TEIE bit to 0.

In simple I^2C mode (when the IICM bit in SIMR1 is 1), 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, see section 27.4, 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 clearing 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 clearing 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 clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI9.SCR 0008 A122h, 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 GM in SMR = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.) 0 1: Clock output 1 x: (Setting prohibited) • When GM in SMR = 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. A 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 in TE and RE. While the SMR.CM bit is 0, writing is enabled under any condition.

For details on interrupt requests, see section 27.11, 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, see section 27.6.8, 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 clearing 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 clearing 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 clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

27.2.7 Serial Status Register (SSR)

Note: • Some bits in SSR have different functions in serial communications interface mode and smart card interface mode.

(1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI9.SSR 0008 A124h, 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.

MPBT Bit (Multi-Processor Bit Transfer)

Sets the value of the multi-processor bit for adding to the transmission frame.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (disabling serial transmission operations)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- The TDR is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When data for transmission are written to the TDR while the SCR.TE is 1

When the TEND flag is cleared by writing transmit data to the TDR register, perform a dummy read of the SSR register in the following order:

(1) Write transmit data to the TDR register.
(2) Read the SSR register and write that value to a general register.
(3) Execute some operations using the read value.

PER Bit (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 (after writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Bit (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 (after writing 0 to it, read the FER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

ORER Bit (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 (after writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

(2) Smart Card Interface Mode (SMIF in SCMR = 1)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI9.SSR 0008 A124h, 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.

MPBT Bit (Multi-Processor Bit Transfer)

This bit should be set to 0 in smart card interface mode.

MPB Bit (Multi-Processor)

This bit is not used in smart card interface mode. It should be set to 0.

TEND Flag (Transmission 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 SCR.TE bit = 0 (disabling serial transmission operations)
  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 one 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 data for transmission are written to the TDR while the SCR.TE is 1

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 (After writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (which indicates that 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

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 (After writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

27.2.8 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI9.SCMR 0008 A126h, 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: Serial communications interface 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	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. Setting values in the SCMR.BCP2 bit BCP2 BCP1 BCP0 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial Value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
Note 2. S is the value of S in BRR (see section 27.2.9, Bit Rate Register (BRR)).

SCMR selects smart card interface mode and its format.

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, asynchronous or clock synchronous 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.

SDIR Bit (Transmitted/Received Data Transfer Direction)

Selects the serial/parallel conversion format.
This bit can be used in the following modes.

- Asynchronous mode
- Clock-synchronous mode
- Smart card interface mode
- Multiprocessor mode
- Simple SPI mode

Set this bit to 1 if operation is to be in simple I2C mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

27.2.9 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI9.BRR 0008 A121h, 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 27.8 shows the relationships 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 at all times, but it can be written to only when the TE and RE bits in SCR are 0.

Table 27.8 Relationships between N Setting in BRR and Bit Rate B

Mode	ABCS Bit in SEMR	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 baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the SMR setting listed in the following table.
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 27.9 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	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 27.10 Clock Source Settings

SMR Setting		
CKS[1:0] Bits	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 27.11 Base Clock Settings in Smart Card Interface Mode

SCMR Setting	SMR Setting	Base Clock Cycles for	
BCP2 Bit	BCP[1:0] Bits	One-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 27.12 lists sample N settings in BRR in normal asynchronous mode. Table 27.13 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 27.15. Examples of BRR (N) settings in smart card interface mode are listed in Table 27.17. Examples of BRR (N) settings in simple I^2C mode are listed in Table 27.19. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 27.6.4, Receive Data Sampling Timing and Reception Margin. Table 27.14 and Table 27.16 list the maximum bit rates with external clock input.

When 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 is two times that of listed in Table 27.12.

Table 27.12 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		
	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02
150	3	64	0.16	3	80	0.47
300	2	129	0.16	2	162	–0.15
600	2	64	0.16	2	80	0.47
1200	1	129	0.16	1	162	–0.15
2400	1	64	0.16	1	80	0.47
4800	0	129	0.16	0	162	–0.15
9600	0	64	0.16	0	80	0.47
19200	0	32	–1.36	0	40	–0.76
31250	0	19	0.00	0	24	0.00
38400	0	15	1.73	0	19	1.73

Note: • This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 27.13 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

Note: • When the ABCS bit in SEMR is set to 1, the bit rate is two times.

Table 27.14 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

Table 27.15 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Bus Mode)

	Operating Frequency PCLK (MHz)									
	8		10		16		20		25	
Bit Rate (bps)	n	N	n	N	n	N	n	N	n	N
110										
250	3	124	—	—	3	249				
500	2	249	—	—	3	124	—	—		
1 k	2	124	—	—	2	249	—	—	3	97
2.5 k	1	199	1	249	2	99	2	124	2	155
5 k	1	99	1	124	1	199	1	249	2	77
10 k	0	199	0	249	1	99	1	124	1	155
25 k	0	79	0	99	0	159	0	199	0	249
50 k	0	39	0	49	0	79	0	99	0	124
100 k	0	19	0	24	0	39	0	49	0	62
250 k	0	7	0	9	0	15	0	19	0	24
500 k	0	3	0	4	0	7	0	9	—	—
1 M	0	1			0	3	0	4	—	—
2 M	0	0*1	—	—	0	1	—	—	—	—
2.5 M			0	0*1			0	1	—	—
4 M					0	0*1	—	—	—	—
5 M							0	0*1	—	—
6.25 M									0	0*1
7.5 M										

Space: Setting prohibited.

—: Can be set, but there will be error.

Note 1. The bit rate can be set only when the serial transfer clock is output, but 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 serial transfer clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 27.16 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7

Table 27.17 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

Table 27.18 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0

Table 27.19 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

Table 27.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Min. Widths at High/Low Level for SCL (µs)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width
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

Min. Widths at High/Low Level for SCL (µs)	Operating Frequency PCLK (MHz)		
	25		
	n	N	High/Low Width
10 k	1	19	44.80/51.20
25 k	1	7	17.92/20.48
50 k	1	3	8.96/10.24
100 k	1	1	4.48/5.12
250 k	0	3	2.24/2.56
350 k	0	2	1.68/1.92

27.2.10 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI9.SEMR 0008 A127h, 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: TMR clock input (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	These bits are 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 27.4 shows a setting example when the TMOn output of TMRn (n = 0 to 3) 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.
Clear 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 always be 0.

ABCS Bit (Asynchronous Mode Base Clock Select)

Selects the number of base clock pulses for 1-bit period.

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.

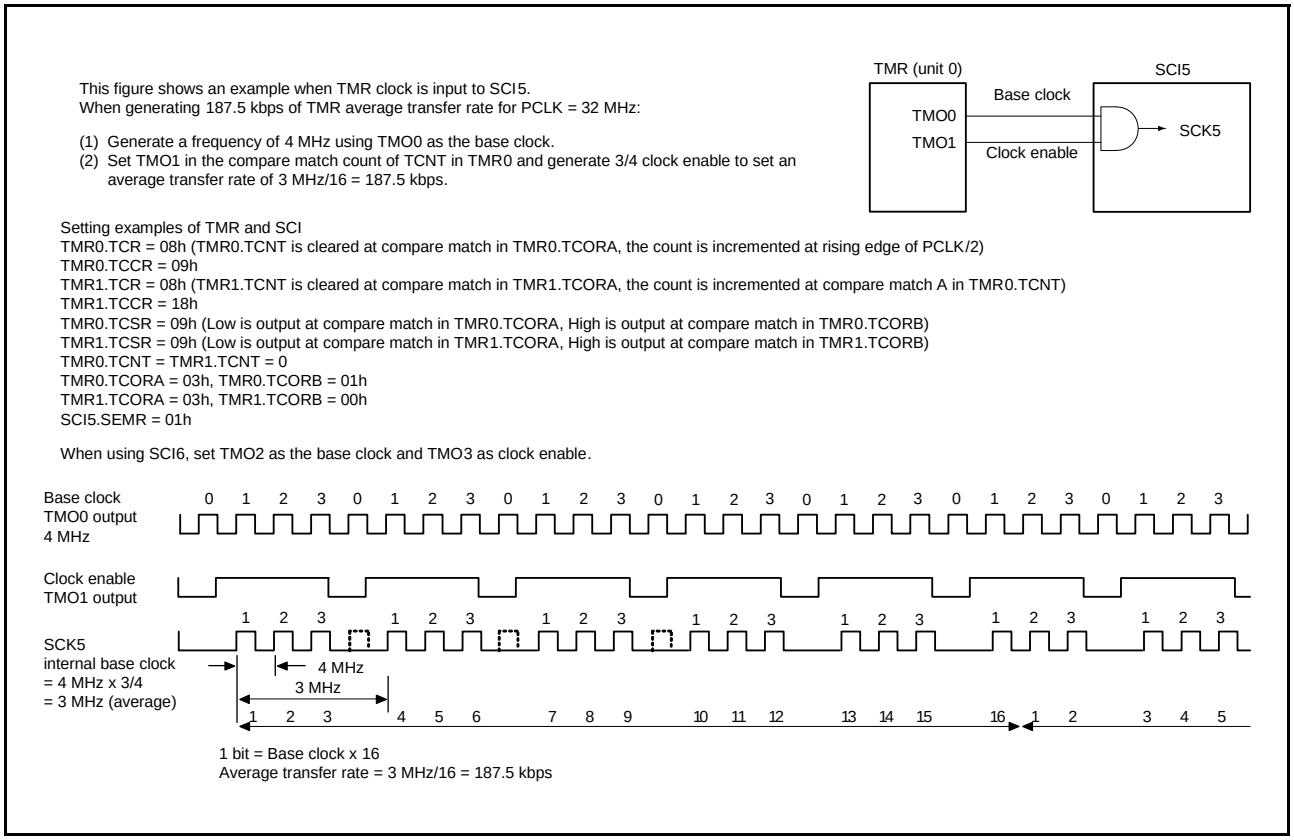


Figure 27.4 Example of Average Transfer Rate Setting when TMR Clock is Input

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

This bit selects the method to detect a start bit during receive operation in asynchronous mode. Operation in a break differs depending on the setting of this bit. Set this bit to 1 to stop receive operation during a break, or to start reception without holding the RXDn pin input at high for one frame period or more after a break ends.

Set this bit to 0 in a mode other than asynchronous mode.

27.2.11 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI9.SNFR 0008 A128h, 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 selected for the on-chip baud rate generator 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 (disabling reception and transmission).

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.

27.2.12 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI9.SIMR1 0008 A129h, 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: Serial interface mode (in asynchronous, 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 Delay Output 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 (disabling reception and transmission).

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.

27.2.13 I²C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, SCI9.SIMR2 0008 A12Ah, 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 (disabling serial reception and transmission).

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 bit 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.

27.2.14 I²C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI9.SIMR3 0008 A12Bh, 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	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2,*3,*4	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2,*3,*4	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: All request generation has been completed.	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 SSDA pin. 1 1: Place the SSDA 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 SSCL pin. 1 1: Place the SSCL 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.

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 execution for the generation of a condition, this bit indicates the state of generation being completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after clearing the IICSTIF flag to 0.

If the TEIE bit in the SCR is enabling interrupt requests, an STI is output on completion of generation of the start, restart, or stop condition when the IICSTIF flag is 1.

[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 zero 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 IICM bit in SIMR1 (when operation is not in simple I^2C mode)
- Writing 0 to the TE bit in SCR

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS and IICSCLS 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 and IICSDAS bits to the same value during normal operations.

27.2.15 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI9.SISR 0008 A12Ch, 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 IICACK flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

27.2.16 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, SCI9.SPMR 0008 A12Dh, 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	SS Pin Function Enable	0: SS pin function is disabled. 1: SS pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS pin function is disabled (RTS output function is enabled). 1: CTS pin 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 (disabling reception and transmission).
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 (SS 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 SS# 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 CTS bits to “enabled” (even if this setting is made, the functions will be disabled).

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 CTS and SSE bits to “enabled” (even if this setting is made, the functions will be disabled).

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 are 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 SS# 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 27.52 for details.
Clear 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 27.52 for details.
Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

27.2.17 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 27.21 Settings of the ESME Bits and Guaranteed Operation by Timer Operation Mode

ESME bit	Timer Mode	Break Field Low Width Judgment 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.

27.2.18 Control Register 0 (CR0)

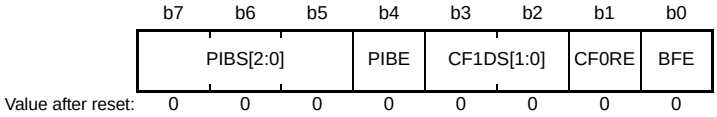
Address(es): SCI12.CR0 0008 B321h

	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

27.2.19 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h



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

27.2.20 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B320h

	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 is enabled (SCI base clock). 0 1 0: Filter is enabled (PCLK/8). 0 1 1: Filter is enabled (PCLK/16). 1 0 0: Filter is enabled (PCLK/32). 1 0 1: Filter is enabled (PCLK/64). 1 1 0: Filter is enabled (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 1. 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. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.

27.2.21 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 Bits (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

27.2.22 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 Selection)

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.

27.2.23 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

27.2.24 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 BCDFL 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

27.2.25 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

27.2.26 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.

27.2.27 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

27.2.28 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 read-only register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

27.2.29 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.

27.2.30 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.

27.2.31 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

27.2.32 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 read-only register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

27.2.33 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

27.2.34 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: Writing is to the reload register and the counter. 1: Writing is only to the reload register	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)

These bits determine whether a value written to TPRE or TCNT is only written to the reload register or is written to both the reload register and the counter.

27.2.35 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 eight-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 1 system operating clock cycle to load a value from the reload register to the counter.

27.2.36 Timer Count Register (TCNT)

Address(es): SCI12.TCNT0008 B333h

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

TCNT consists of an eight-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 1 system operating clock cycle to load a value from the reload register to the counter.

27.3 Operation in Asynchronous Mode

Figure 27.5 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 the space state (low level) while the SEMR.RXDESEL bit is 0 and a falling edge of the space state (low level) while the SEMR.RXDESEL bit is 1, it recognizes a start bit and starts serial communications.
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.

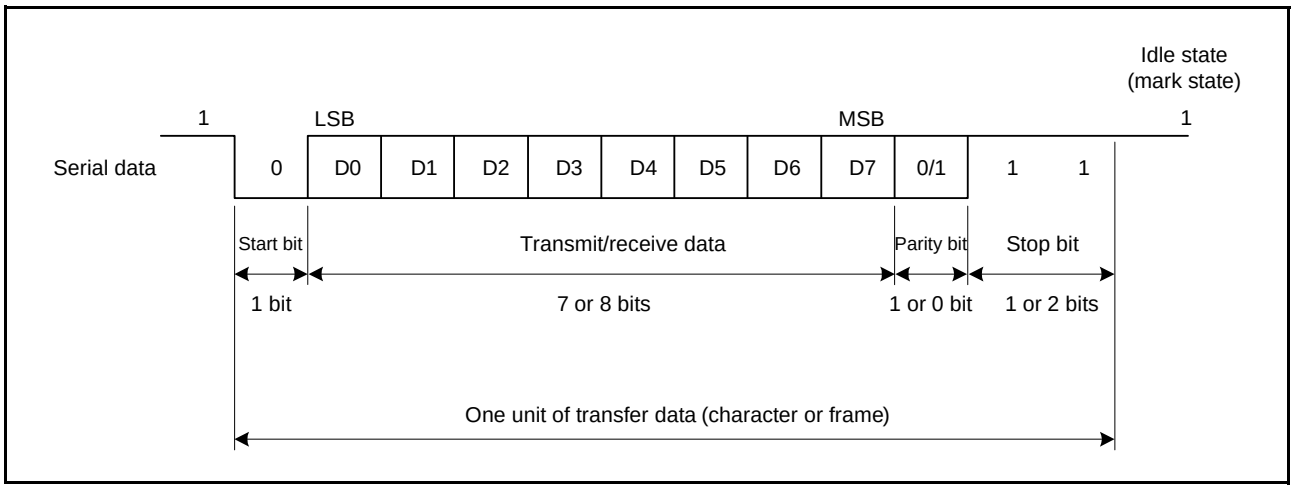


Figure 27.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

27.3.1 Serial Data Transfer Format

Table 27.22 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, see section 27.4, Multi-Processor Communications Function.

Table 27.22 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

27.3.2 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 27.6. 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 12)
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.

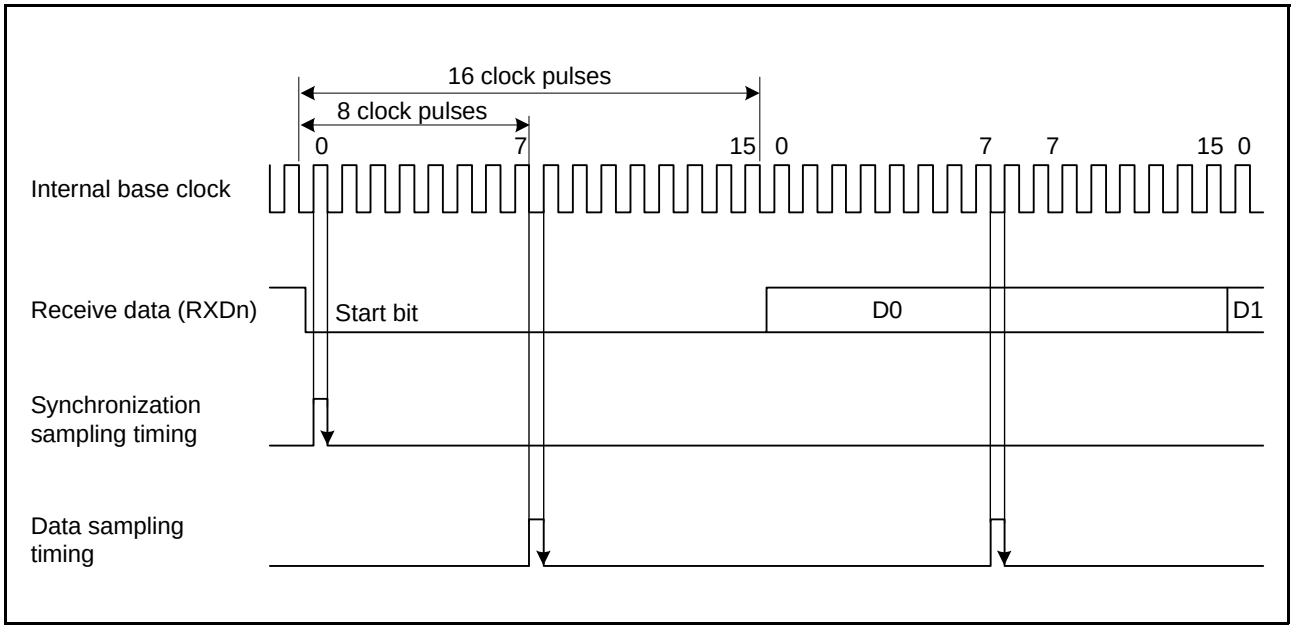


Figure 27.6 Receive Data Sampling Timing in Asynchronous Mode

27.3.3 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 ACS0 bit in SEMR of SCIn (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 27.7.

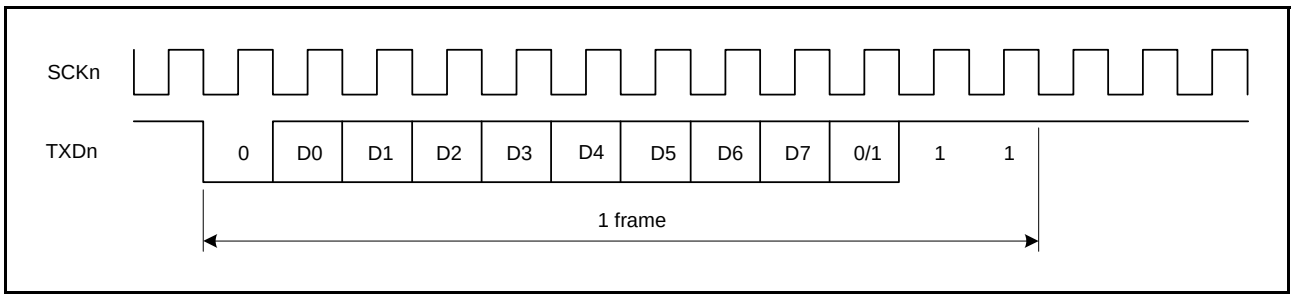


Figure 27.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

27.3.4 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 low 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.

27.3.5 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 the sample flowchart (Figure 27.8). 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 clearing 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.

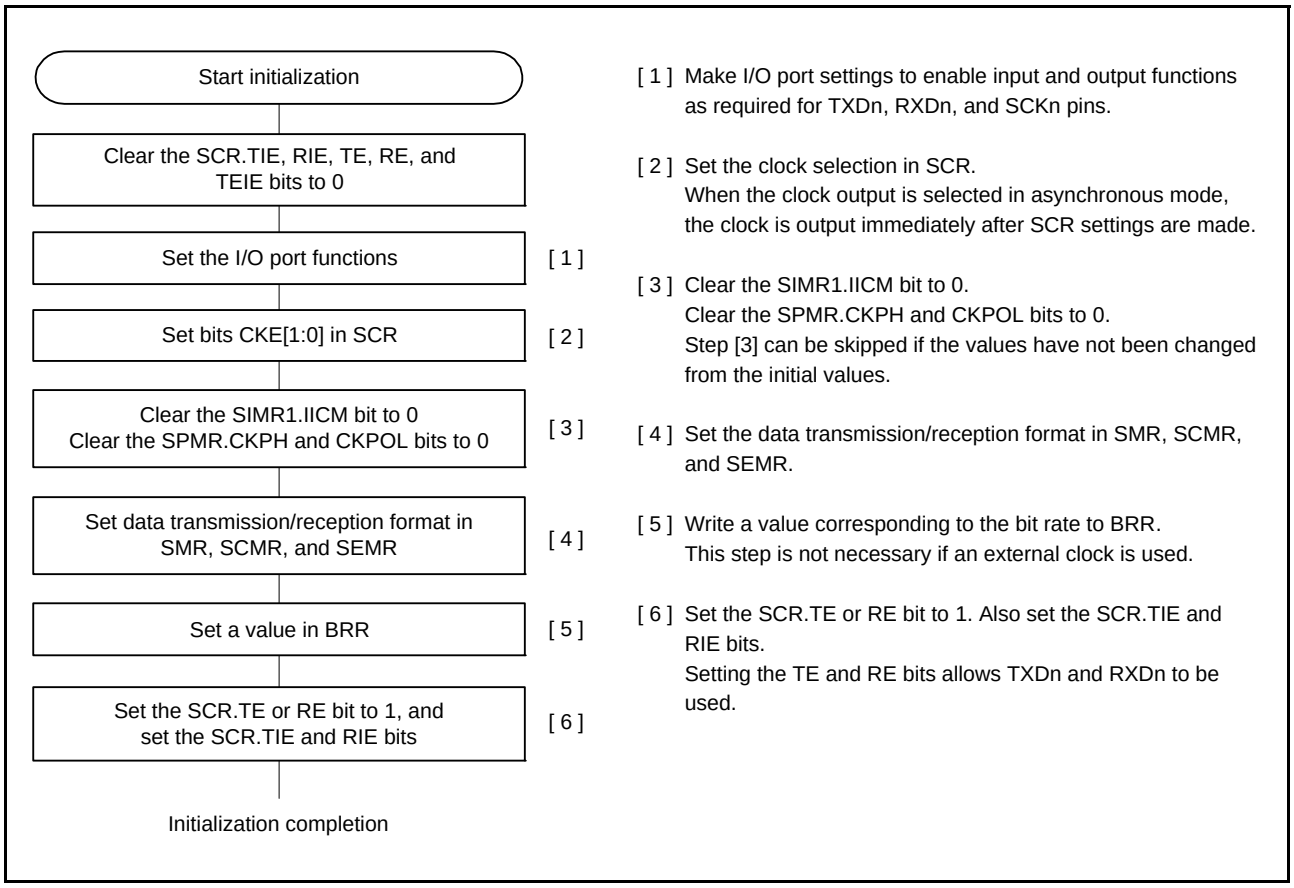
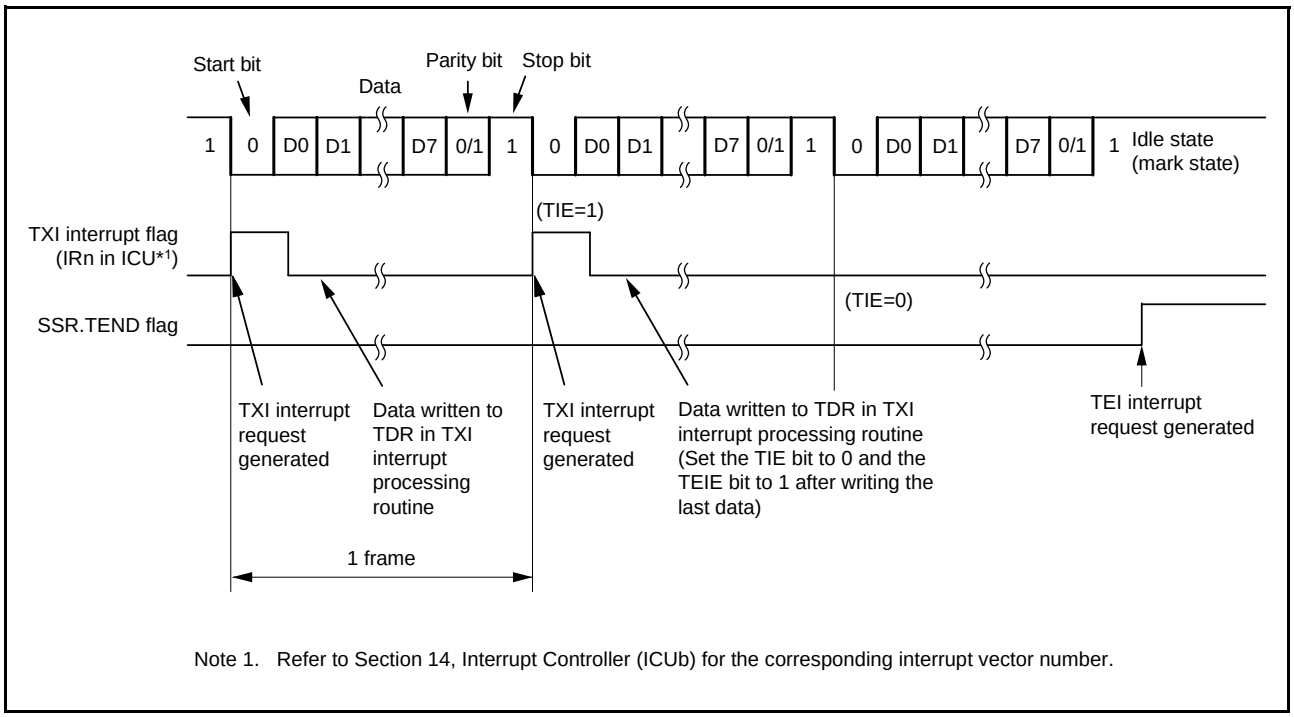


Figure 27.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

27.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 27.9 shows an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

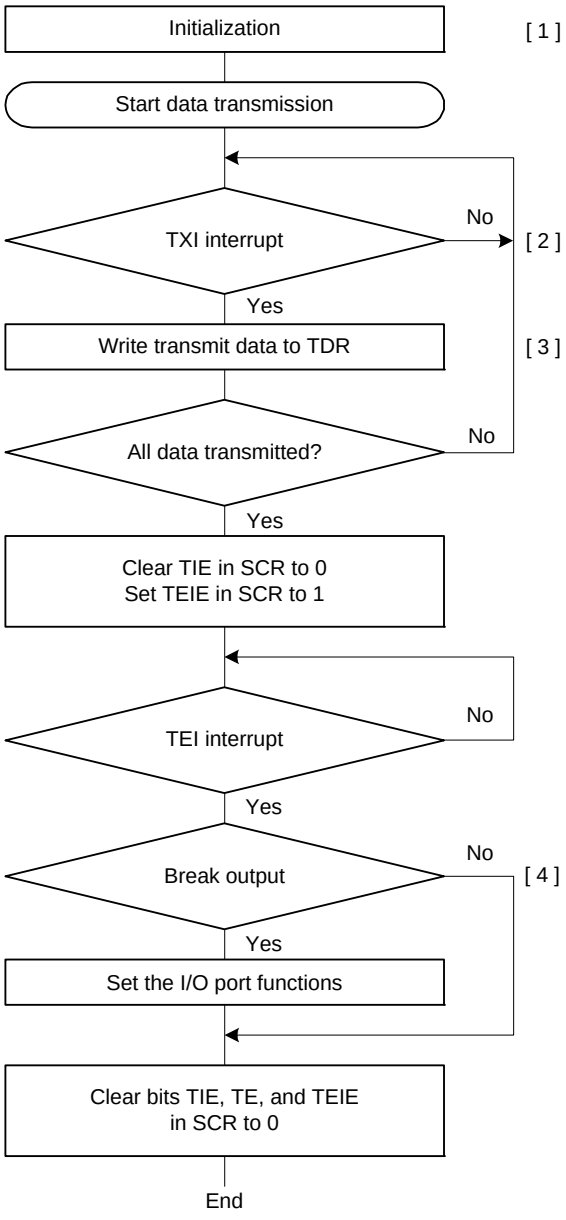
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (disabling the CTS function) and a low level on the CTS# 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 data for transmission to TDR in the TXI interrupt processing routine before transmission of the current data for transmission is completed.
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 disabled) or a low level input on the CTSn# pin cause the next transfer of the next data for transmission 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 flag in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.



Note 1. Refer to Section 14, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 27.9 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion) (Example with 8-Bit Data, Parity, One Stop Bit)

Figure 27.10 shows a sample flowchart for serial transmission in asynchronous mode.



[1] SCI initialization:
The TXDn automatically becomes the output pin for data being transmitted.
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 processing 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 DMAC or DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

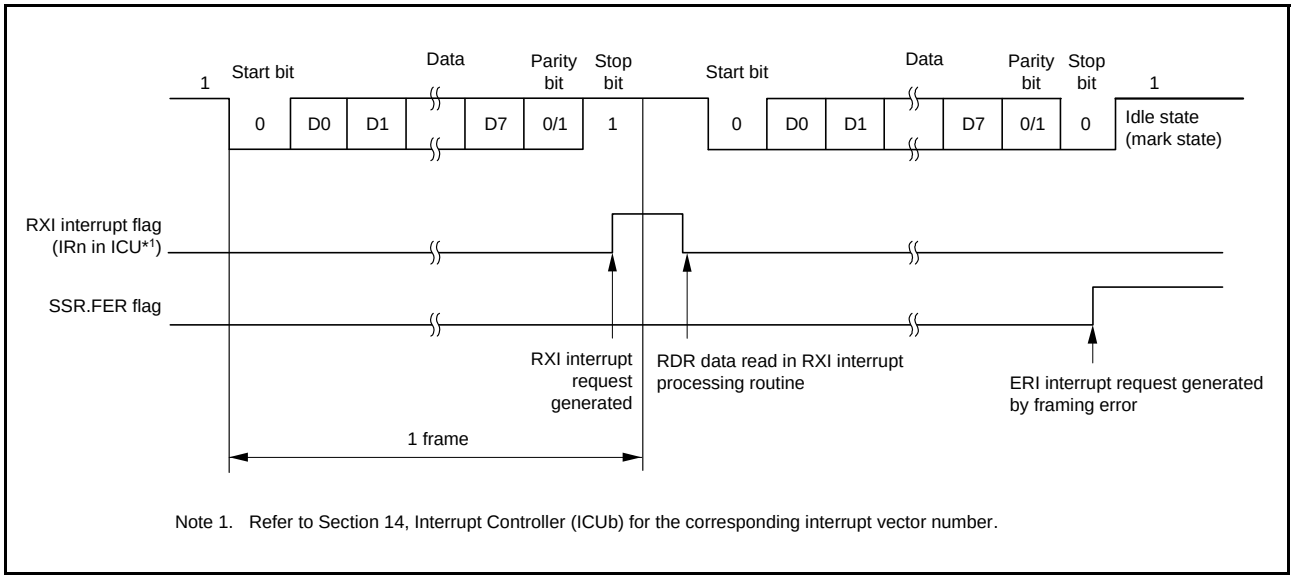
[4] To output a break in serial transmission, set the I/O port that shares a pin with the TXDn pin to “low output”, and switch from the TXDn pin to the general I/O port before setting the TE bit in the SCR register to 0.

Figure 27.10 Example of Serial Transmission Flowchart in Asynchronous Mode

27.3.7 Serial Data Reception (Asynchronous Mode)

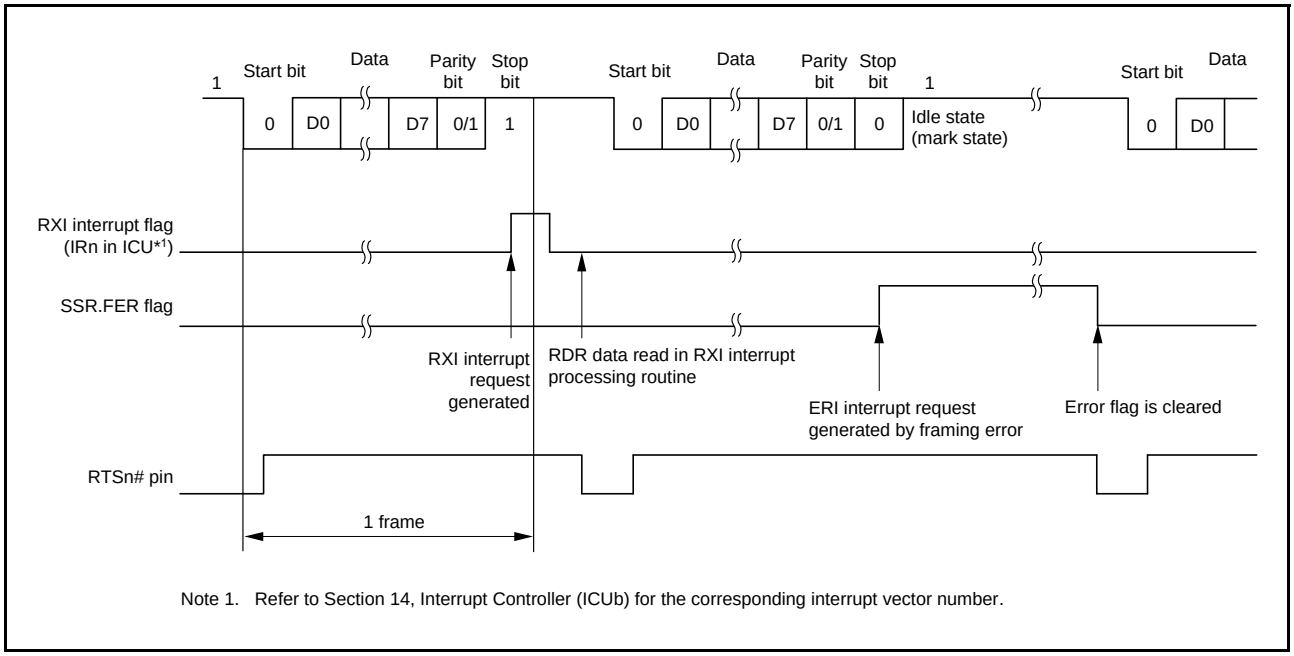
Figure 27.11 and Figure 27.12 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

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 processing 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.



Note 1. Refer to Section 14, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 27.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (Example with 8-Bit Data, Parity, One Stop Bit)



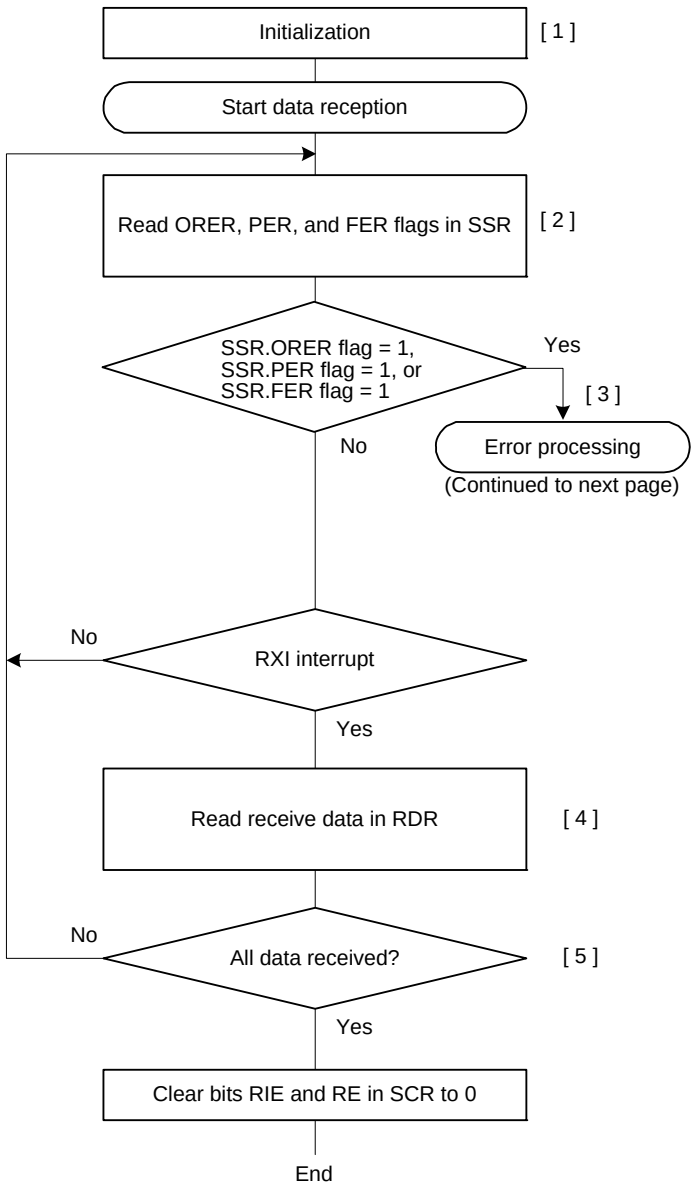
Note 1. Refer to Section 14, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 27.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function Is Used) (Example with 8-Bit Data, Parity, One Stop Bit)

Table 27.23 lists the states of the SSR status flags 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, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.
Figure 27.13 and Figure 27.14 show samples of flowcharts for serial data reception.

Table 27.23 SSR Status Flags and Receive Data Handling

SSR Status Flag				
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



[1] SCI initialization:
The RXDn automatically becomes the input pin for data being received.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to clear the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxDn pin.

[4] Read the receive data in RDR once in the RXI interrupt processing routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt processing routine. The RDR data can also be read by activating the DMAC or DTC.

Figure 27.13 Example of Serial Reception Flowchart (1) (Asynchronous Mode)

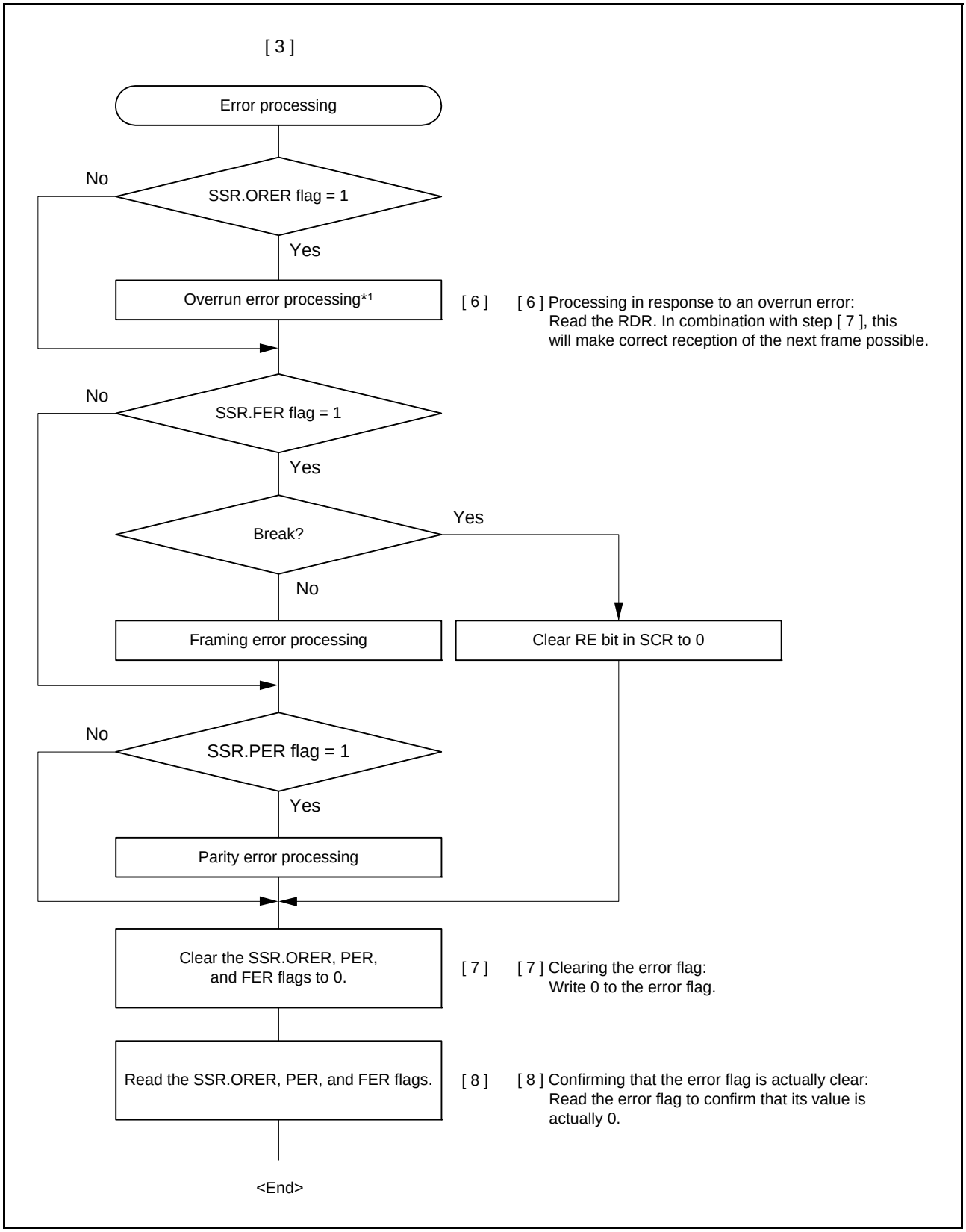


Figure 27.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode)

27.4 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 27.15 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 transmission 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 reception 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.

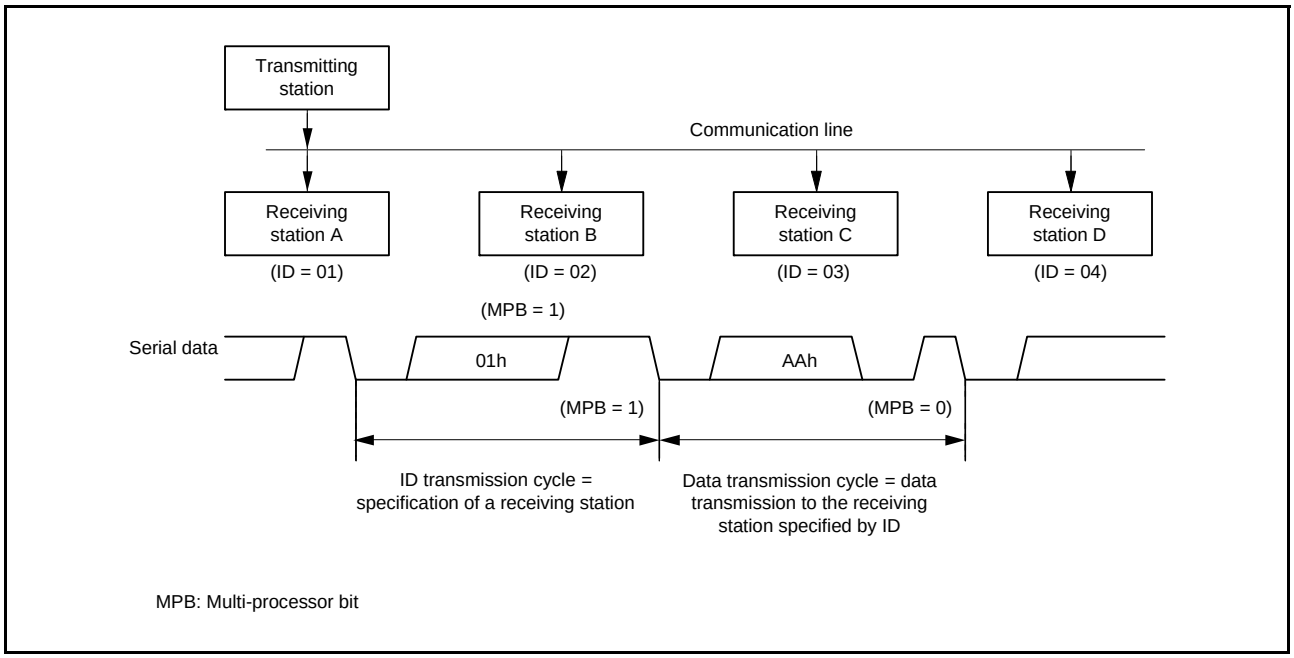


Figure 27.15 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

27.4.1 Multi-Processor Serial Data Transmission

Figure 27.16 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.

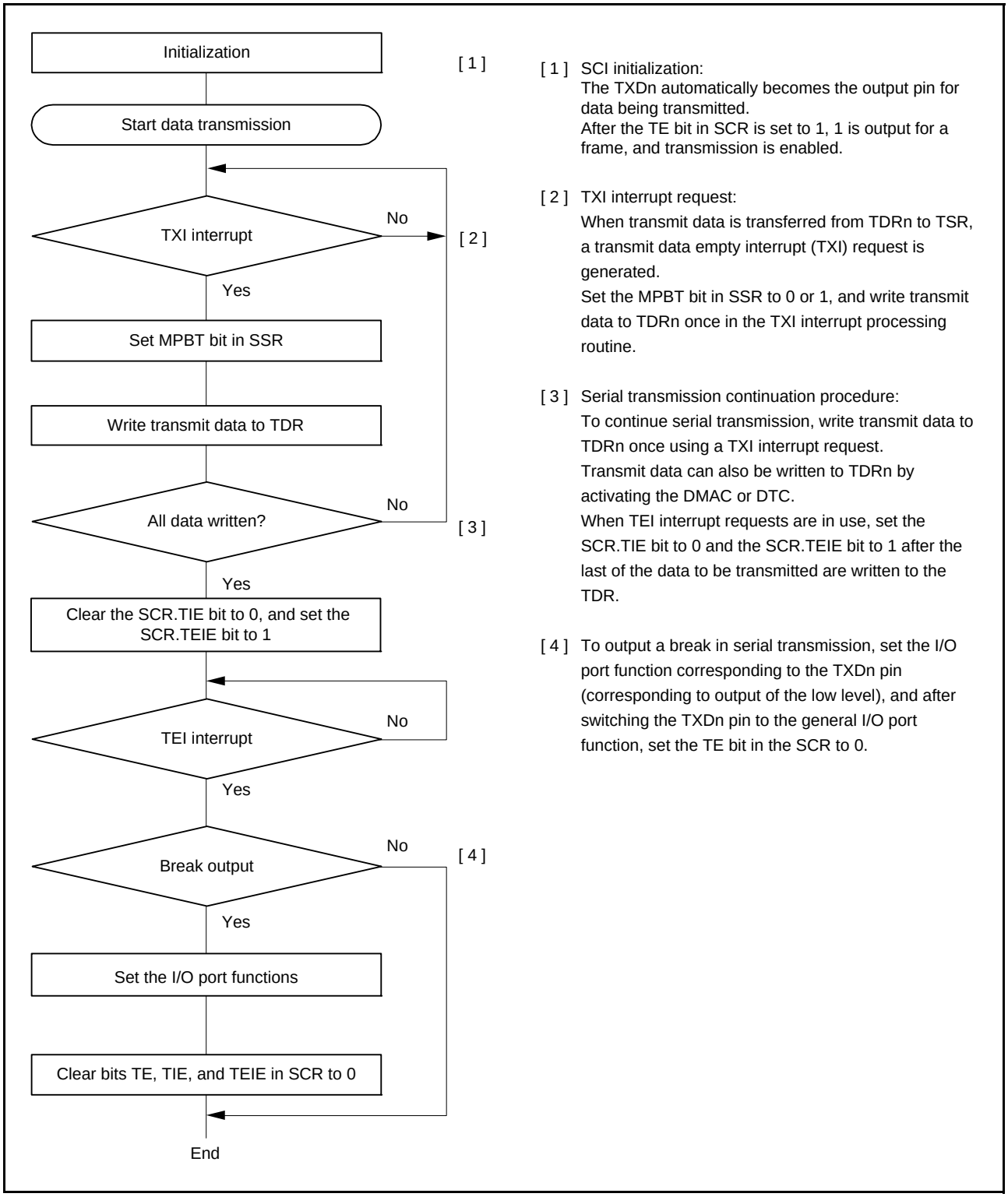


Figure 27.16 Example of Multi-Processor Serial Transmission Flowchart

27.4.2 Multi-Processor Serial Data Reception

Figure 27.18 and Figure 27.19 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 27.17 is the example of operation for reception.

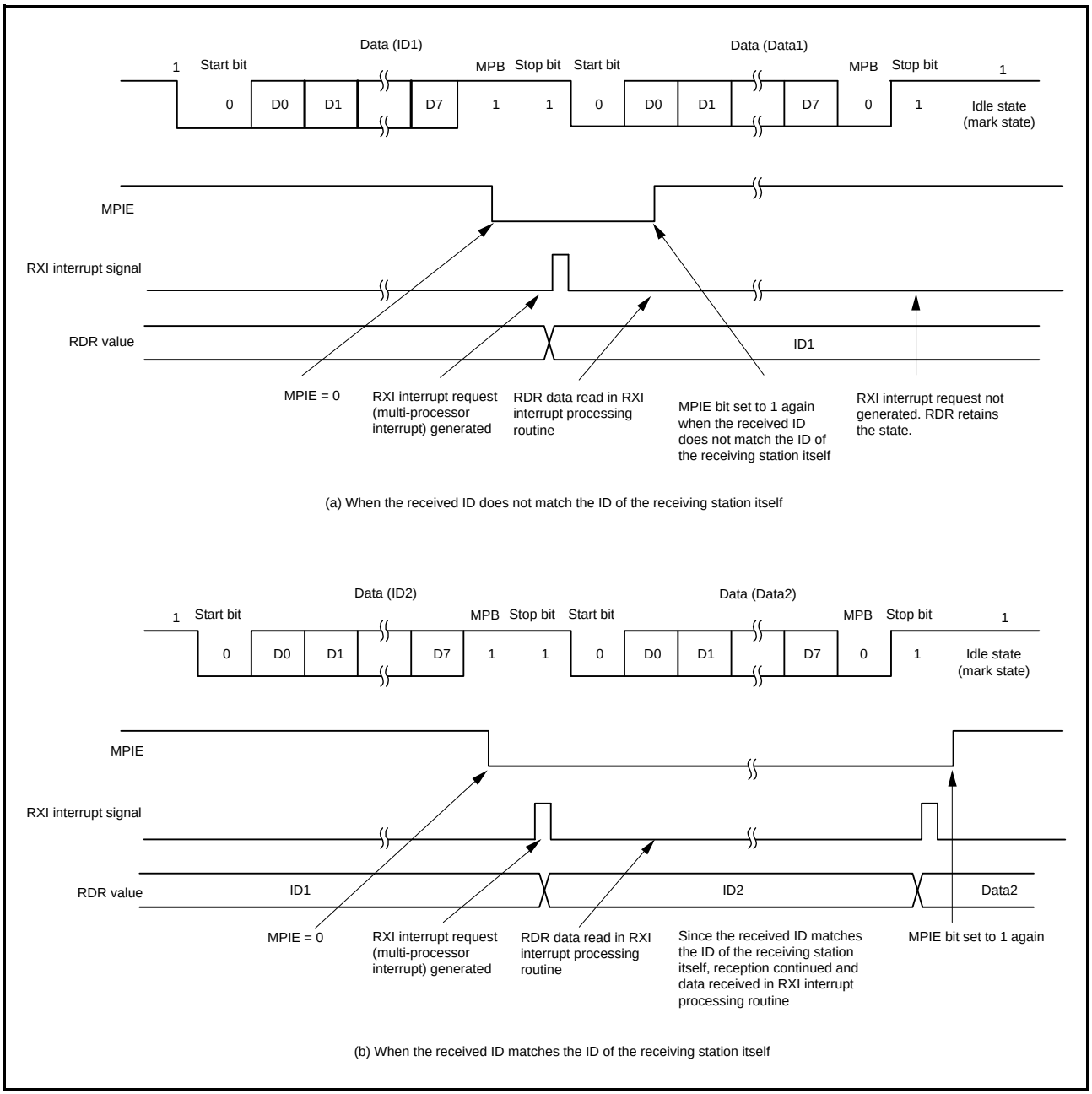


Figure 27.17 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit)

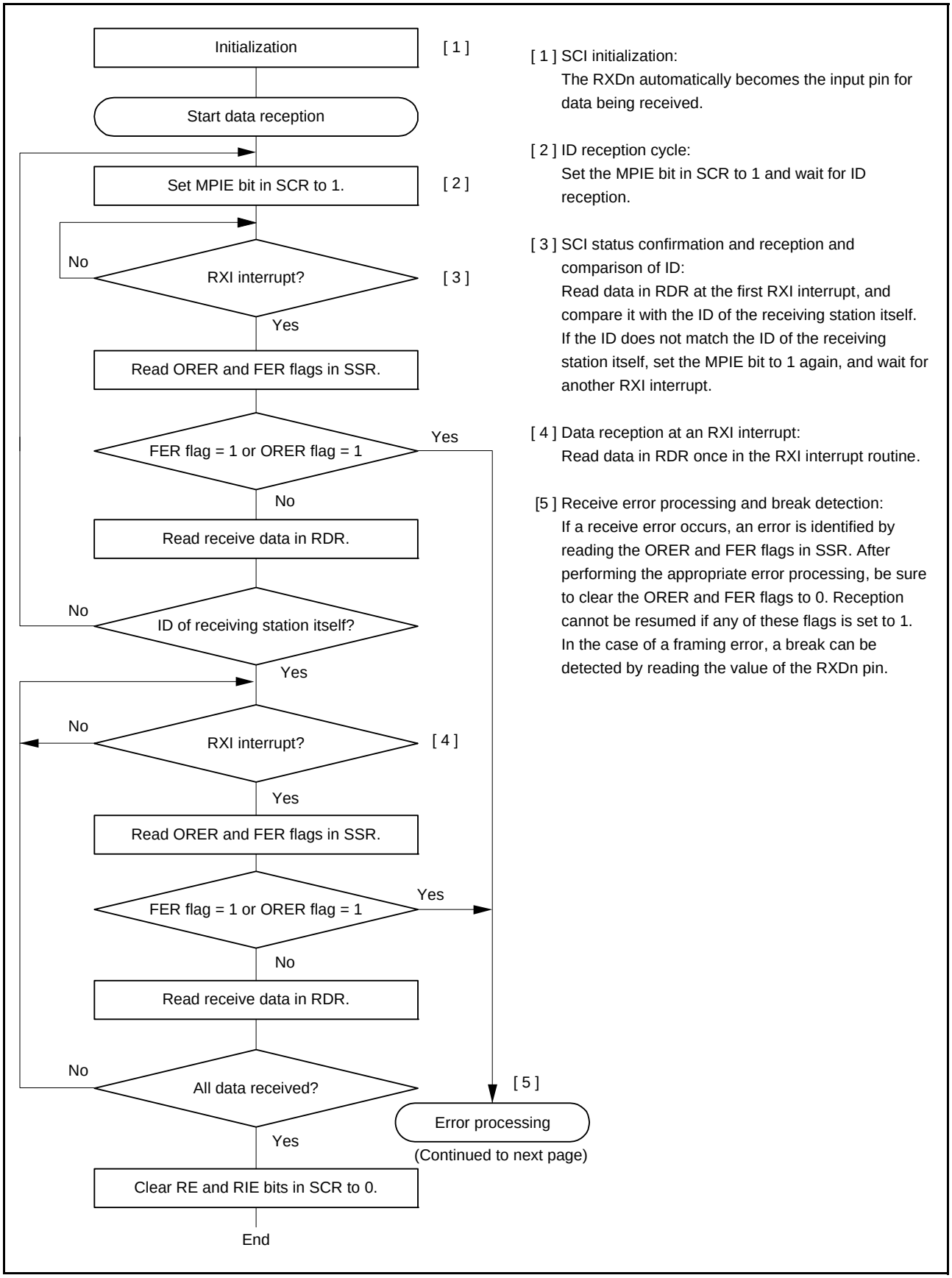


Figure 27.18 Example of Multi-Processor Serial Reception Flowchart (1)

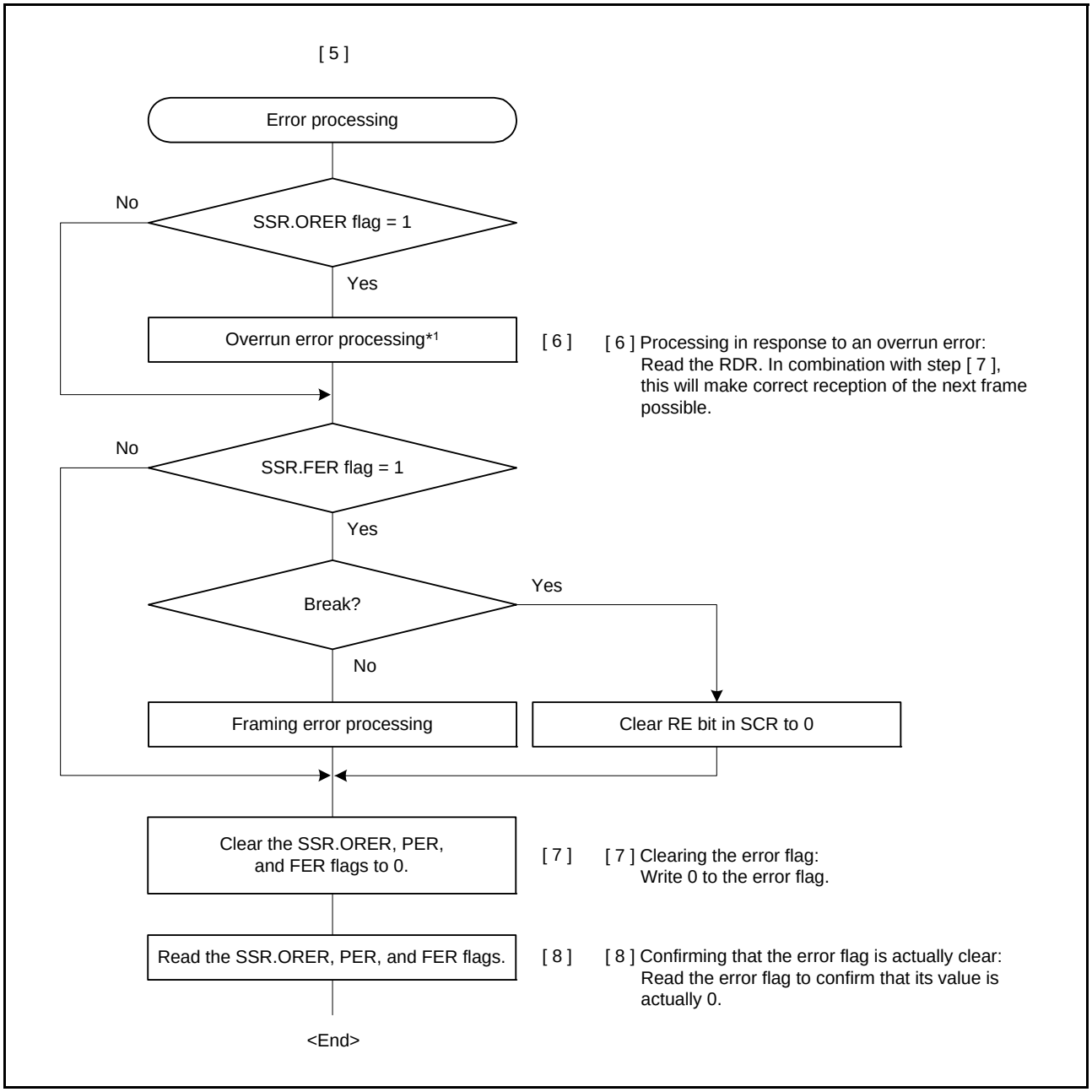


Figure 27.19 Example of Multi-Processor Serial Reception Flowchart (2)

27.5 Operation in Clock Synchronous Mode

Figure 27.20 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 transmission 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.

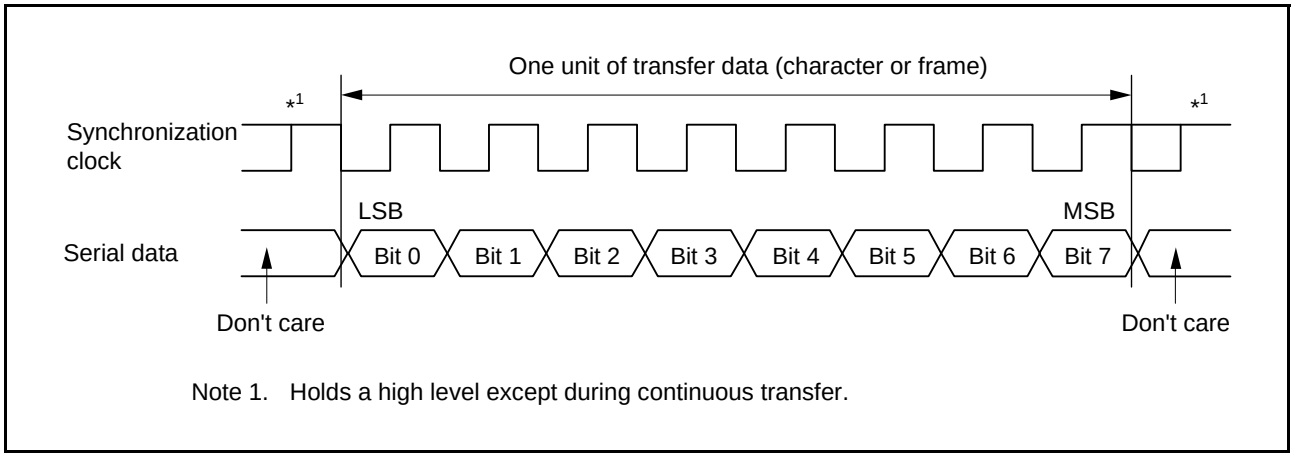


Figure 27.20 Data Format in Clock Synchronous Serial Communications (LSB-First)

27.5.1 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 CKE[1:0] bits in SCR.
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, output of the synchronizing clock signal continues until the CTS function is enabled and the high level is input on the CTSn# pin, an overflow error occurs, or the RE bit in SCR is set to 0. When the CTS function is enabled, the synchronous clock signal output is stopped if the CTSn# pin input is high on completion of the frame reception.

27.5.2 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.

27.5.3 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 the sample flowchart (Figure 27.21). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.
Note that clearing 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.

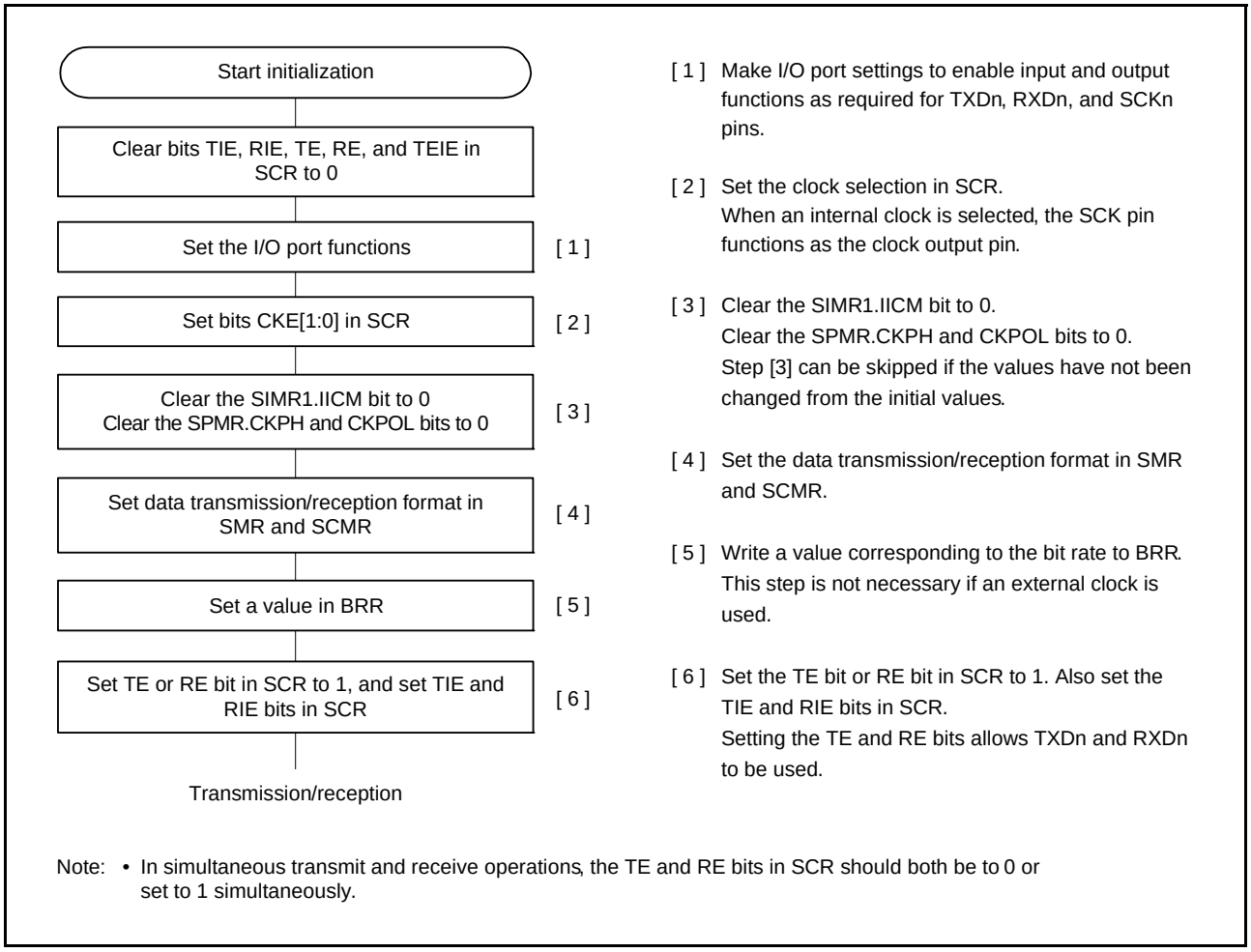


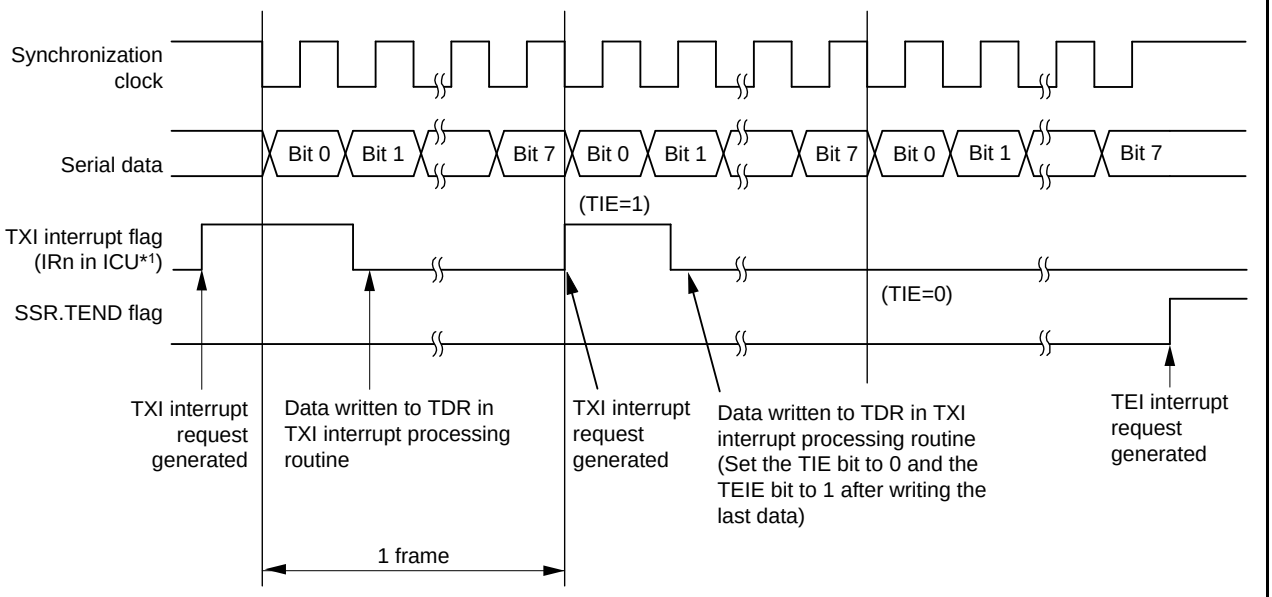
Figure 27.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

27.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 27.22 shows an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

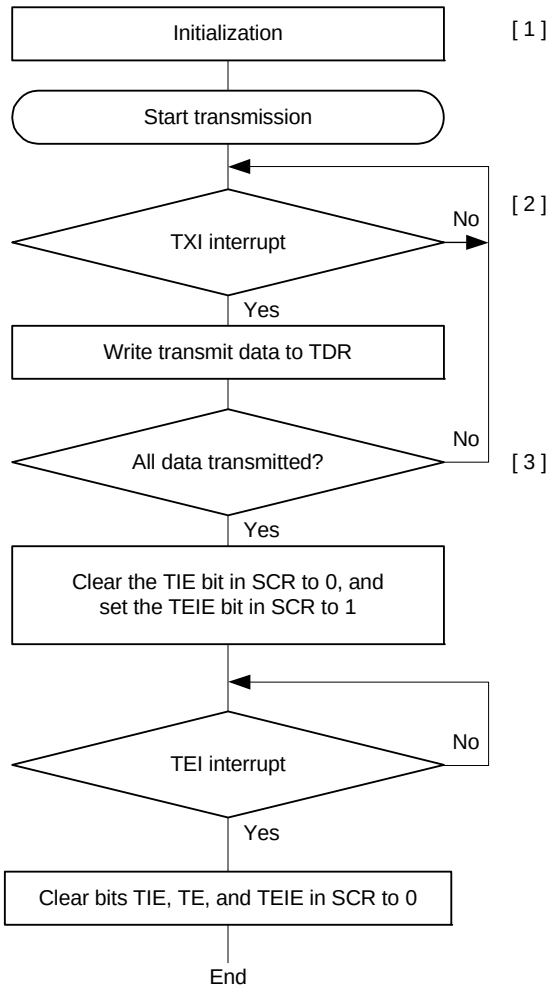
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously 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 processing routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing 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 (enabling the CTS function).
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 flag in TEND 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 27.23 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 clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.



Note 1. Refer to Section 14, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 27.22 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
The TXDn automatically becomes the output pin for data being transmitted.

[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 processing 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). 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.

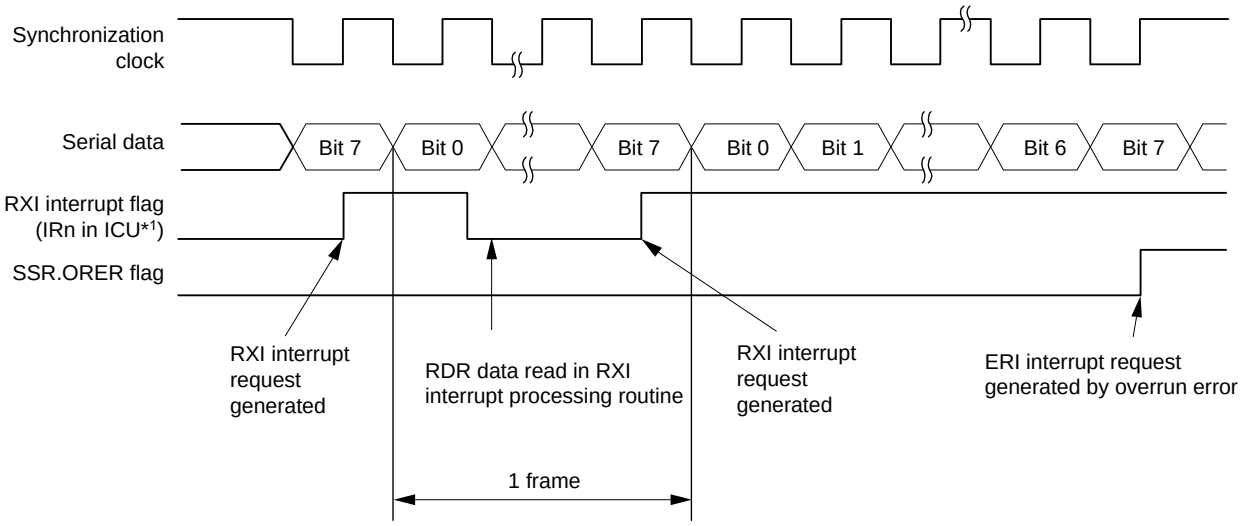
Note: • When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Clearing the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 27.23 Example of Serial Transmission Flowchart (Clock Synchronous Mode)

27.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 27.24 and Figure 27.25 shows 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 RTS 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 processing 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 Section 14, Interrupt Controller (ICUb) for the corresponding interrupt vector number.

Figure 27.24 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Function is not Used)

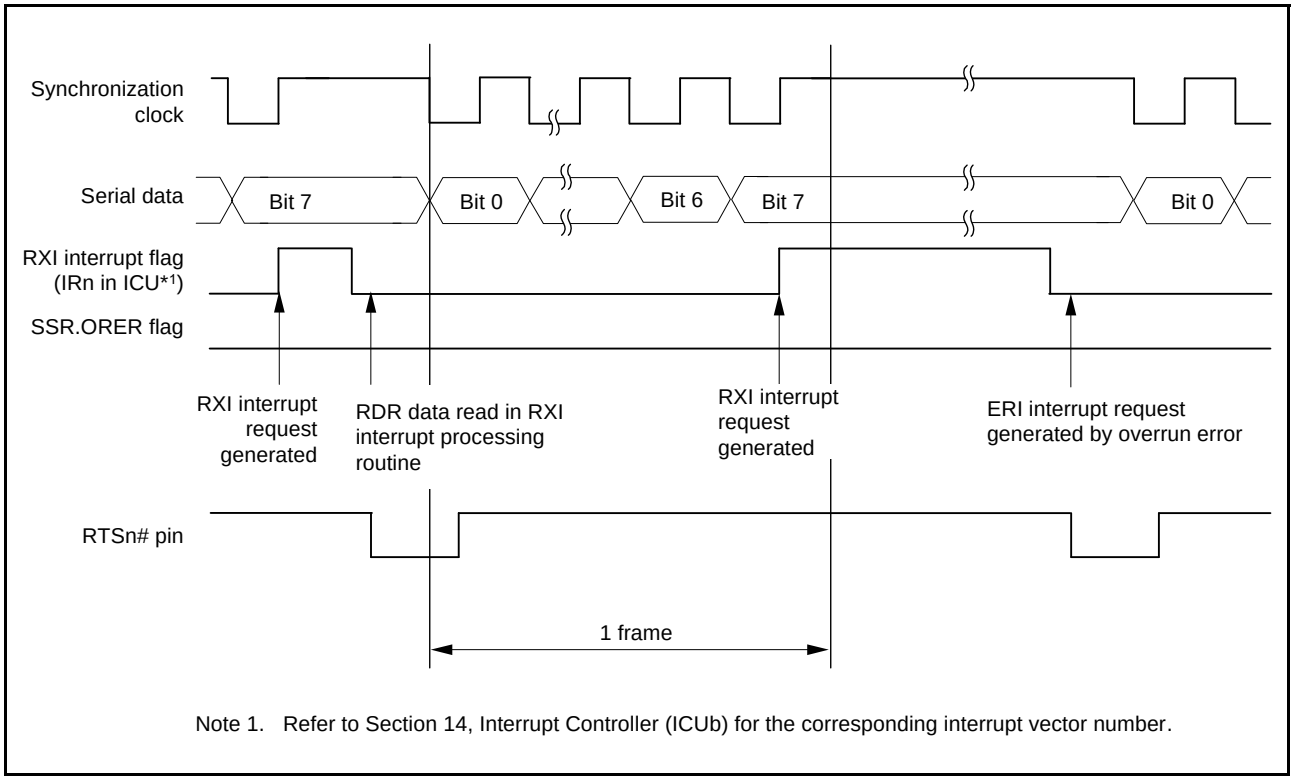
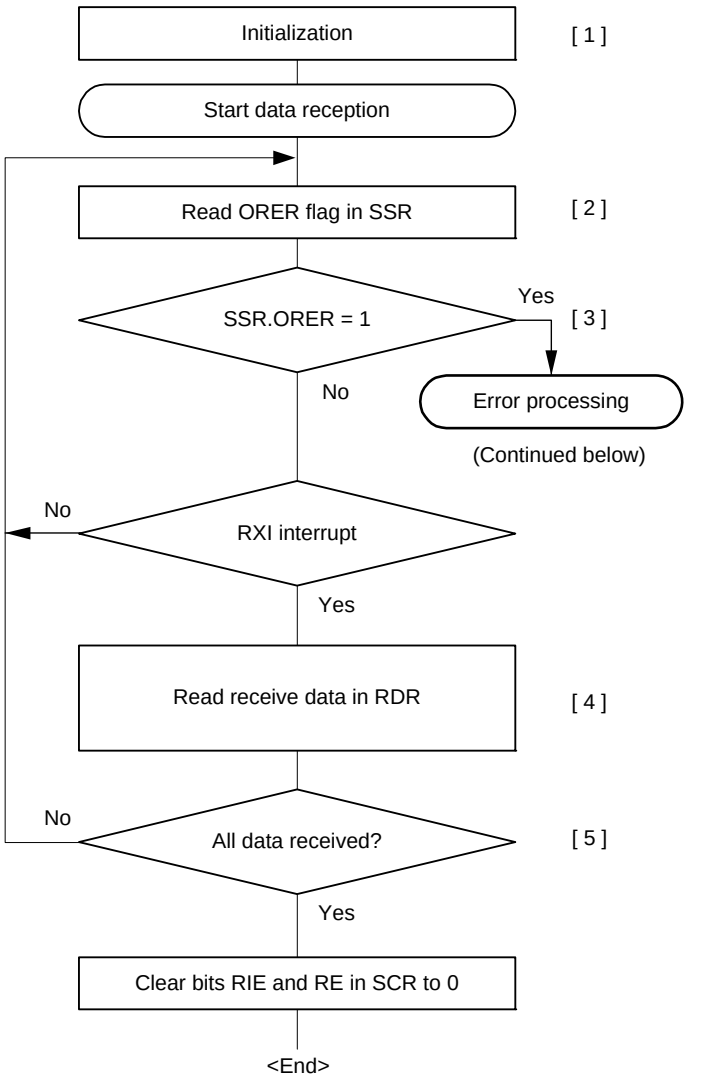
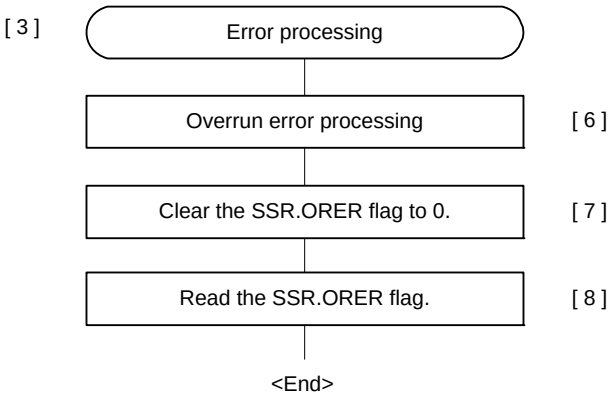


Figure 27.25 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (when RTS Function is Used)

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.
Figure 27.26 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 clear 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 processing 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 actually clear:
Read the error flag to confirm that its value is actually 0.

Figure 27.26 Example of Serial Reception Flowchart (Clock Synchronous Mode)

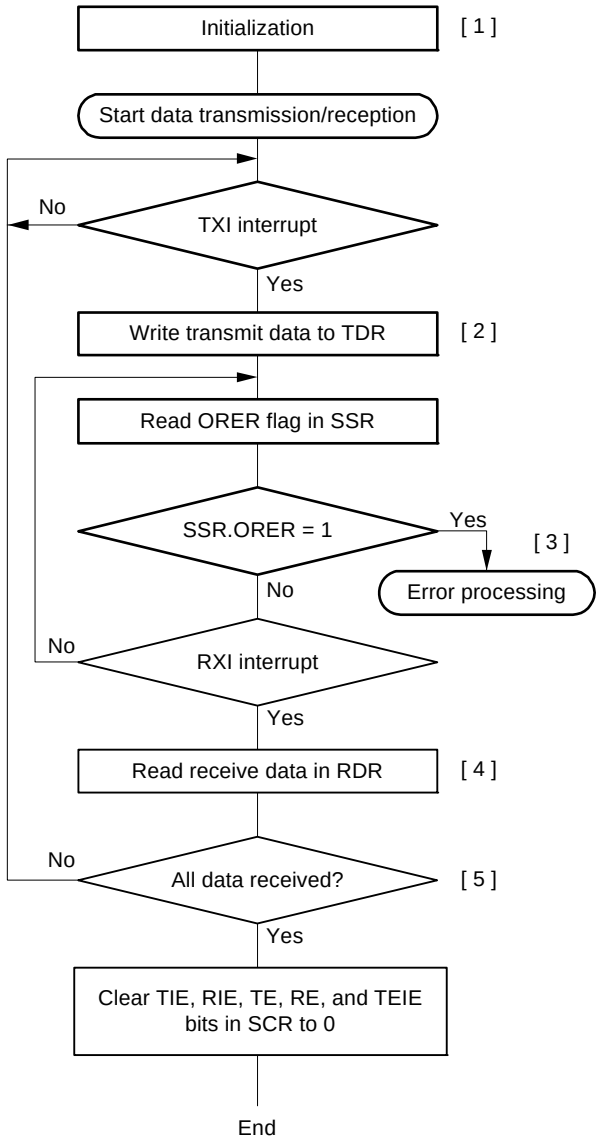
27.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 27.27 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 clear 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 request processing routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then clear 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 processing 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 clear 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 27.27 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode)

27.6 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard, as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

27.6.1 Sample Connection

Figure 27.28 shows a sample connection between a smart card (IC card) and this LSI.
As in the figure, since this LSI 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 RX220 can be used to output a reset signal.

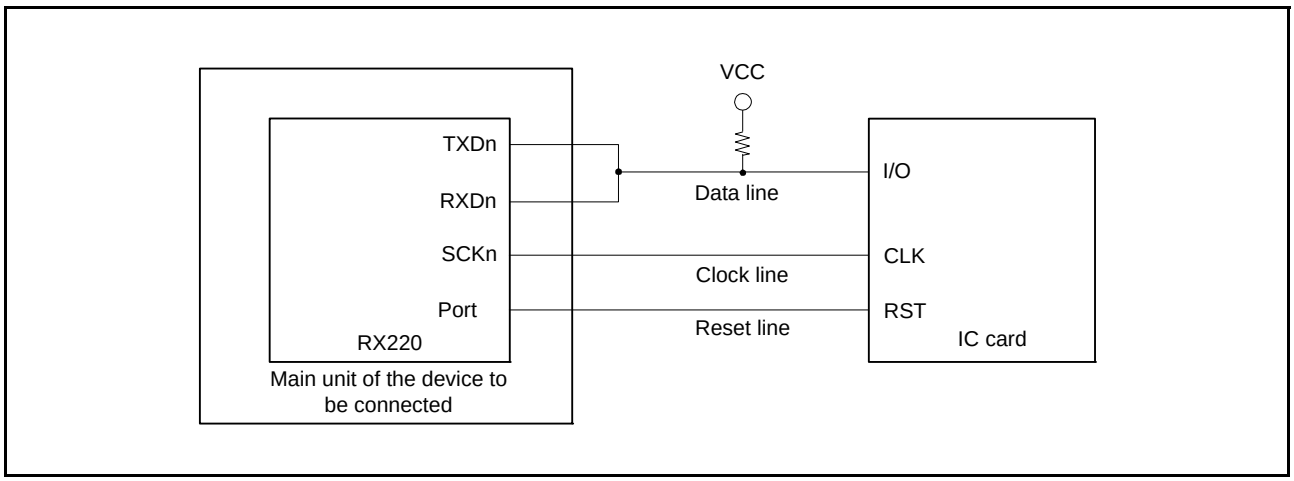


Figure 27.28 Sample Connection with a Smart Card (IC Card)

27.6.2 Data Format (Except in Block Transfer Mode)

Figure 27.29 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 one 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 re-transmitted after at least 2 etu.

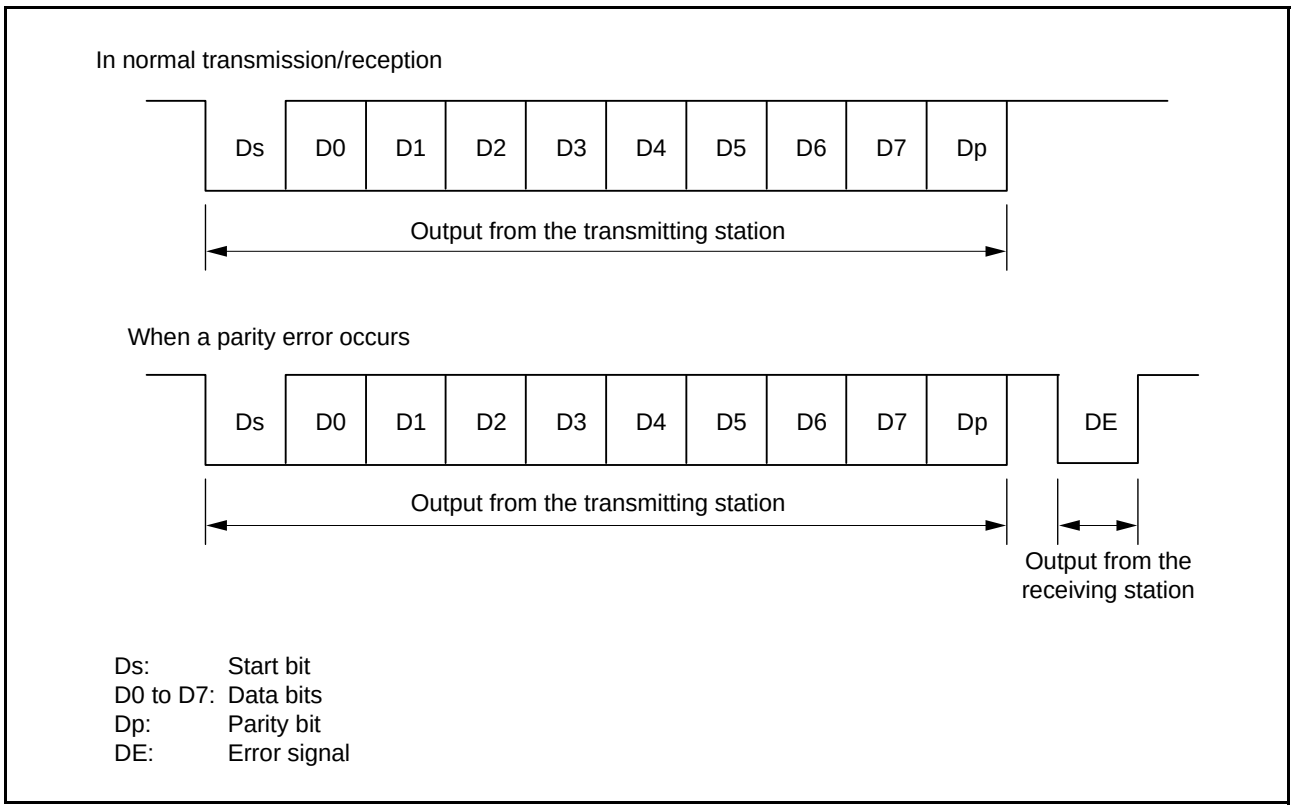


Figure 27.29 Data Formats in Smart Card Interface Mode

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 27.30. 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.

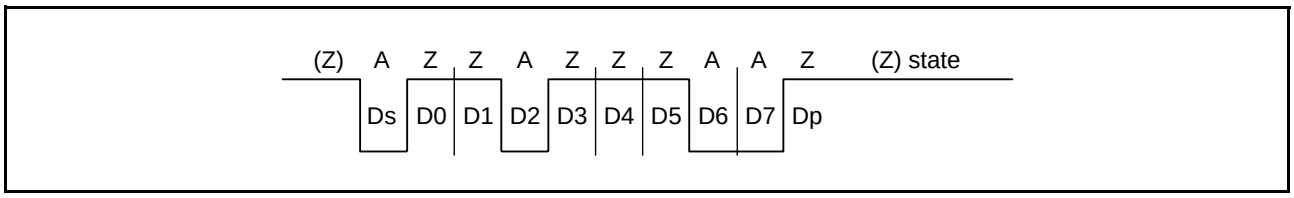


Figure 27.30 Direct Convention (SDIR in SCMR = 0, SINV in SCMR =0, PM in SMR = 0)

(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 27.31. 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 RX220 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.

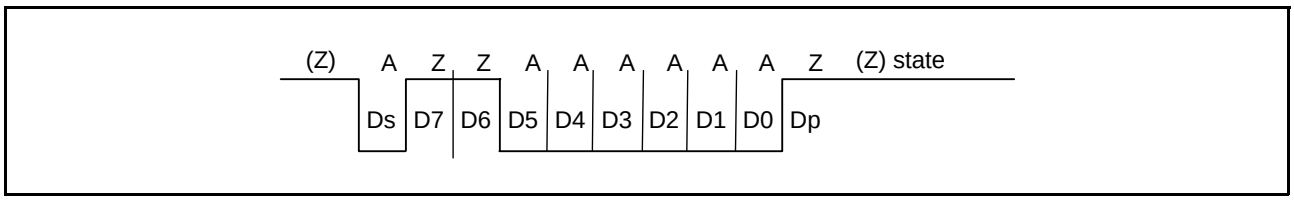


Figure 27.31 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR =1, PM in SMR = 1)

27.6.3 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 re-transmitted 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 always read as 0 because no error signal is transferred.

27.6.4 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 27.32. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

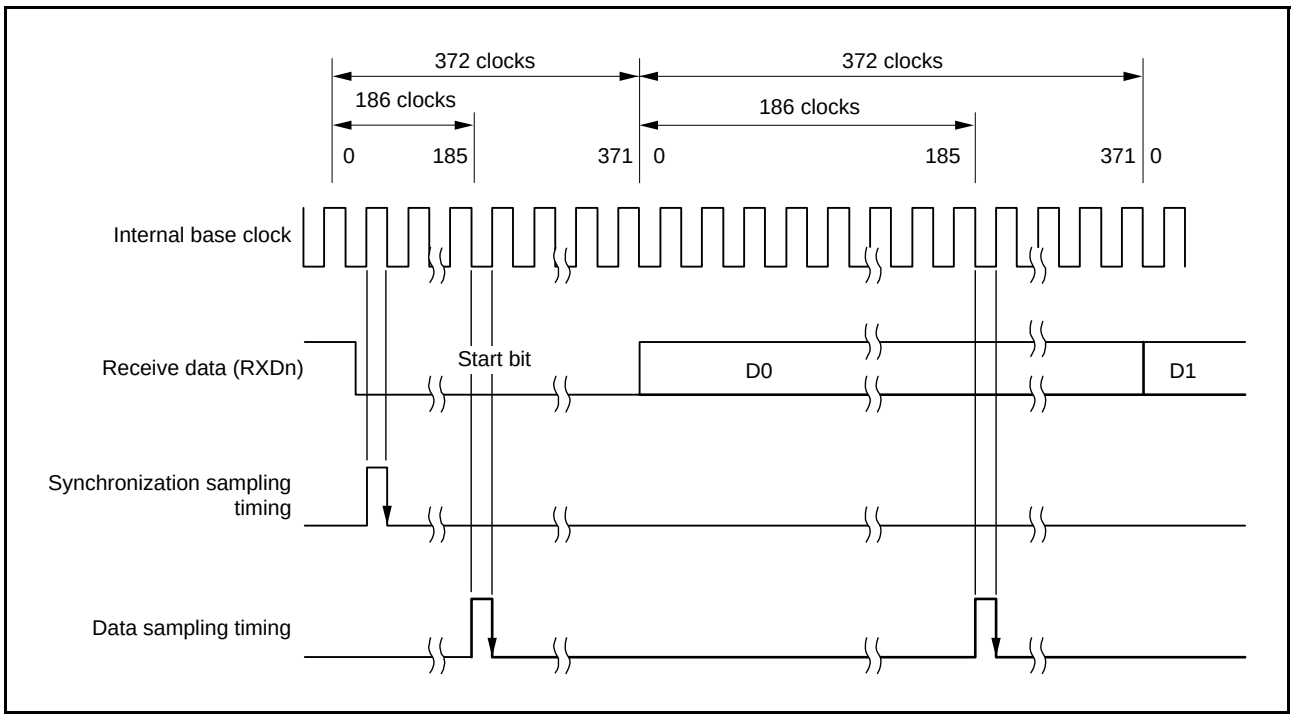


Figure 27.32 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

27.6.5 Initialization of the SCI (Smart Card Interface Mode)

Before transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Write the initial value “00h” to the SCR.
2. Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
3. Set the error flags ORER, ERS, and PER in SSR to 0.
4. Clear the SIMR1.IICM bit and the SPMR.CKPH and CKPOL bits to 0.
   This step can be skipped if the values have not been changed from the initial values.
5. Set bits GM, BLK, PM, BCP[1:0], and CKS[1:0] in SMR and the BCP2 bit in SCMR appropriately. Also set the PE bit in SMR to 1.
6. Set bits SDIR, SINV, and SMIF in SCMR appropriately. Set the SEMR.RXDESEL bit to 0. Then, the TXDn and RXDn pins are placed in the high impedance state.
7. Set the value corresponding to the bit rate in BRR.
8. Set the CKE[1:0] bits in SCR appropriately, and set bits TIE, RIE, TE, RE, and TEIE in SCR to 0 at the same time. When the CKE0 bit is set to 1, the SCKn pin is allowed to output clock pulses.
9. Set the TIE, RIE, TE, and RE bits in SCR to 1. Setting the TE and RE bits to 1 simultaneously is prohibited except for self-diagnosis.

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.

27.6.6 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 re-transmitted, is different from that in normal serial communications interface mode. Figure 27.33 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 27.35 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 cleared to 0 when the DTC or DMAC transfers the data.
If an error occurs, the SCI automatically re-transmits 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, see section 16, DMA Controller (DMACA) and section 17, Data Transfer Controller (DTCa).

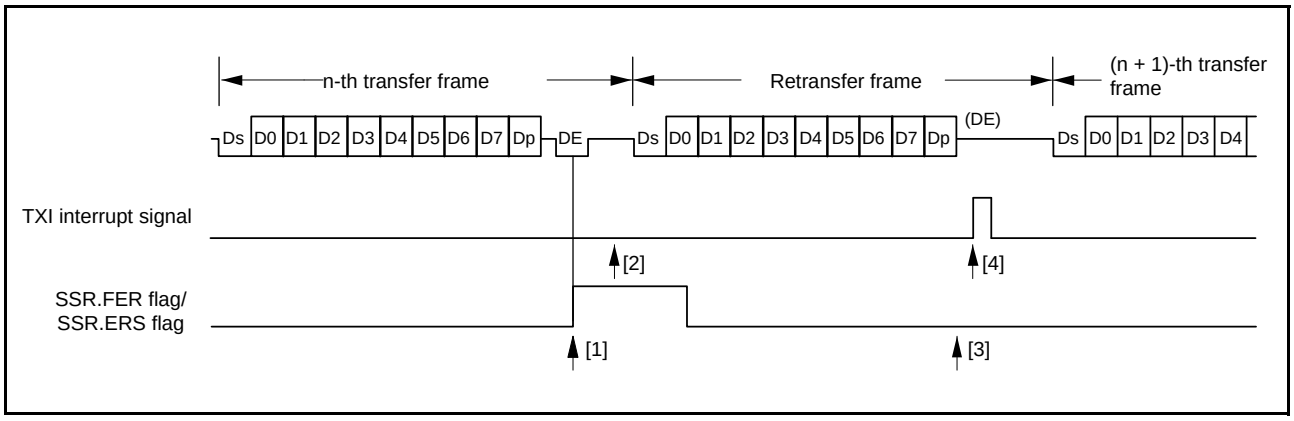


Figure 27.33 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 27.34 shows the TEND flag generation timing.

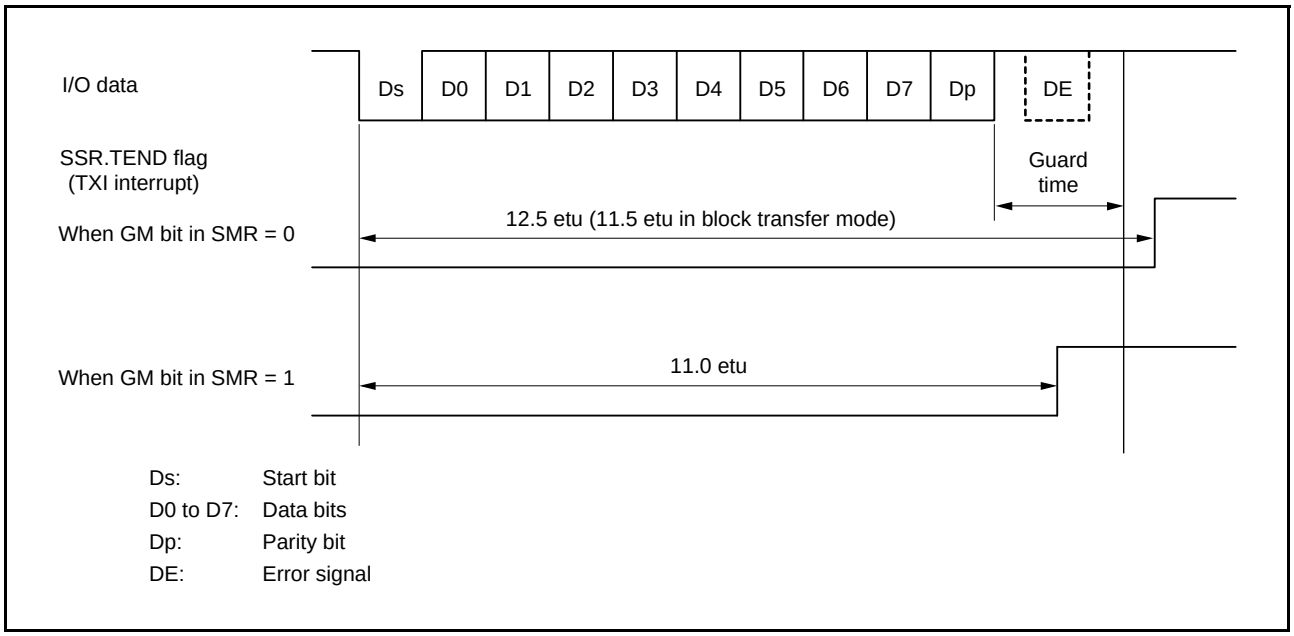


Figure 27.34 SSR.TEND Flag Generation Timing during Transmission

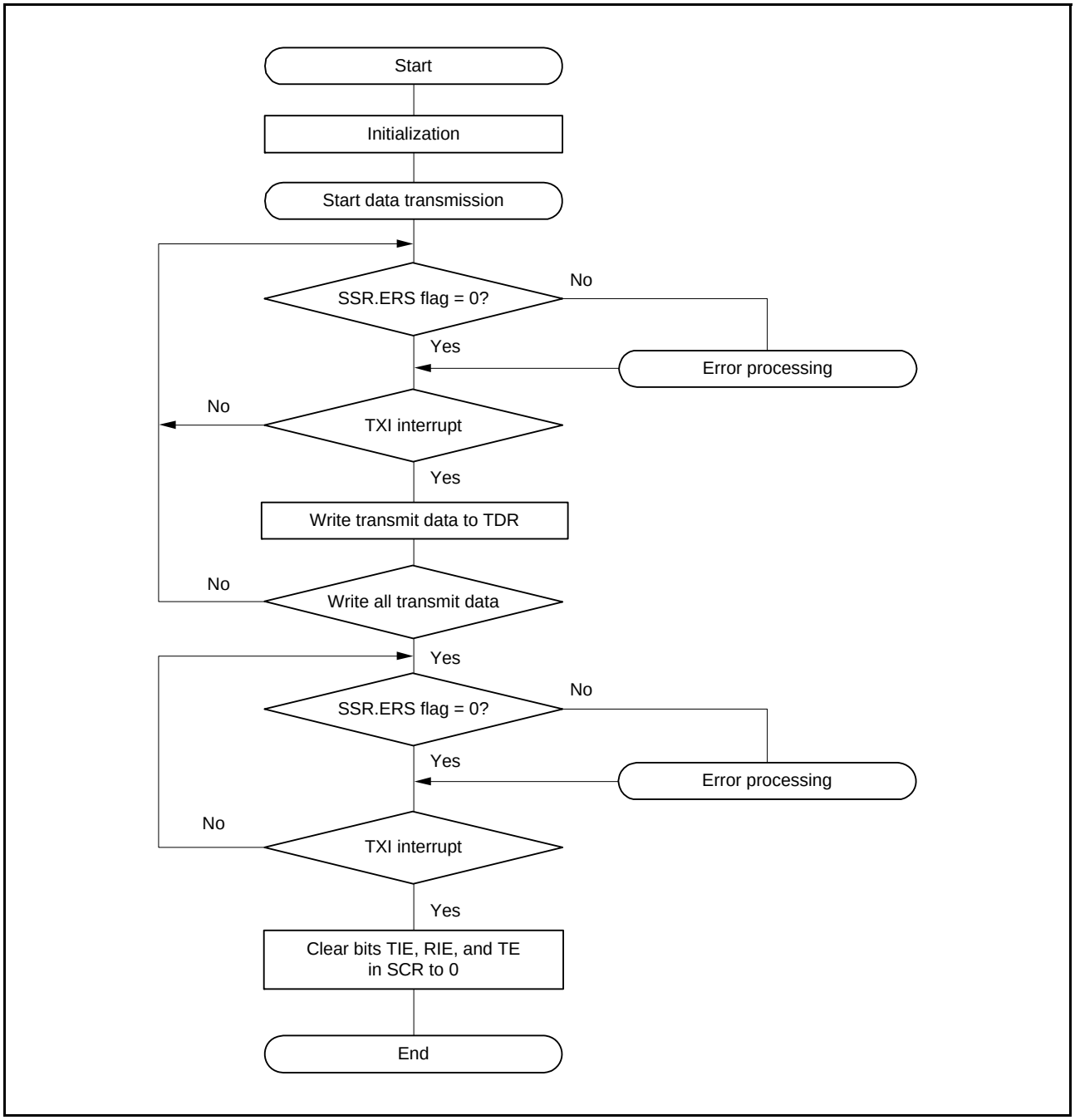


Figure 27.35 Sample Smart Card Interface Transmission Flowchart

27.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in serial communications interface mode. Figure 27.36 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 27.37 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, see section 27.3, Operation in Asynchronous Mode.

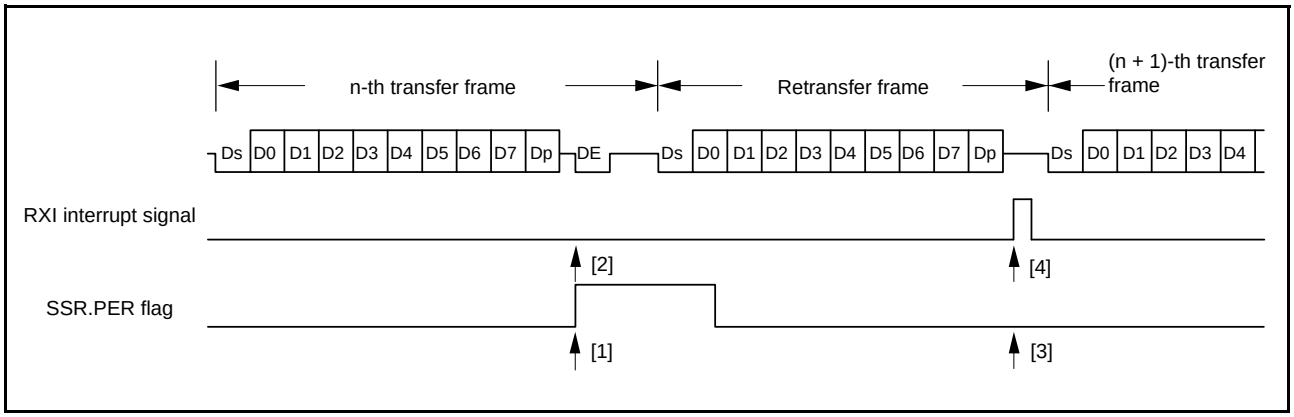


Figure 27.36 Data Retransfer Operation in SCI Reception Mode

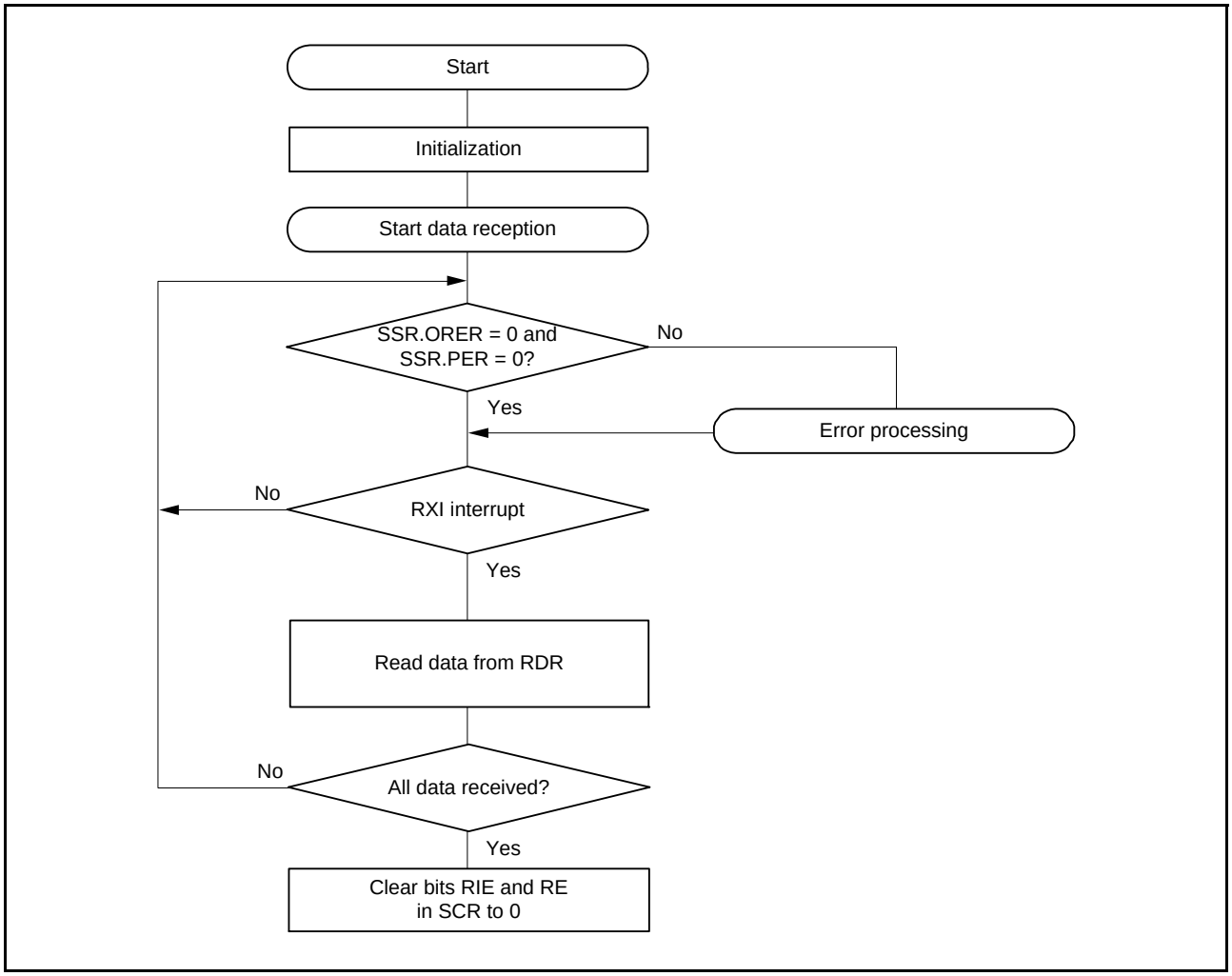


Figure 27.37 Sample Smart Card Interface Reception Flowchart

27.6.8 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 27.38 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

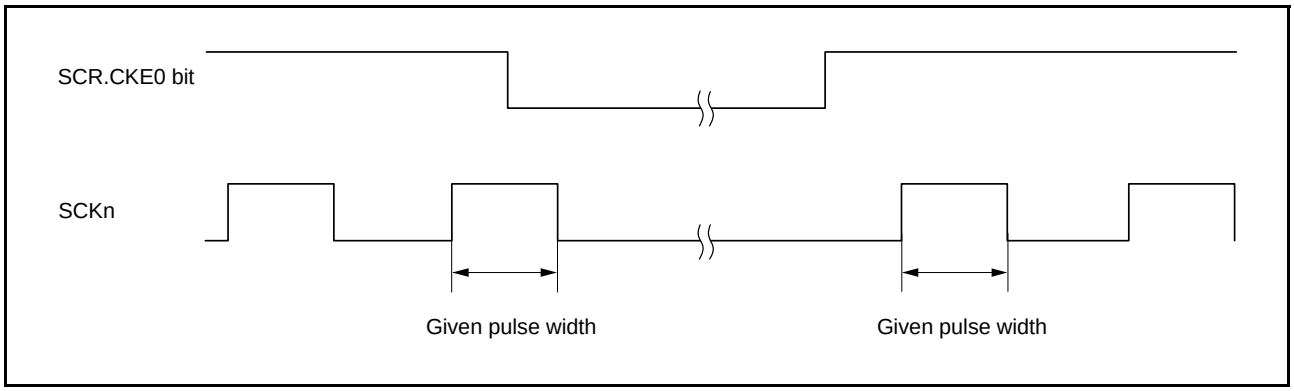


Figure 27.38 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, 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.

(2) At Mode Switching

(a) At transition from smart card interface mode to software standby mode

1. Set I/O port functions to make the SCKn pin fixed with a desired output value in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception.
   Simultaneously, set the SCR.CKE[1] bit to the value for the output fixed state in software standby mode.
3. Write 0 to the SCR.CKE[0] bit to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to the specified level with the duty cycle retained.
5. After switching the SCKn pin to the general I/O port function, make a transition to software standby mode.

(b) Return from software standby mode to smart card interface mode

6. Cancel software standby mode.
7. Set the SCR.CKE[0] bit to 1 to start clock output. A clock signal with the appropriate duty cycle is then generated.

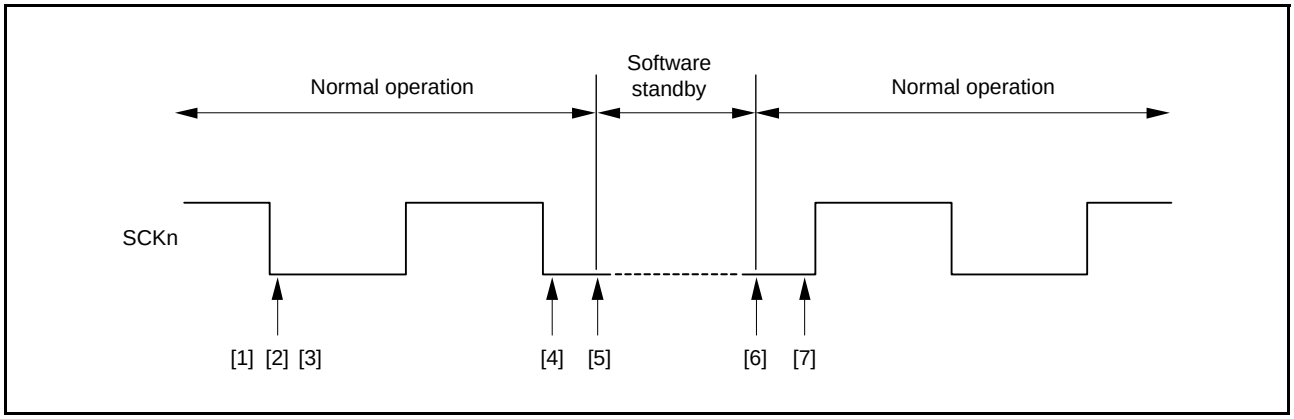


Figure 27.39 Clock Stop and Restart Procedure

27.7 Operation in Simple I2C Mode

Simple I2C bus master format is composed of eight 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 eight 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 27.40 and Figure 27.41.

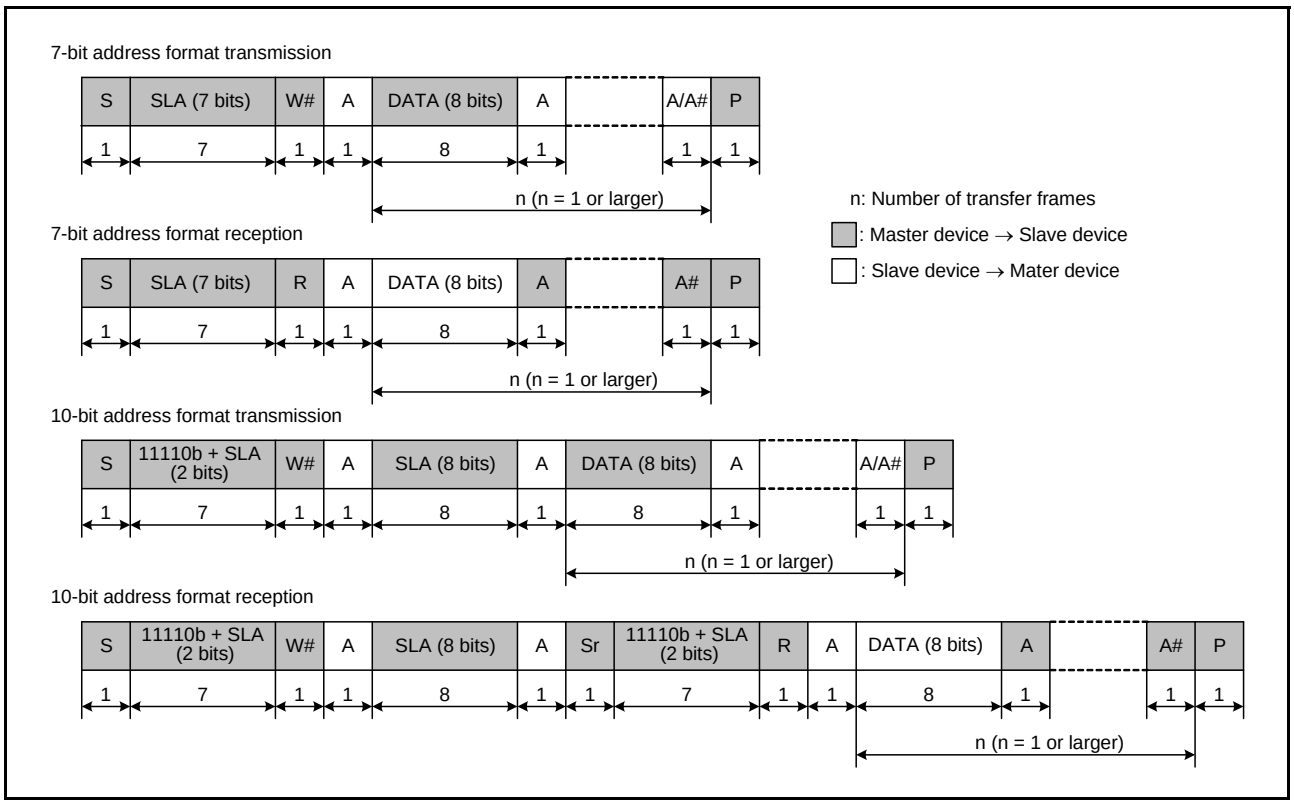


Figure 27.40 I2C Bus Master Format

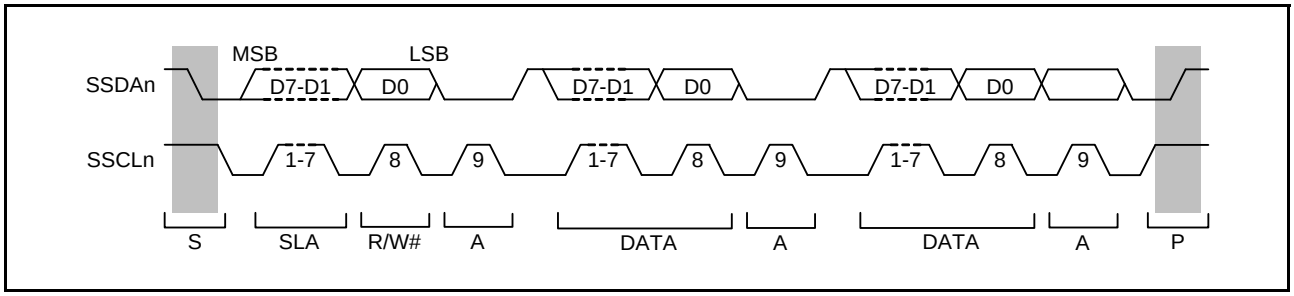


Figure 27.41 I2C Bus Timing (When SLA is 7 Bits)

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.

27.7.1 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 cleared (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 cleared (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 cleared (to 0), and a stop-condition generated interrupt is output.

Figure 27.42 shows the timing of operations in the generation of start, restart, and stop conditions.

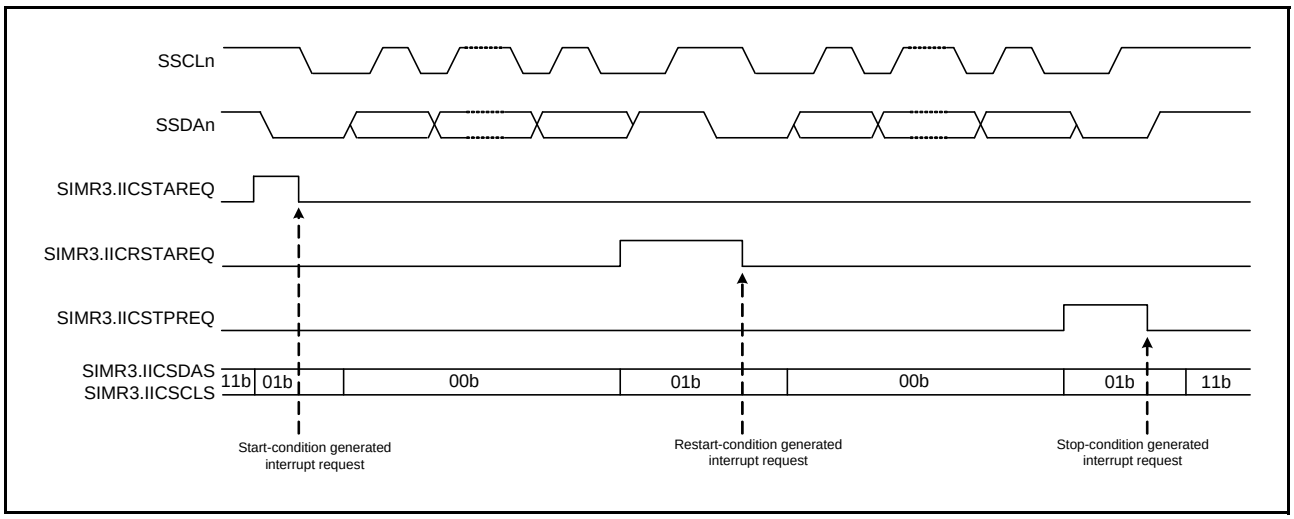


Figure 27.42 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

27.7.2 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 27.43 shows an example of operations to synchronize the clocks.

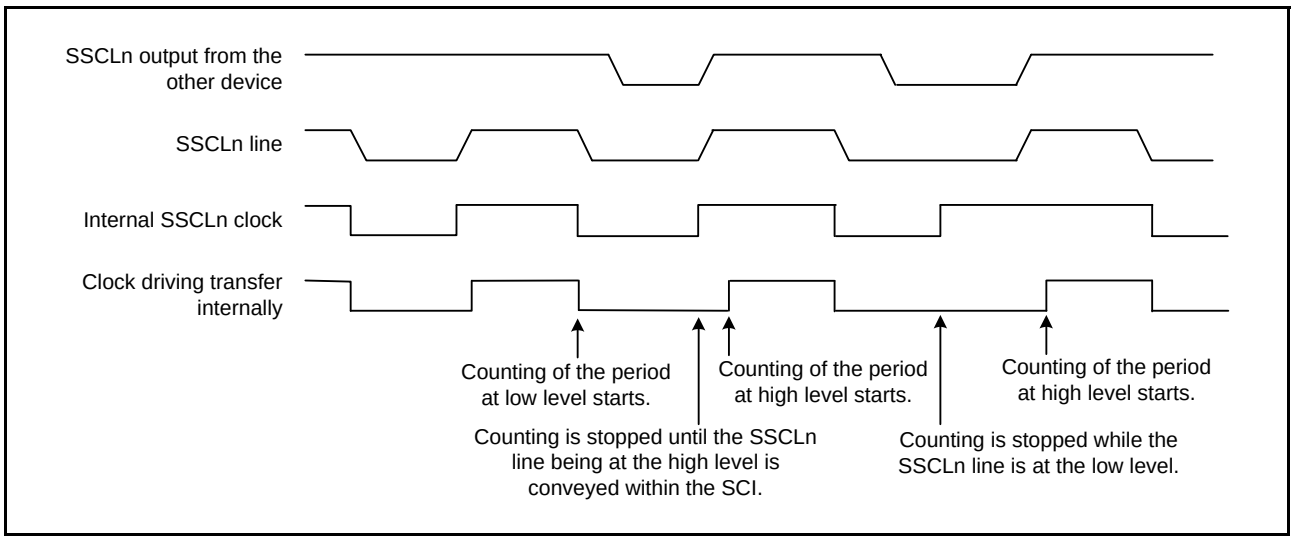


Figure 27.43 Example of Operations for Clock Synchronization

27.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SSDA 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 SSDA 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 27.44 shows the timing of delays in output on the SSDA pin.

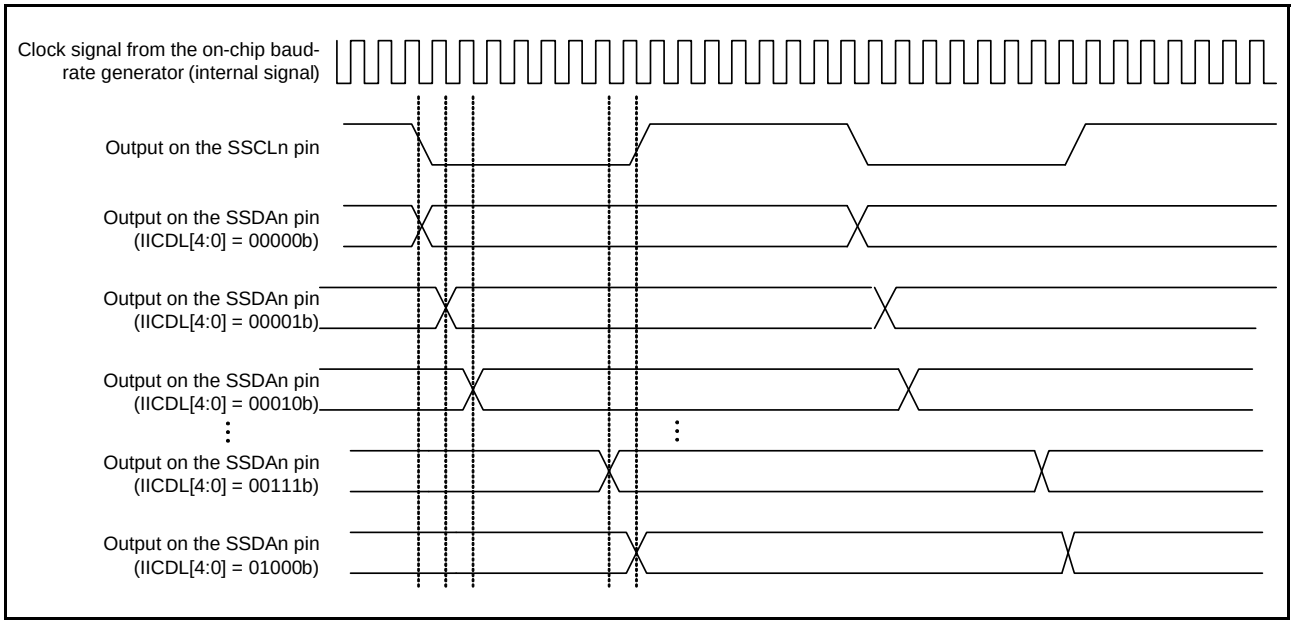


Figure 27.44 Timing of Delays in SSDA Output

27.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface in accordance with the flowchart shown as Figure 27.45.
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.

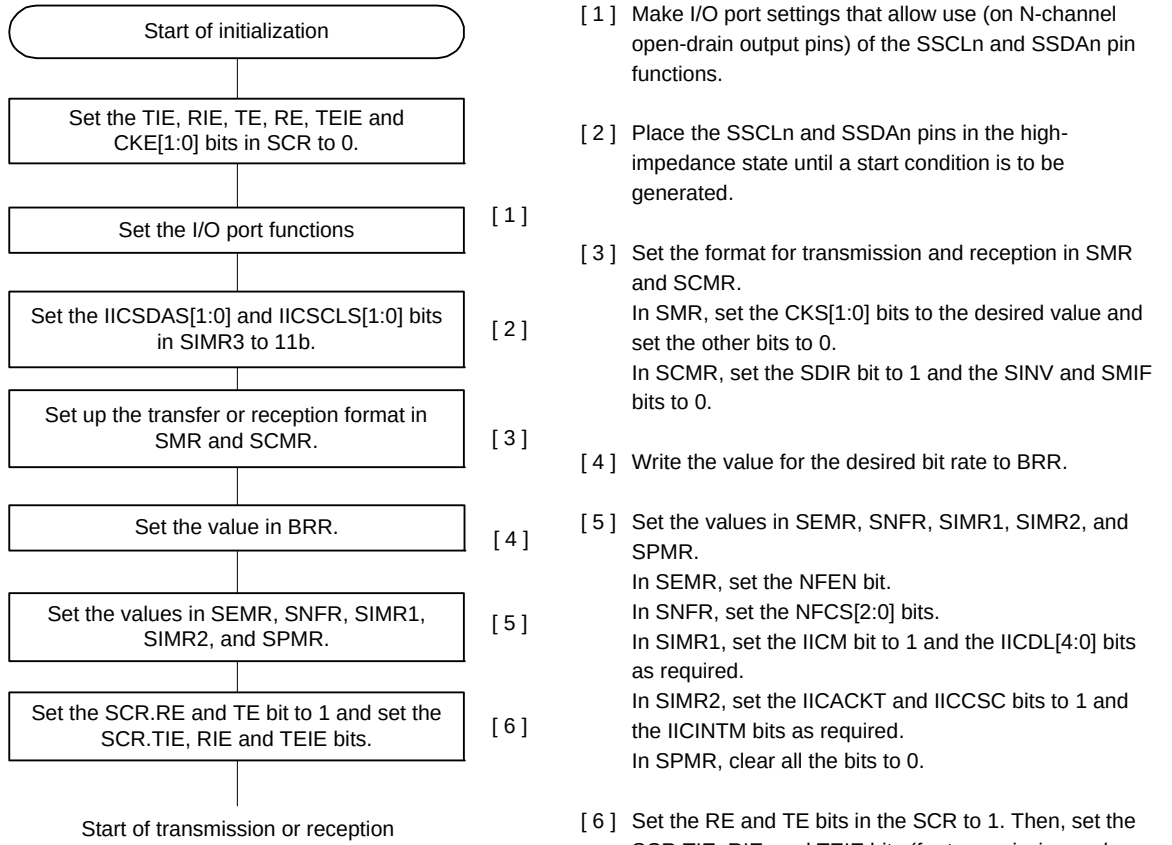


Figure 27.45 Example of the Flow of SCI Initialization (for Simple I2C Mode)

27.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 27.46 and Figure 27.47 show examples of operations in master transmission and Figure 27.48 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use) and the value of the SCR.RIE bit is assumed to be 0 (disabling reception interrupt requests). See Table 27.28 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 27.48 are repeated twice.
In simple I2C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.

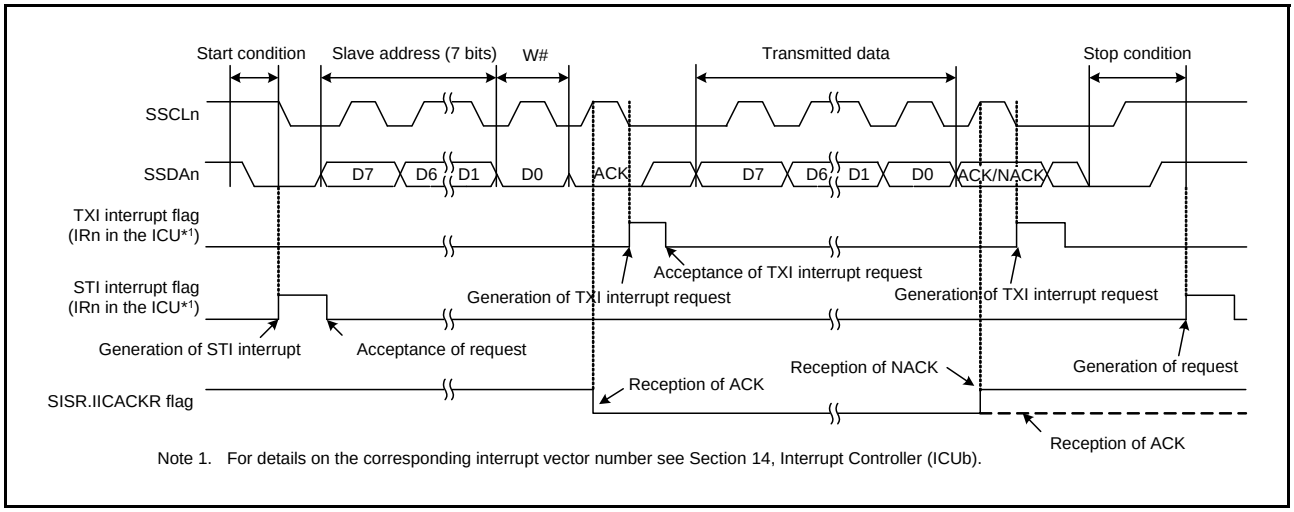


Figure 27.46 Example of Operations for Master Transmission in I2C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (ACK and NACK interrupts are used) 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 re-transmission, is performed by the NACK interrupt as the trigger.

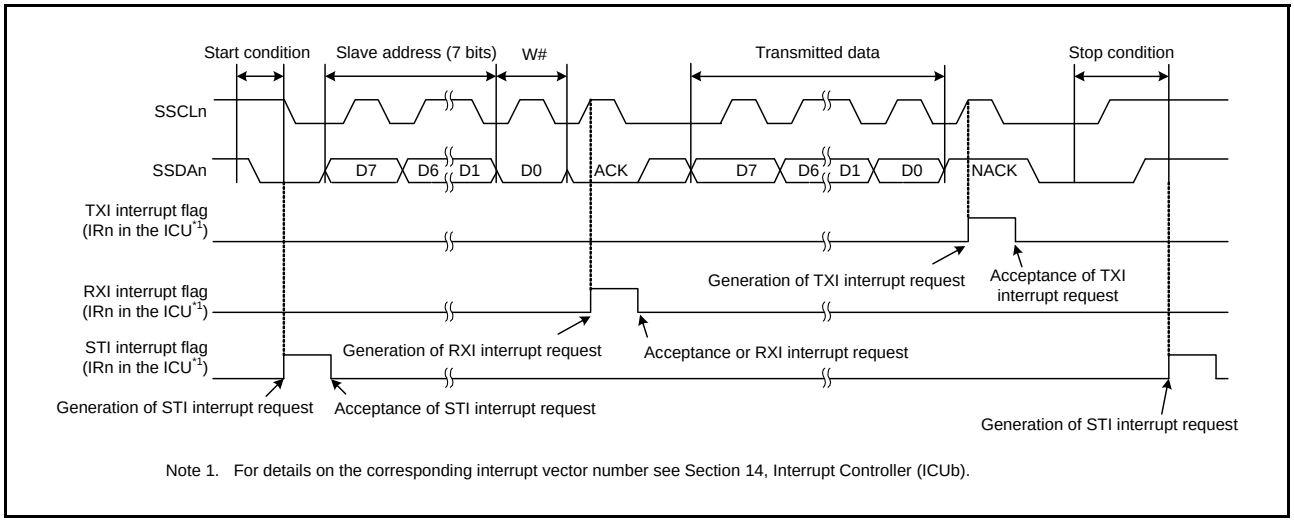


Figure 27.47 Example 2 of Operations for Master Transmission in Simple I2C Bus Mode (with Seven-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)

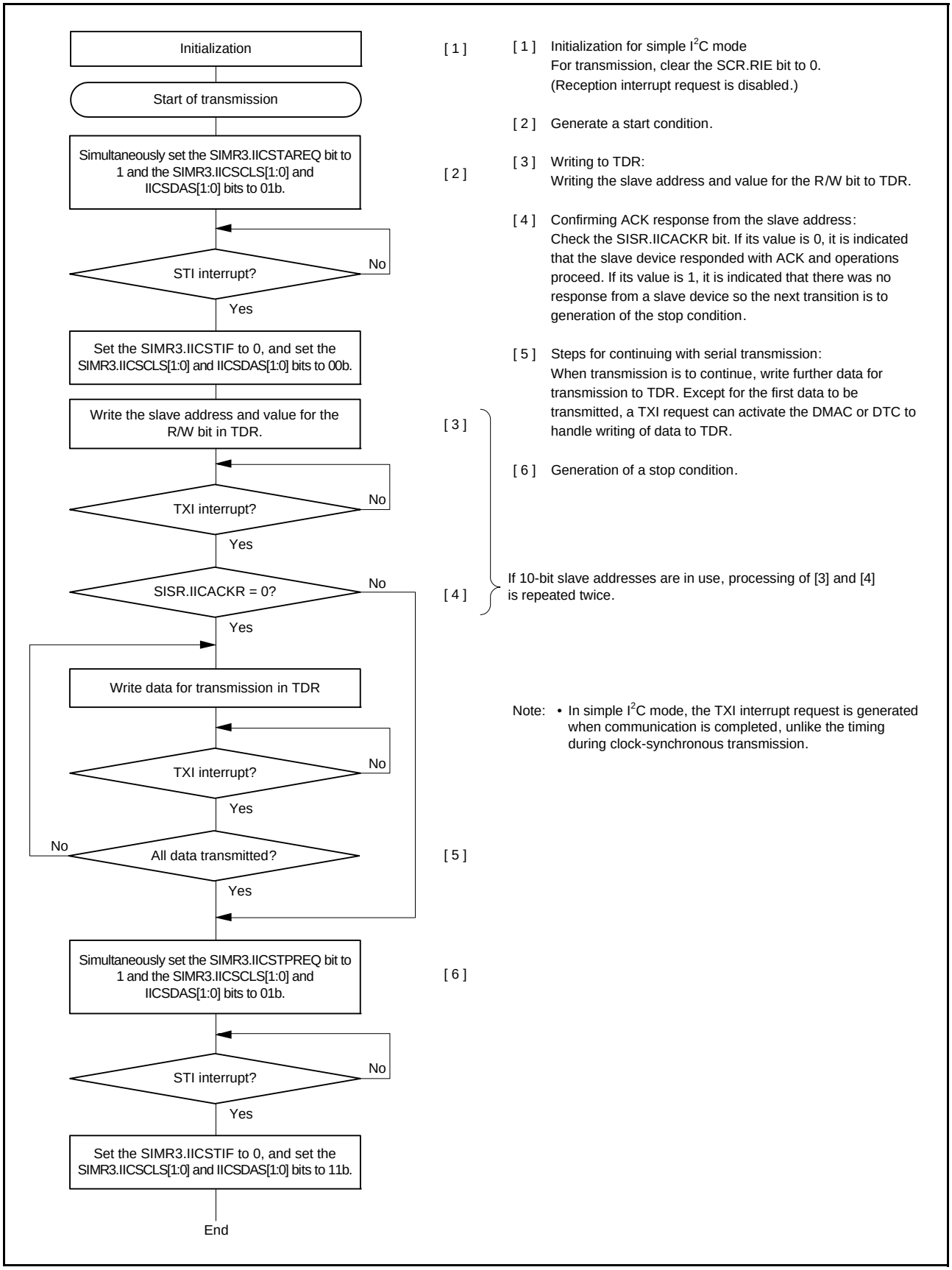
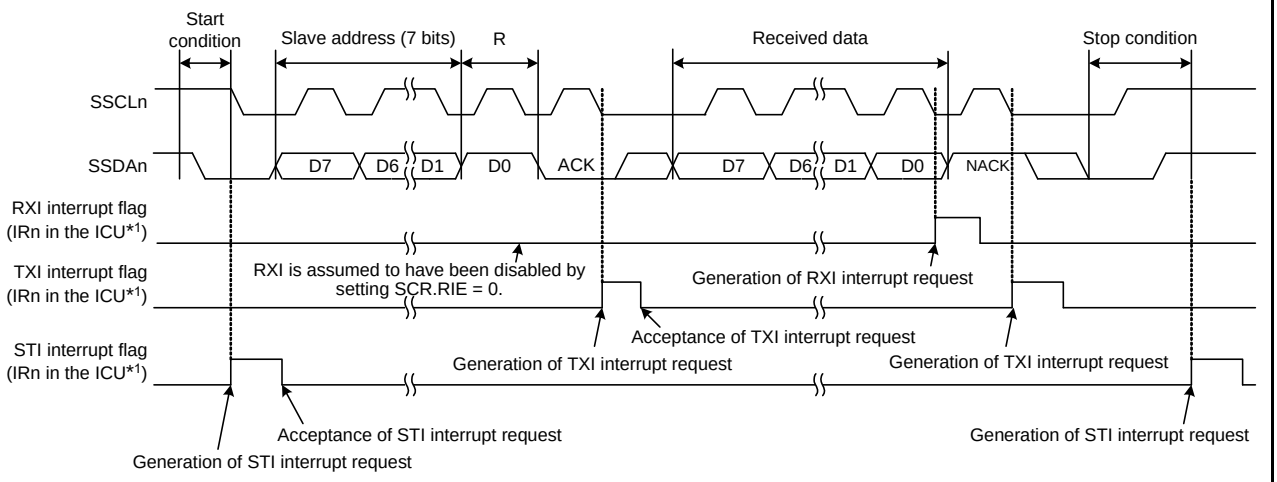


Figure 27.48 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

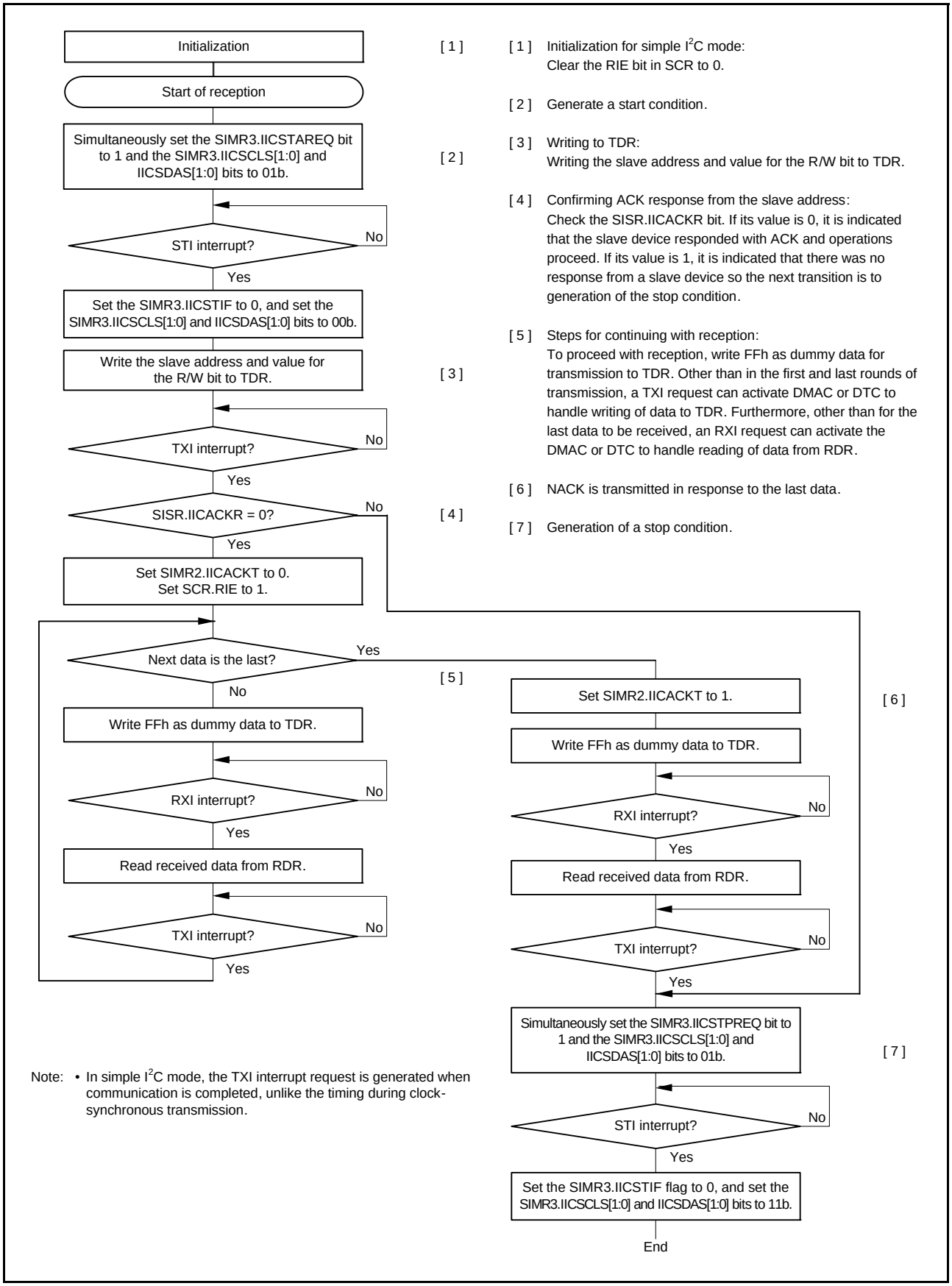
27.7.6 Master Reception (Simple I²C Mode)

Figure 27.49 shows an example of operations in simple I²C mode master reception and Figure 27.50 is a flowchart showing the procedure for master reception.
The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use).
In simple I²C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.



Note 1. For details on the corresponding interrupt vector number, see Section 14, Interrupt Controller (ICUb).

Figure 27.49 Example of Operations for Master Reception in I²C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Clear 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 data for transmission 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 27.50 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

27.8 Operation in Simple SPI Mode

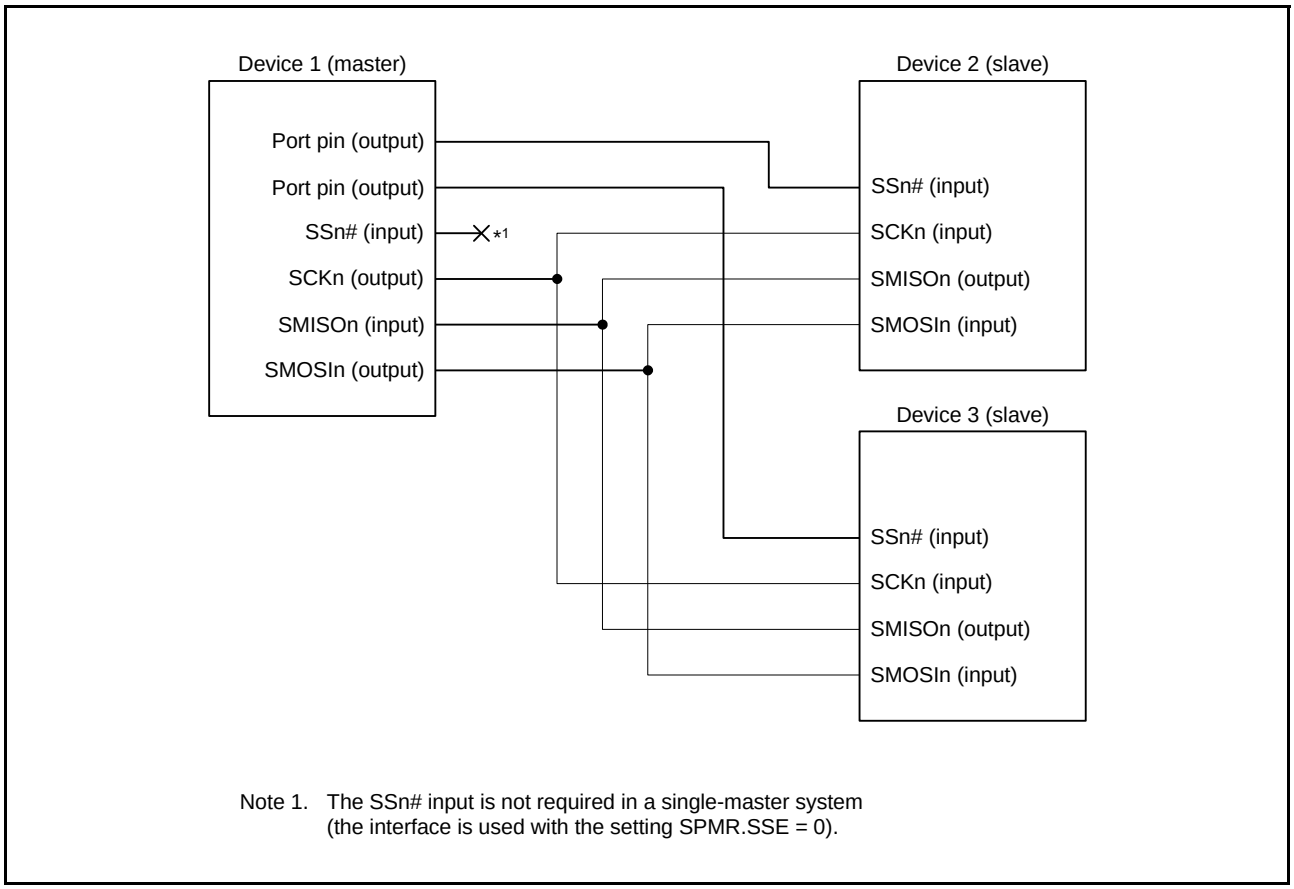
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

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 27.51 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 eight 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 reception and transmission sections are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both sections have a buffered structure, writing of further data for transmission 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 27.51 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

27.8.1 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 27.24 lists the states of pins according to the mode and the level on the SSn# pin.

Table 27.24 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 transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

27.8.2 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 output pins 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.

27.8.3 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.

27.8.4 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 27.52. 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, see section 27.8.2, SS Function in Master Mode.

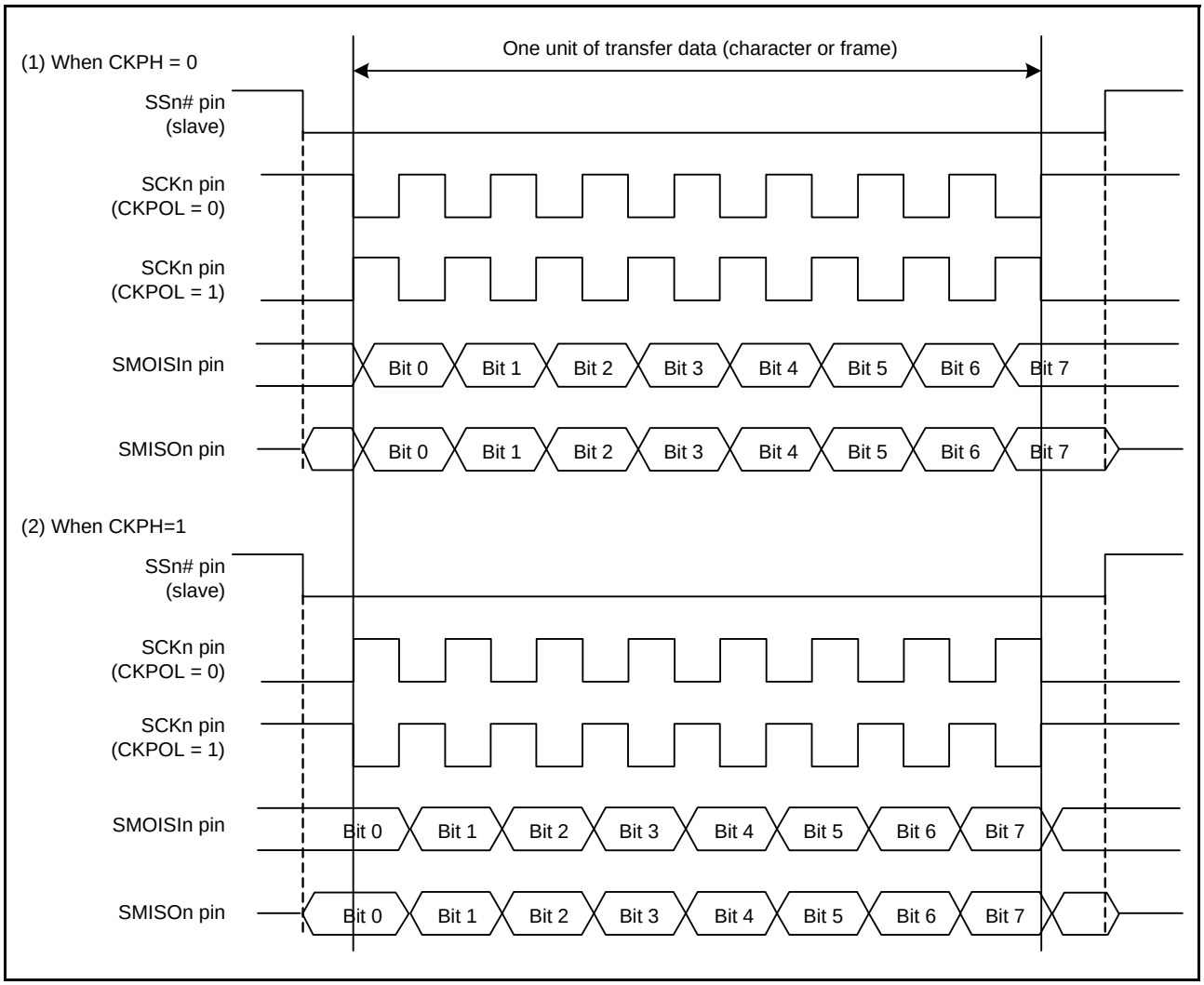


Figure 27.52 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

27.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock-synchronous mode Figure 27.21, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR must be set to endure 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 transmission interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

27.8.6 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.

27.9 Extended Serial Mode Control Section: Description of Operation

27.9.1 Serial Transfer Protocol

In conjunction with an SCIe module, the extended serial mode control section of an SCIf module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 27.53.
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.

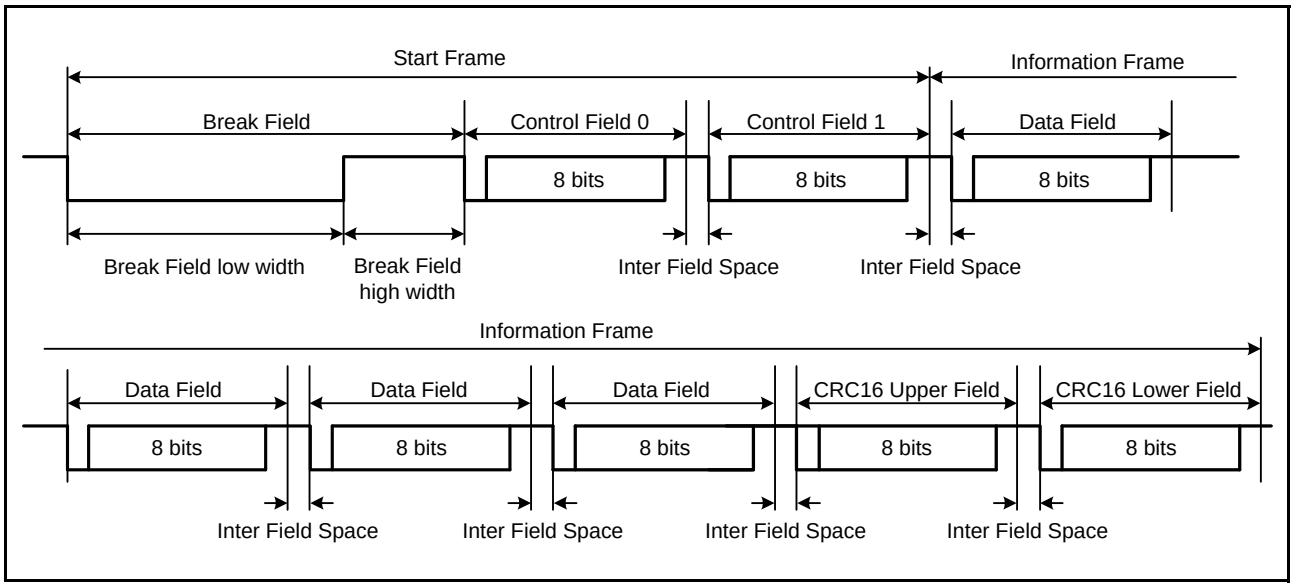


Figure 27.53 Protocol for Serial Transfer by the Extended Serial Mode Control Section

27.9.2 Transmitting a Start Frame

Figure 27.54 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 27.55 and Figure 27.56 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.

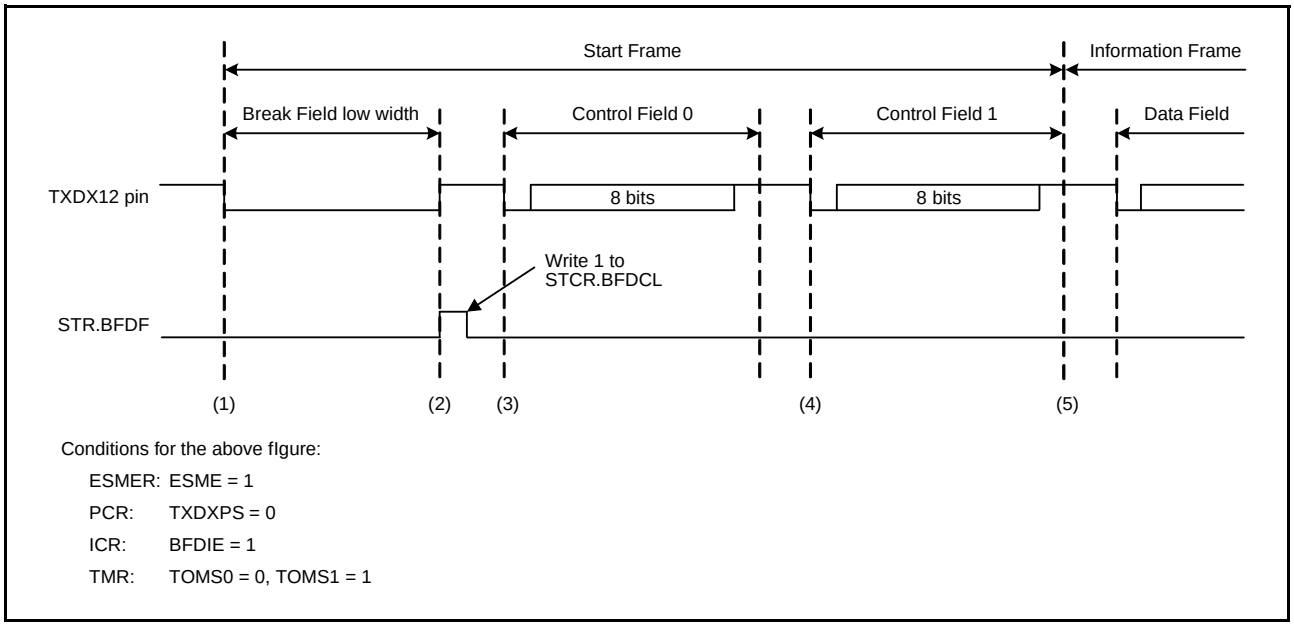


Figure 27.54 Example of Operations at the Time of Start-Frame Transmission

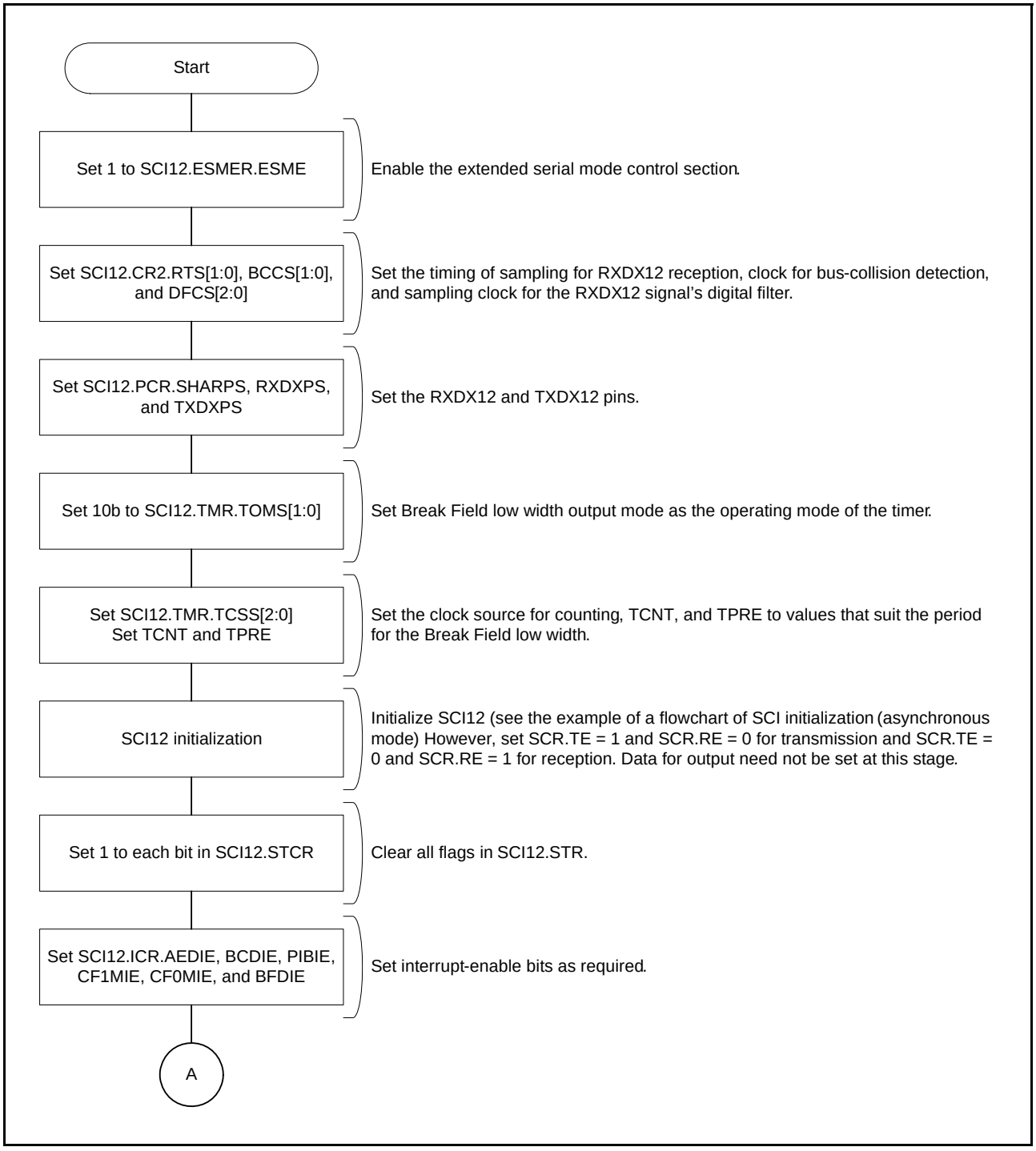


Figure 27.55 Sample Flowchart for Transmission of a Start Frame (1)

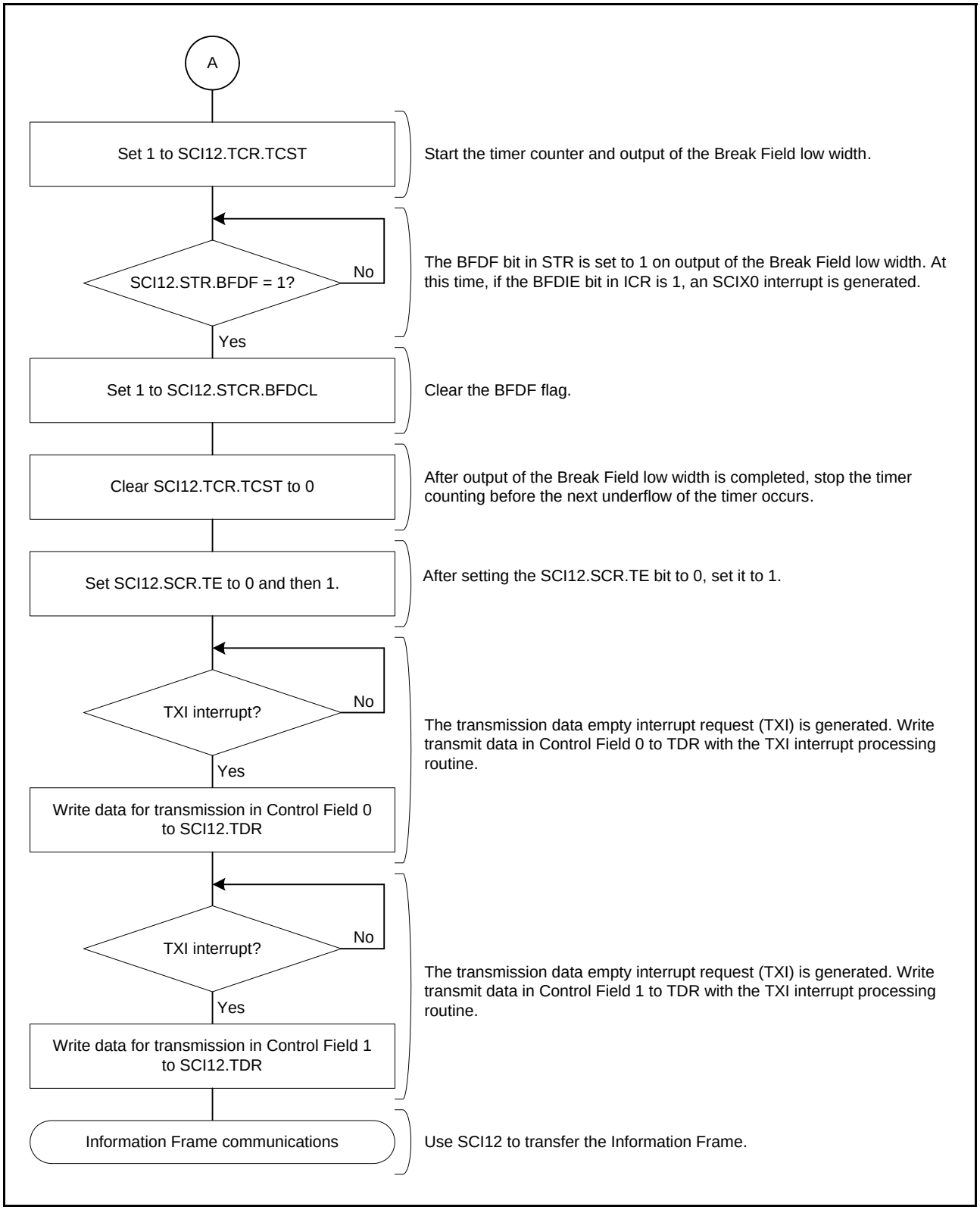


Figure 27.56 Sample Flowchart for Transmission of a Start Frame (2)

27.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 27.25.

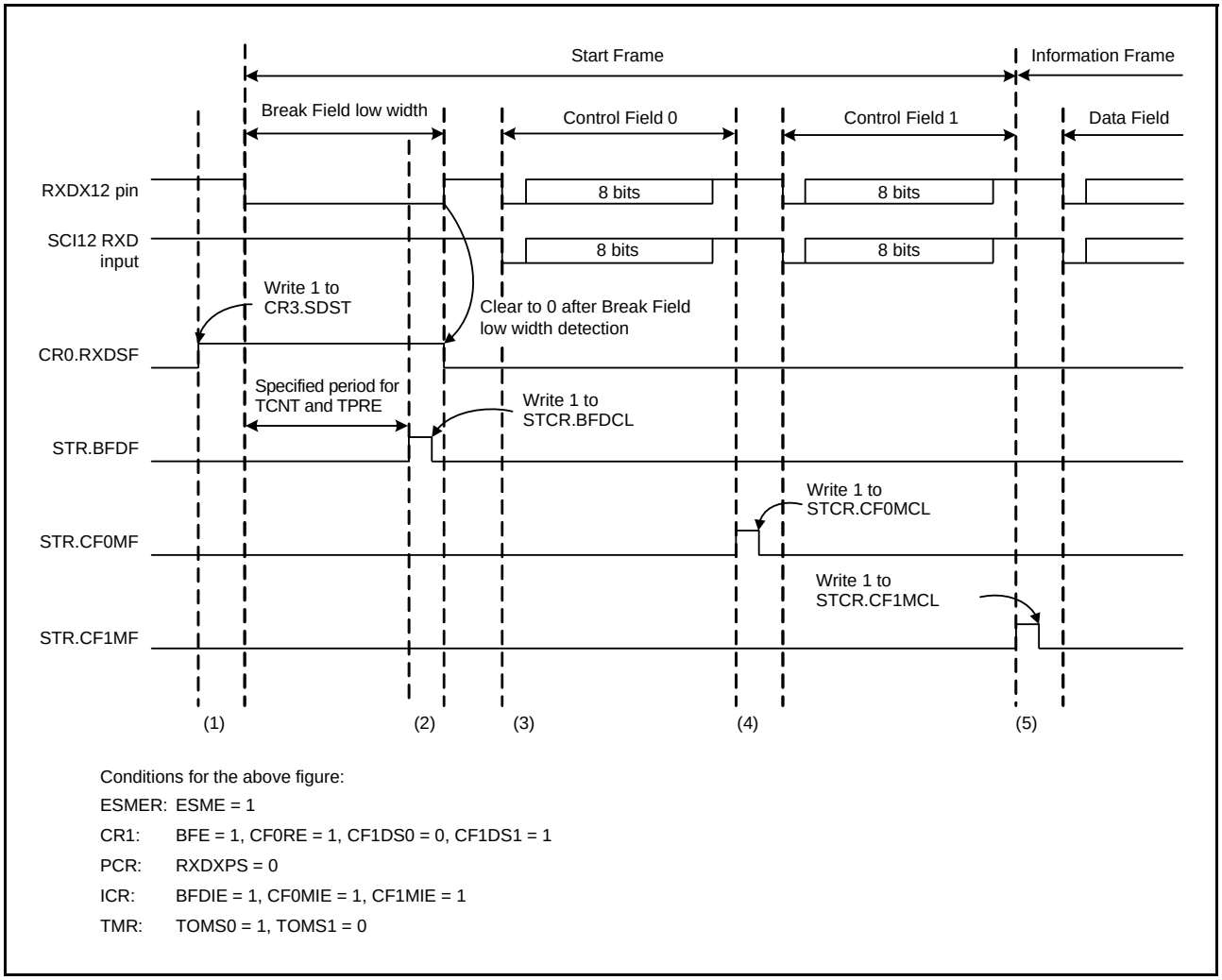
Table 27.25 Structures of Start Frames

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 27.57 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 27.58 and Figure 27.59 are flowcharts for the reception of a Start Frame, and Figure 27.60 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 module 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 zero and reception of Control Field 0 by the SCI12 module 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 module 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 module 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.



Conditions for the above figure:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS0 = 0, CF1DS1 = 1

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS0 = 1, TOMS1 = 0

Figure 27.57 Example of Operations at the Time of Start-Frame Reception

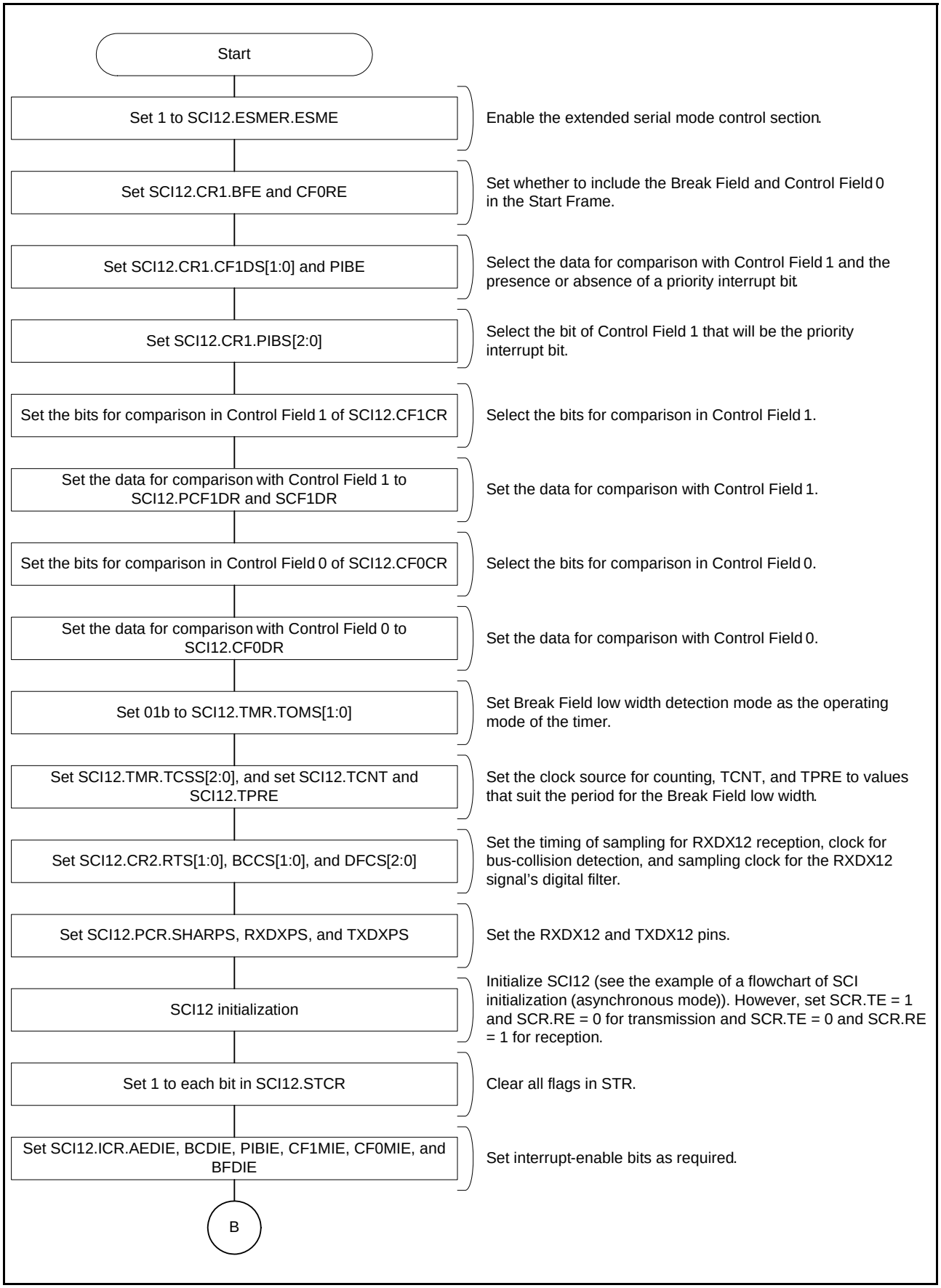


Figure 27.58 Sample Flowchart for Reception of a Start Frame (1)

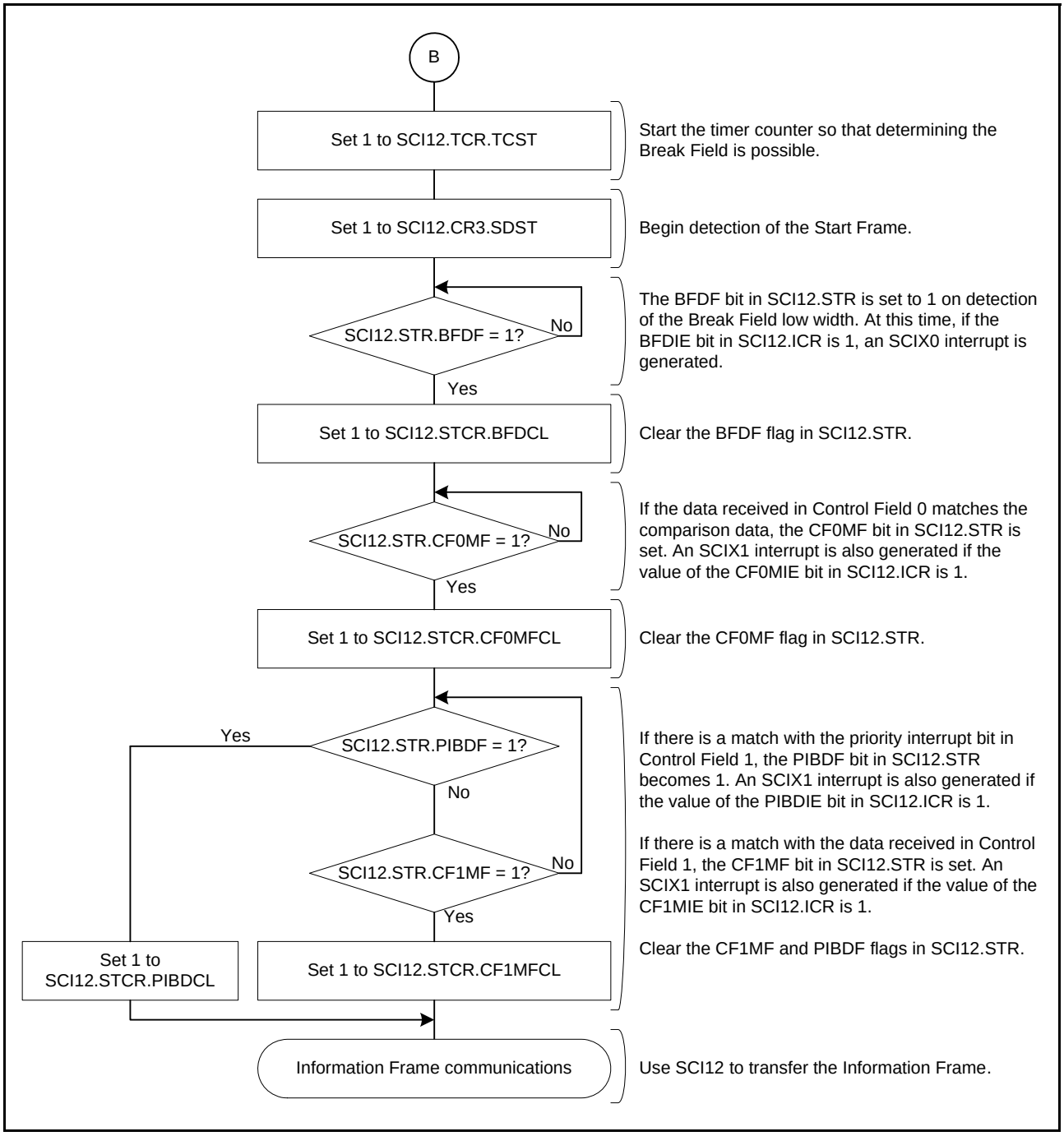


Figure 27.59 Sample Flowchart for Reception of a Start Frame (2)

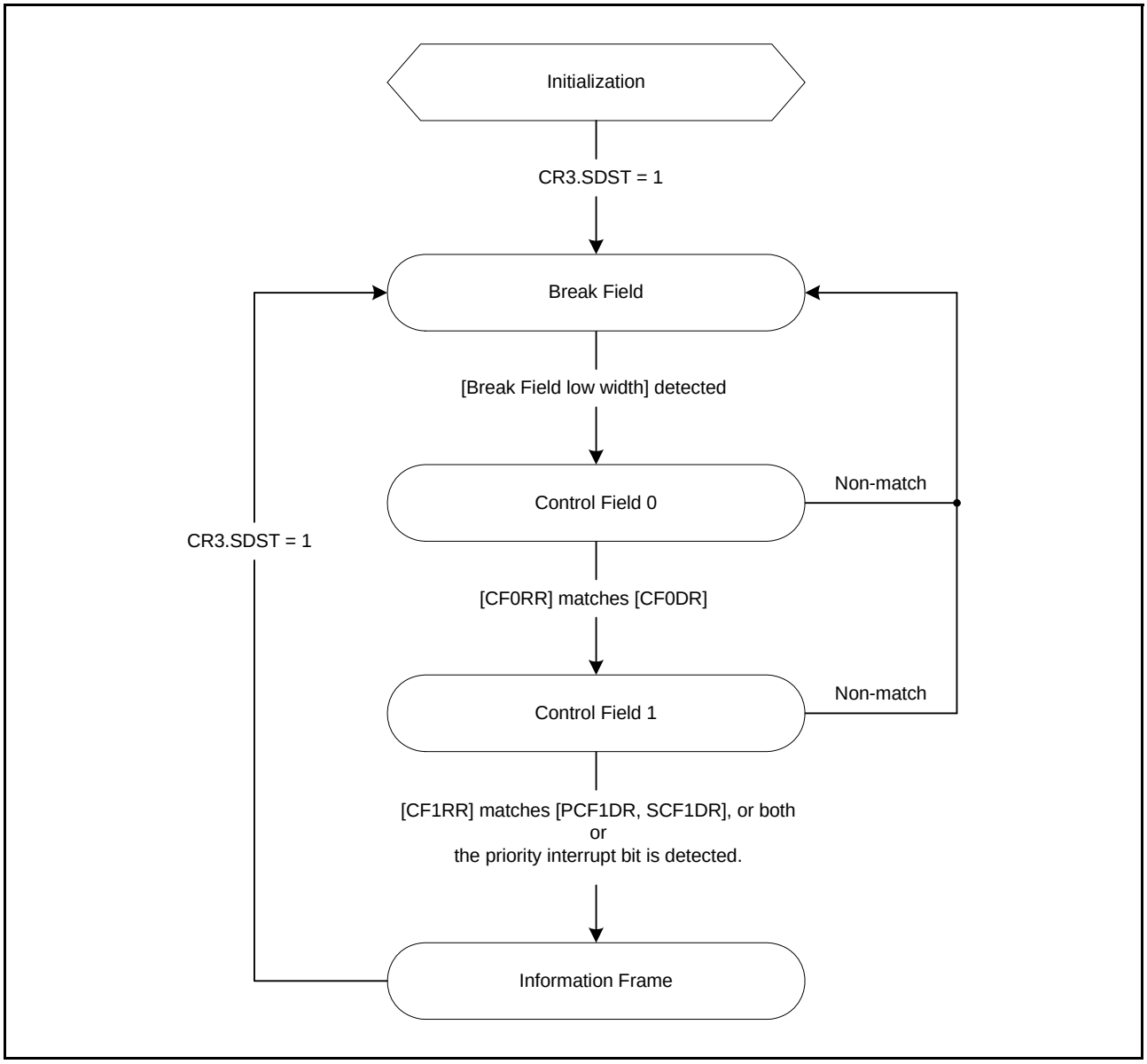


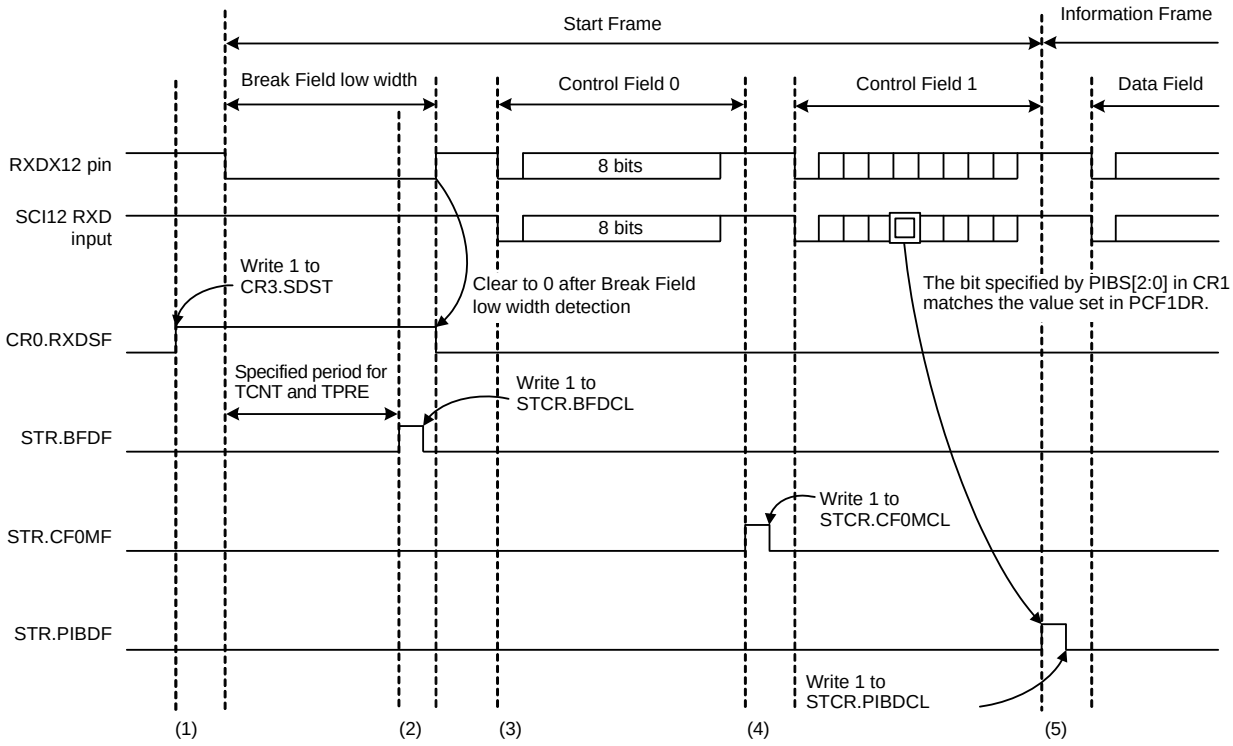
Figure 27.60 Diagram of State Transitions at the Time of Start-Frame Reception

27.9.3.1 Priority Interrupt Bit

Figure 27.61 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 27.57, 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 module 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.



Conditions for the above figure:
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: TOMS0 = 1, TOMS1 = 0

Figure 27.61 Example of Operations at the Time of Start-Frame Reception (with Priority Interrupts in Use)

27.9.4 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 module are in progress when the ESMER.ESME and the SCI12.SCI. TE are set to 1.

Figure 27.62 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.

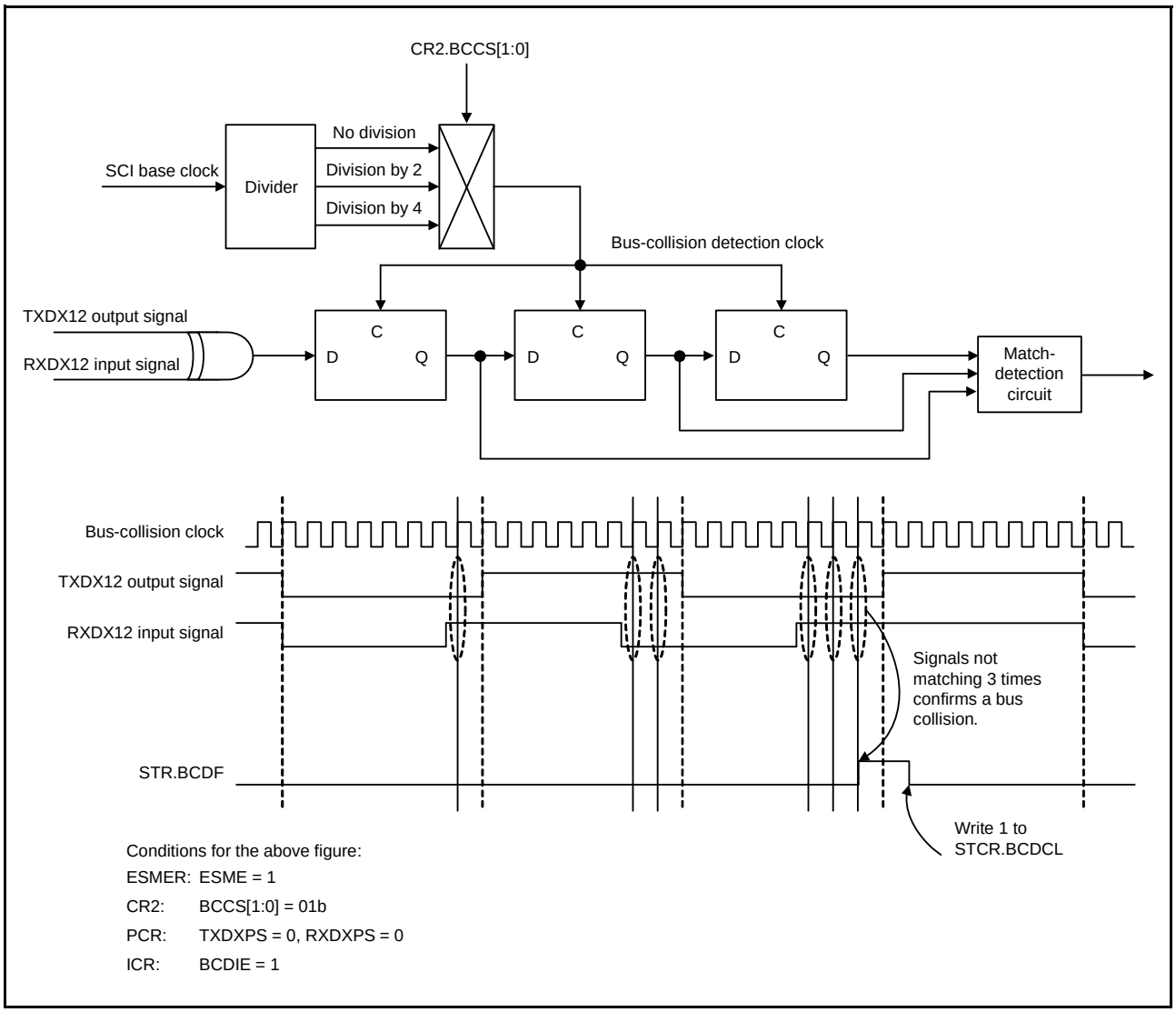


Figure 27.62 Example of Operations with Bus-Collision Detection

27.9.5 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 27.63 shows an example of operations with the digital filter.

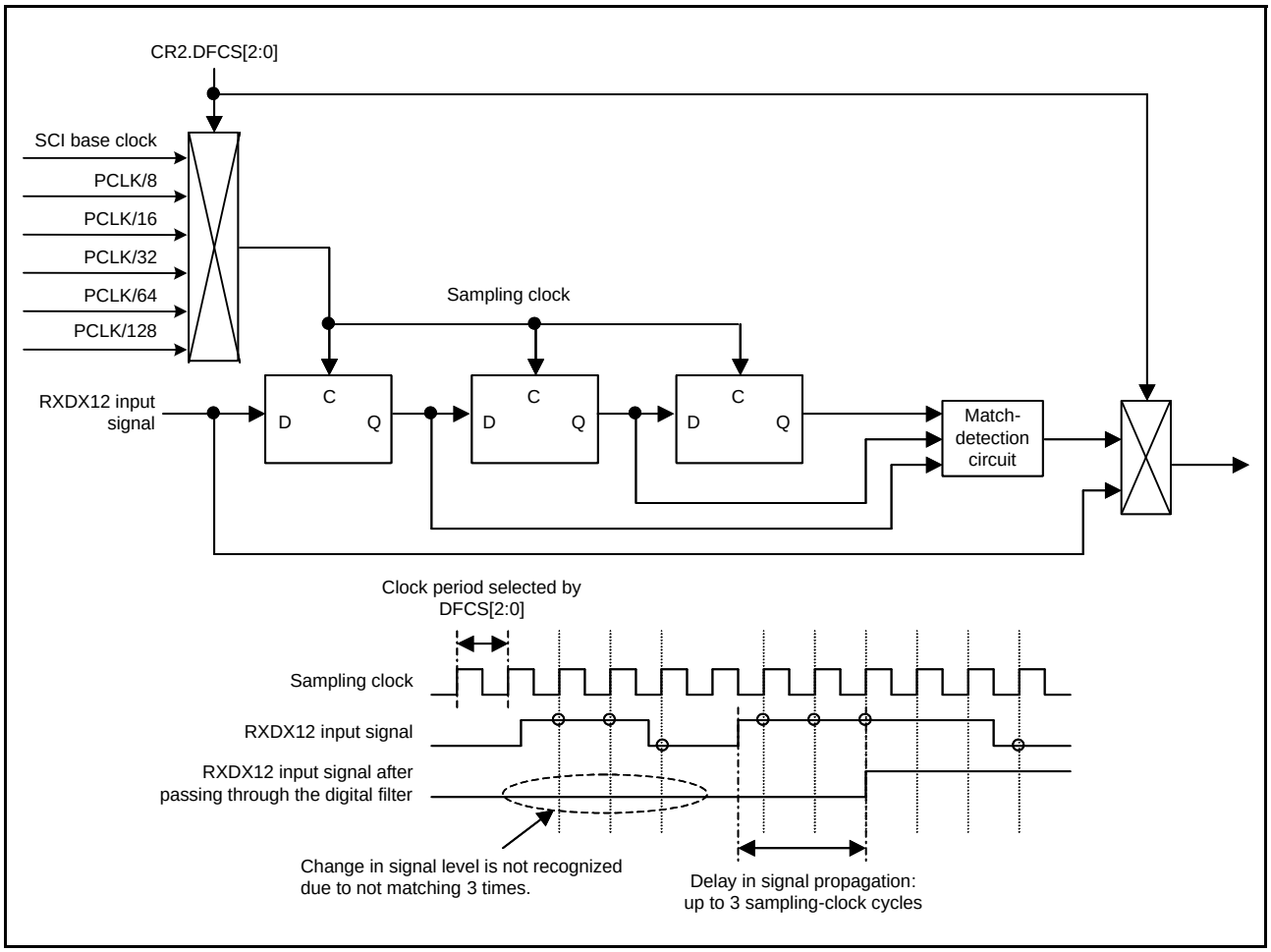
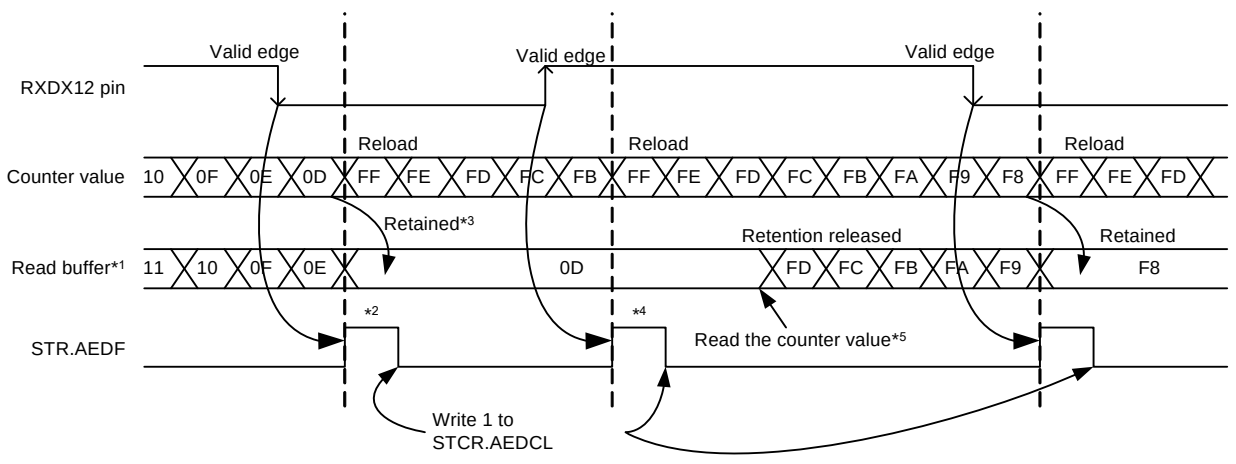


Figure 27.63 Example of Operations with the Digital Filter

27.9.6 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 27.64 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 module. 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 27.64 Example of Operations for Bit-Rate Measurement

27.9.7 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 module by setting the RTS0 and RTS1 bits in CR2 to select the rising edges of 8, 10, 12, or 14 cycles of 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 module. Figure 27.65 shows timing for the sampling of data received through RXDX12.

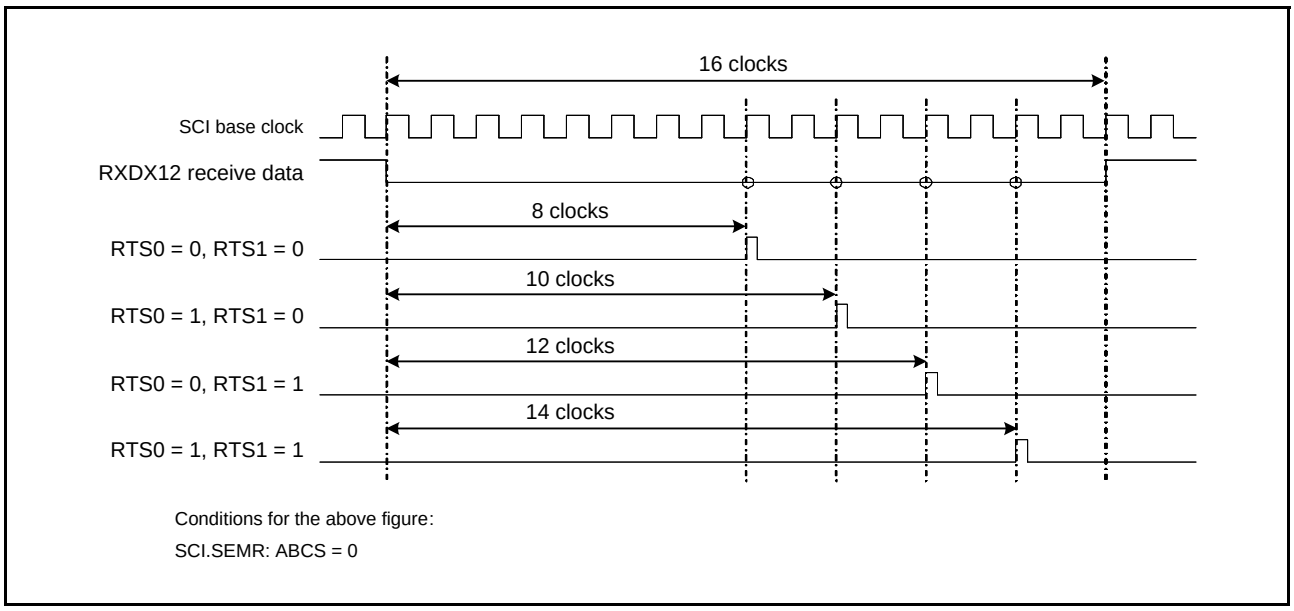


Figure 27.65 Timing for Sampling of Data Received through RXDX12

27.9.8 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 time of transmitting a Start Frame. Setting TOMS0 to 0 and TOMS1 to 1 in TMR 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 27.66 shows an example of operations in Break Field low width output mode.

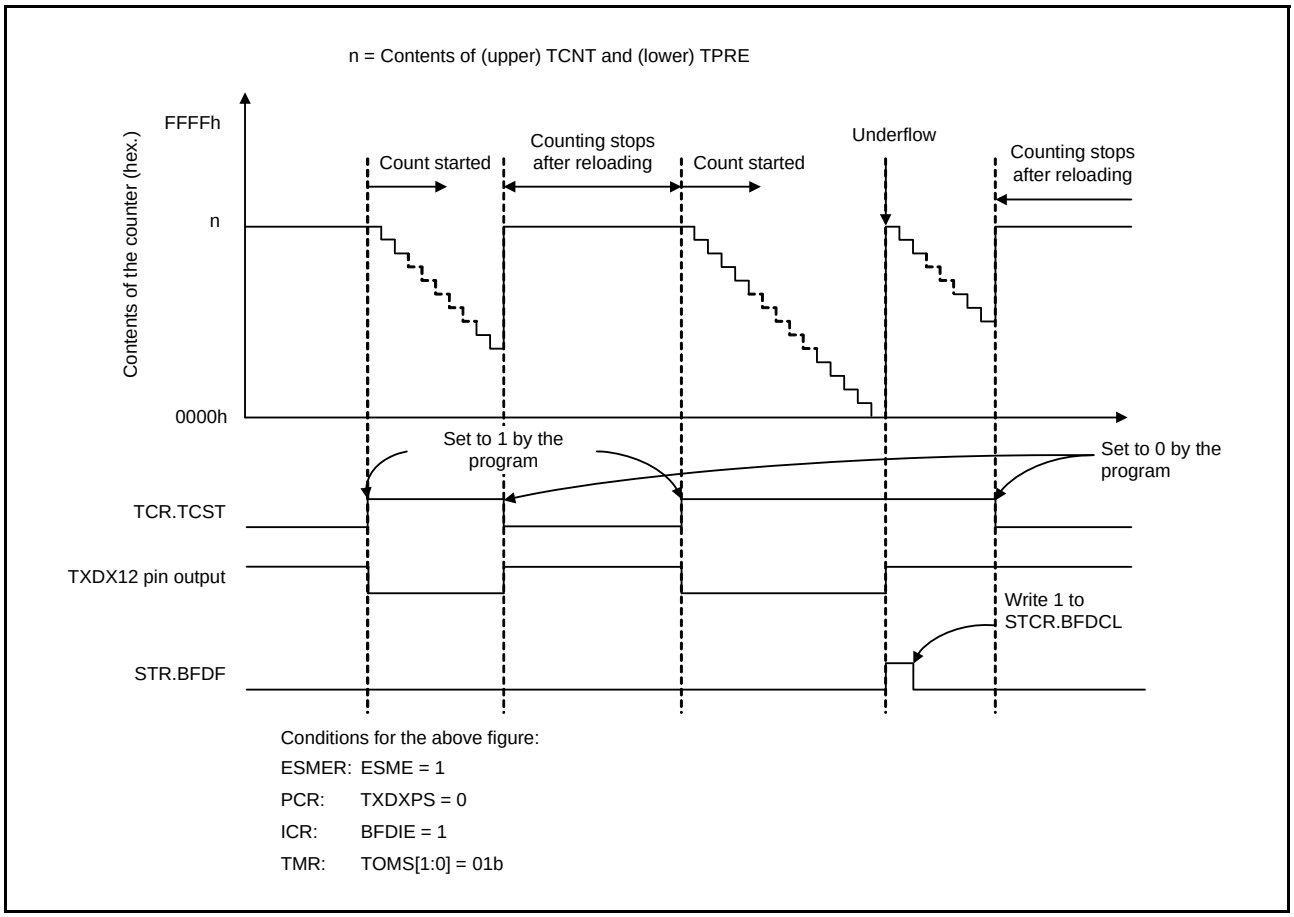


Figure 27.66 Example of Operations in Break Field Low Width Output Mode

(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 time of receiving 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 27.67 shows an example of operations in Break Field low width output mode.

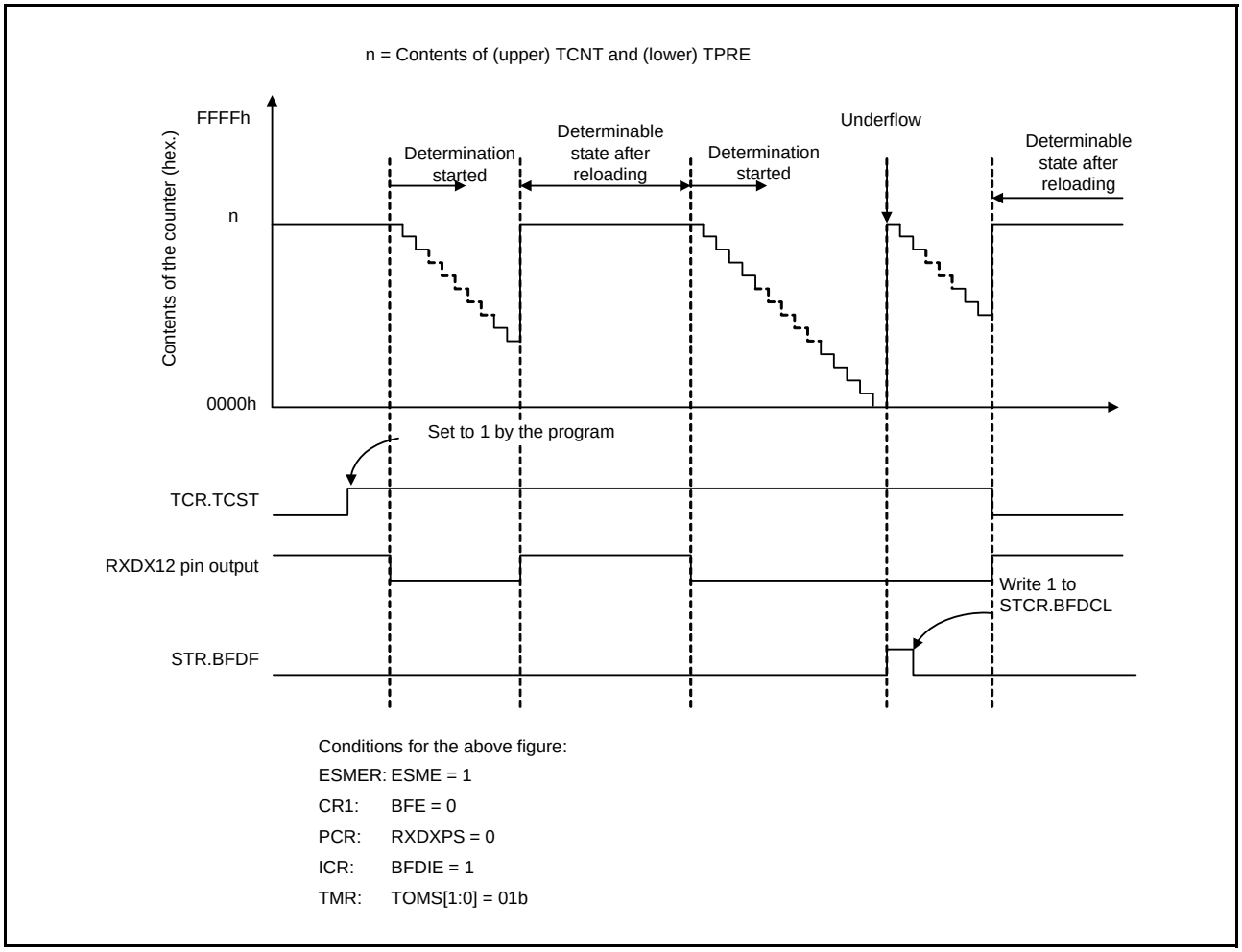


Figure 27.67 Example of Operations in Break Field Low Width Determination Mode

(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 counting. 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.

27.10 Noise Cancellation Function

Figure 27.68 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.

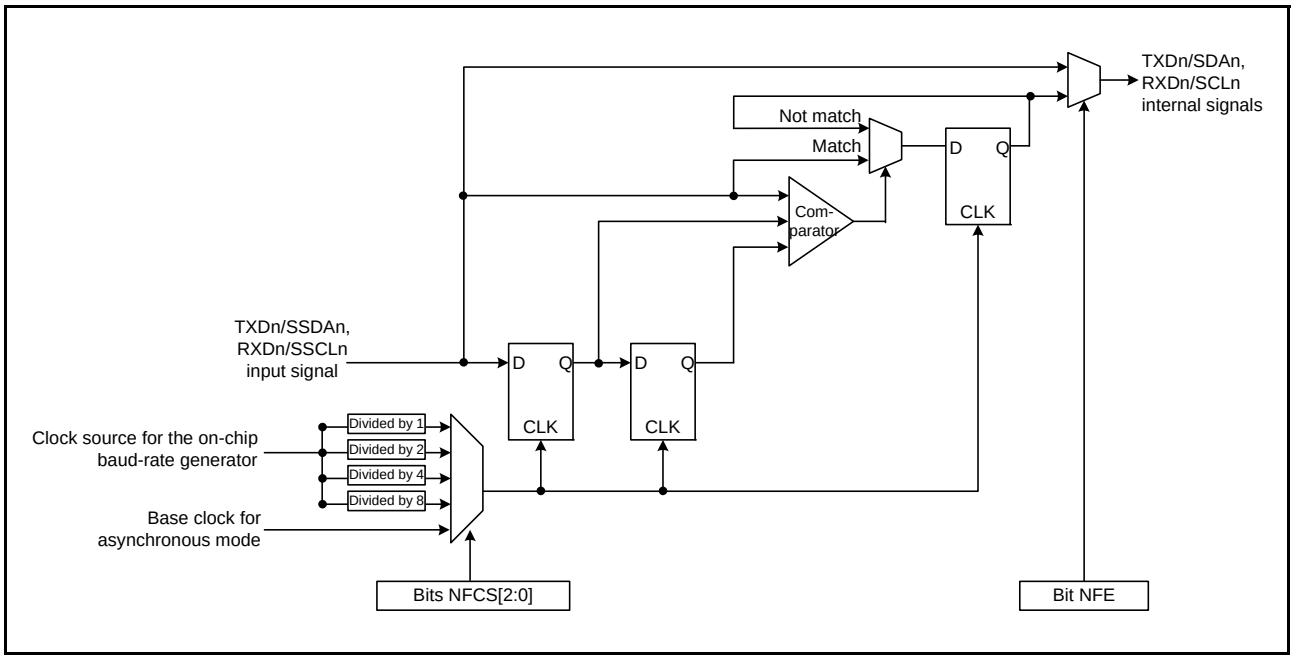


Figure 27.68 Block Diagram of Digital Noise Filter Circuit

27.11 Interrupt Sources

27.11.1 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.

27.11.2 Interrupts in Serial Communications Interface and Simple SPI Mode

Table 27.26 lists interrupt sources in serial communication interface 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 data for transmission are 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 are not written by the time of transmission of the last bit of the current data for transmission 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 data for transmission 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 are 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 27.26 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

27.11.3 Interrupts in Smart Card Interface Mode

Table 27.27 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 27.27 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 cleared to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically re-transmits 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 or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, see section 16, DMA Controller (DMACA) and section 17, 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.

27.11.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 27.28. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC and 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 SSCL 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 data for transmission.

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 27.28 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 generating 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 (selecting reception and interrupts).

27.11.5 Interrupts 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 27.29.

Table 27.29 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

27.12 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 (reception 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) Received-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 eighth-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 received-data full events are not used.

(3) Transmission-data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that data for transmission 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) Transmission-completed 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.

27.13 Usage Notes

27.13.1 Setting the Module Stop Function

Module stop control registers B and C (MSTPCRB and MSTPCRC) are used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. The registers of the modules only become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

27.13.2 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), and the PER flag in SSR may also be set to 1 (parity error). When the SEMR.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is cleared to 0 (no framing error), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI stops the receive operation while waiting for the start bit of the next frame to be detected, with the SSR.FER flag set to 1. If the SSR.FER flag is set to 0 at this time, this flag remains 0 during a break. After the RXDn pin is pulled high and the break ends, the SCI detects the beginning of the start bit at the first falling edge of the RXDn pin and starts the receive operation.

27.13.3 The Mark State and Production of Breaks

When the SCR.TE bit is 0 (prohibiting serial transmission), 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 (permitting serial transmission), 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 transfer circuit 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 transmission section is initialized regardless of the current state of transmission.

27.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

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 clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception disabled).

27.13.5 Writing Data to TDR

Data can always 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 processing routine.

27.13.6 Restrictions on Clock Synchronous Transmission

When the external clock source is used as a synchronization clock, update TDR by the DMAC or DTC and wait for at least five PCLK clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR is updated, the SCI may malfunction.

27.13.7 Restrictions on Using DTC or DMAC

When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt (RXI) as the activation source of the relevant SCI.

27.13.8 Points to Note on Starting Transfer

At the point where transfer starts when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1).

- 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 actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

27.13.9 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. Clearing the TE bit to 0 resets the TSR and the TEND bit in the SSR. Depending on the port settings, output pins may output the high level after release from the module-stopped state or software standby. 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 27.69 shows a sample flowchart for transition to software standby mode during transmission. Figure 27.70 and Figure 27.71 show the port pin states during transition to software standby mode.

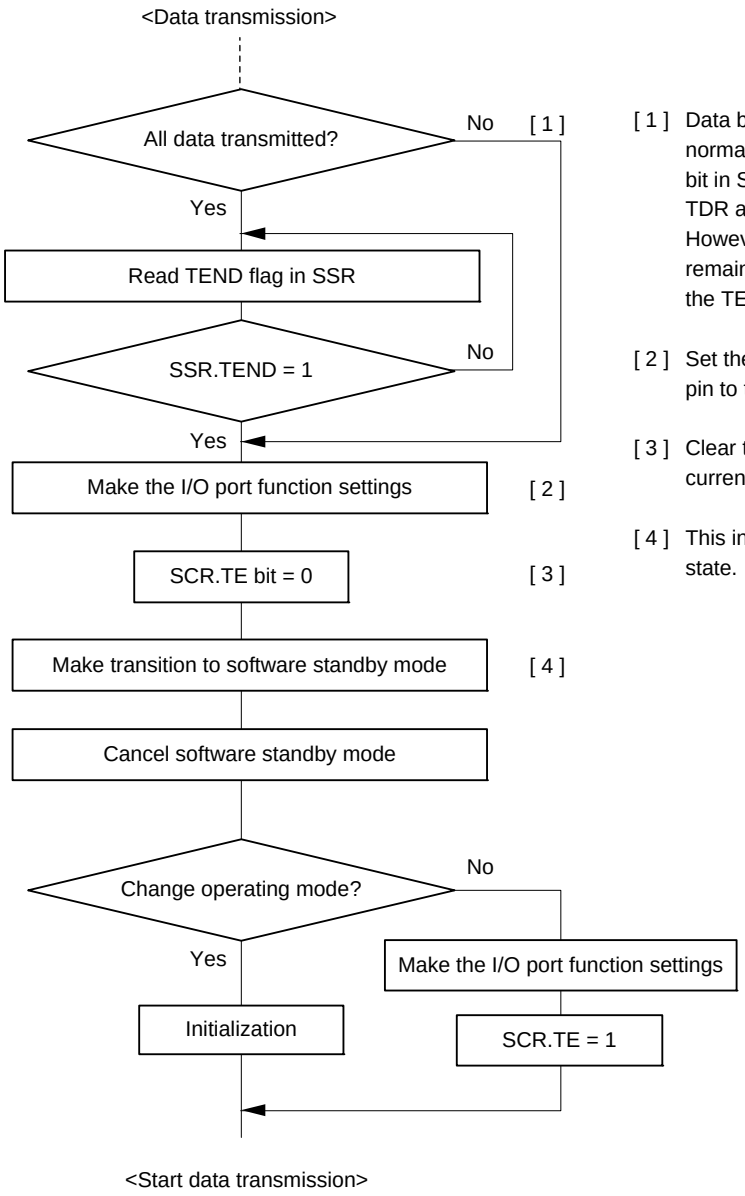
Before specifying the module stop state or making a transition to software standby mode from the transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The TXI interrupt flag is set to 1 and transmission 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 27.72 shows a sample flowchart for transition to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the TE bit in SCR to 1, reading SSR, and writing data to TDR after canceling software standby mode. However, if the DTC has been activated, the data remaining in the DTC will be transmitted when both the TE and TIE bits in SCR are set to 1.

[2] Set the I/O port function, and switch from the TXDn pin to the general I/O port.

[3] Clear the TIE and TEIE bits in the SCR if they are currently set to 1.

[4] This includes the setting for the module-stopped state.

Figure 27.69 Example of Flowchart for Transition to Software Standby Mode during Transmission

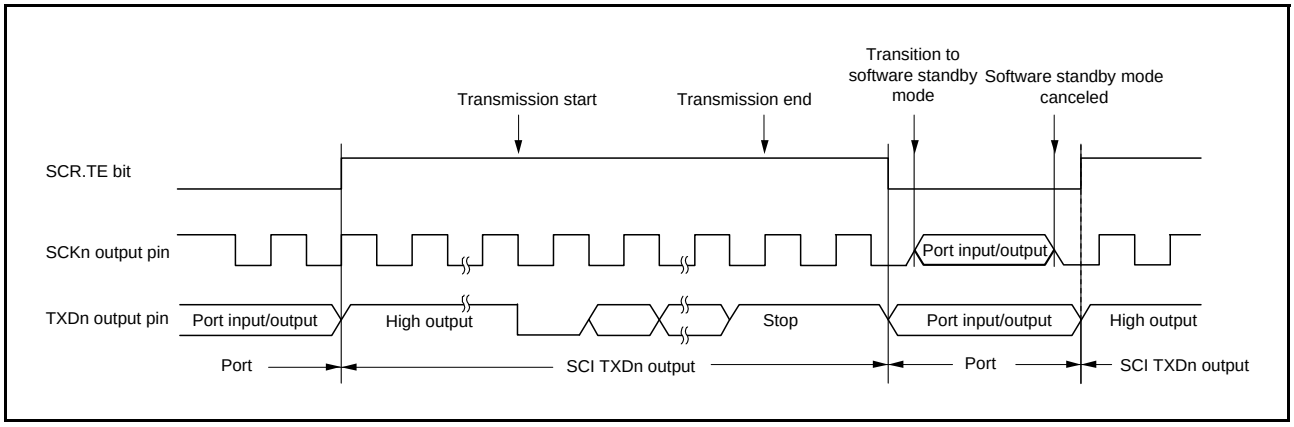
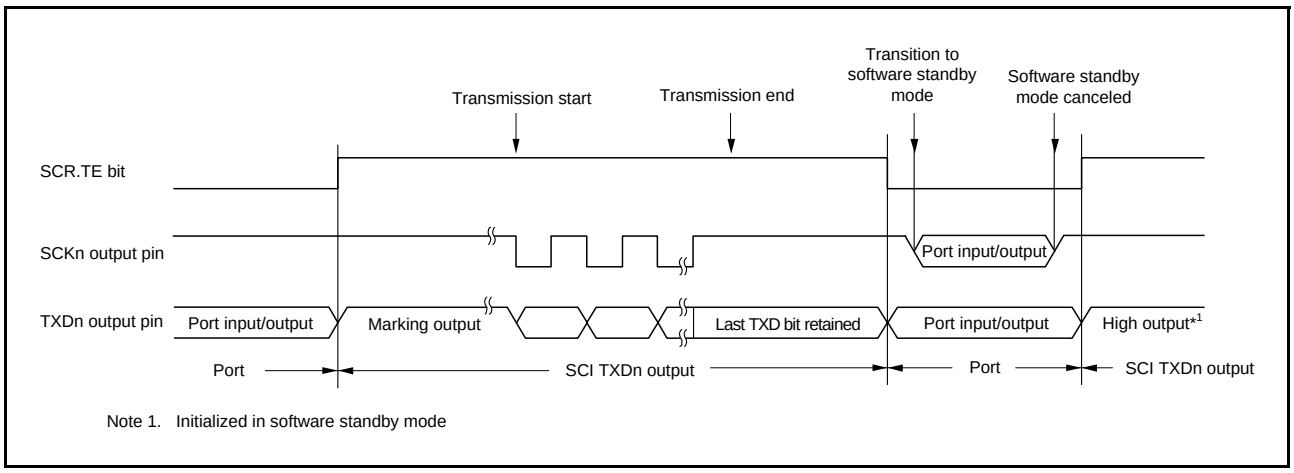


Figure 27.70 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)



Note 1. Initialized in software standby mode

Figure 27.71 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

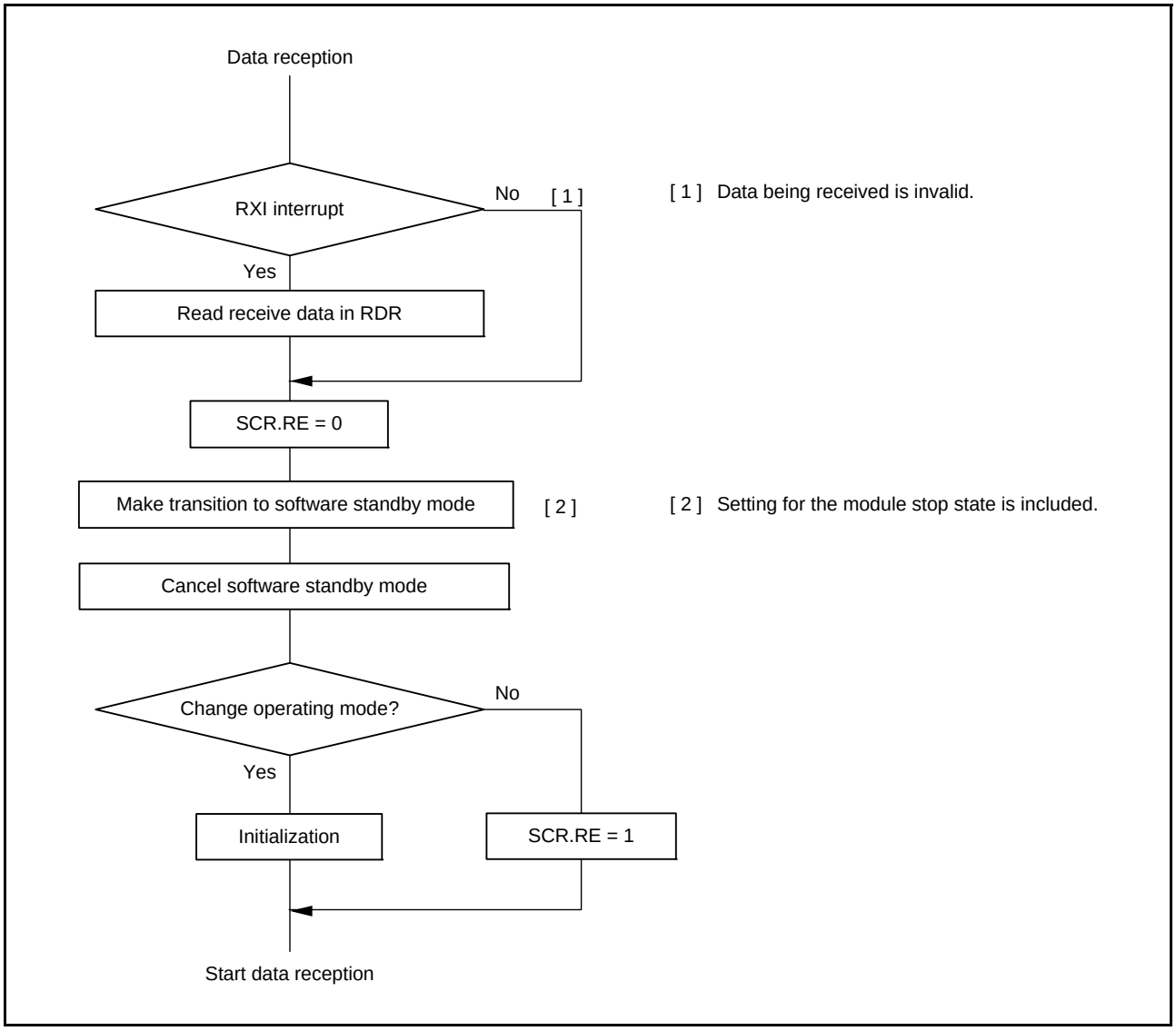


Figure 27.72 Example of Flowchart for Transition to Software Standby Mode during Reception

27.13.10 External Clock Input in Clock Synchronous Mode

In clock synchronous 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

27.13.11 Limitation on Using Simple SPI Bus 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 CKPH and CKPOL bits in SPMR.
- In the case of the setting for clock delay (the SPMR.CKPH bit is 1), the received data full interrupt (RXI interrupt) is generated before the final clock edge on the SCKn pin as indicated in Figure 27.73. 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.
- When operation is in multi-master mode, take care because the SCKn pin output becomes high impedance while the input on the SS# 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.

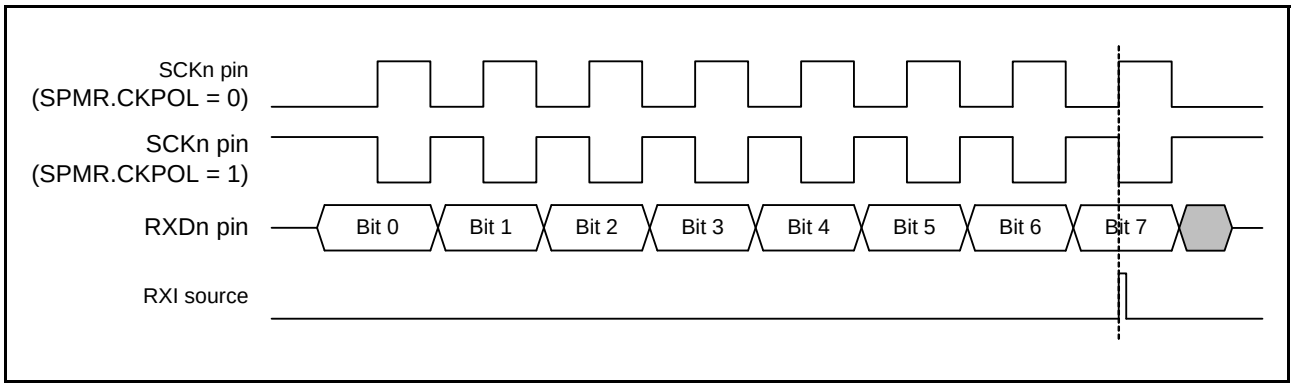


Figure 27.73 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- When data for transmission are written to the TDR, secure at least five cycles of the PCLK from input of the low level on the SSn# pin to input of the external clock.
- 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.

27.13.12 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.

27.13.13 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 reception 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 SCI (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 reception 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.

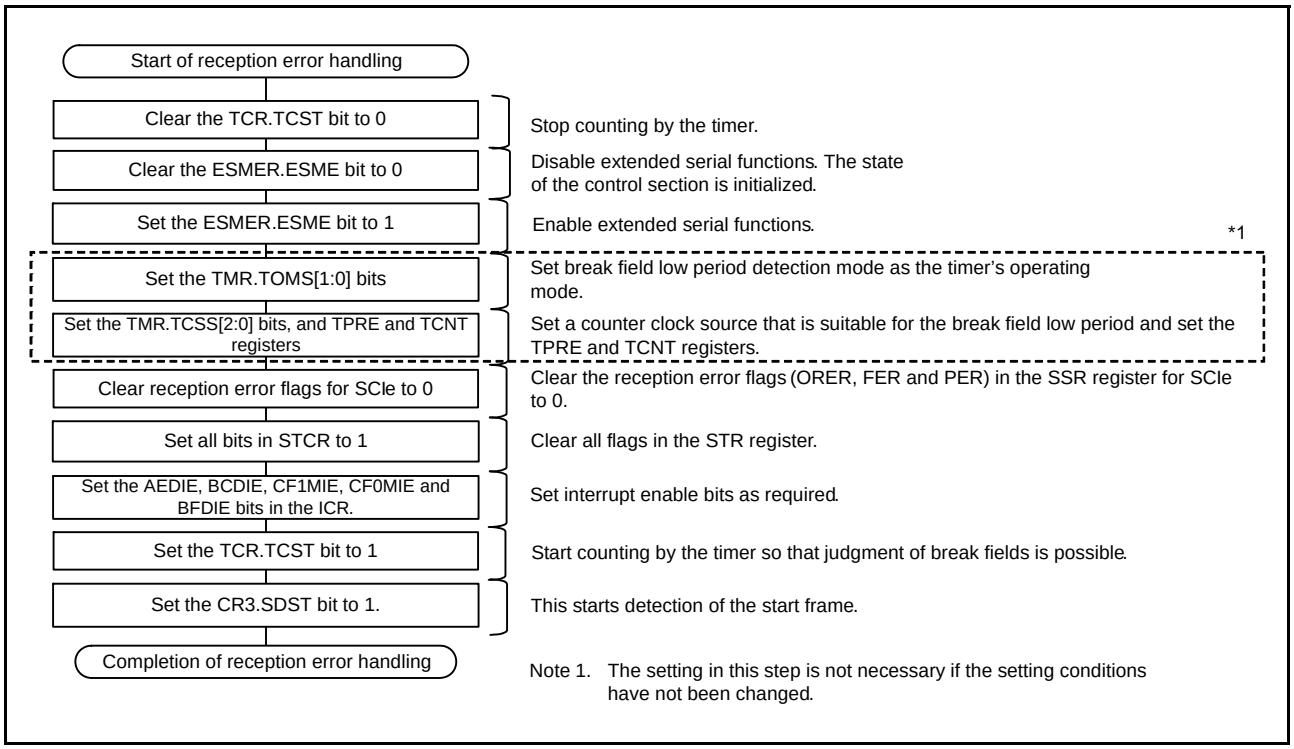


Figure 27.74 Example of Flowchart for Reception Error Handling (During Reception of the Start Frame)

27.13.14 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" (n = 1, 5, 6, 9, 12), 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".
Note that a TXI interrupt request is generated when both bits SCR.TE and SCR.TIE become 1.

28. IrDA Interface

The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI5 based on the IrDA (Infrared Data Association) standard 1.0.

28.1 Overview

Enabling the IrDA function by using the IRE bit in the IRCR register allows encoding and decoding the TXD5 and RXD5 signals of the SCI5 to the waveforms conforming to the IrDA standard 1.0 (IRTXD5 and IRRXD5 pins). Connecting these waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Since the IrDA interface cannot change the transfer rate automatically, the transfer rate should be changed through software.

Figure 28.1 is a block diagram showing cooperation between the IrDA and SCI5.

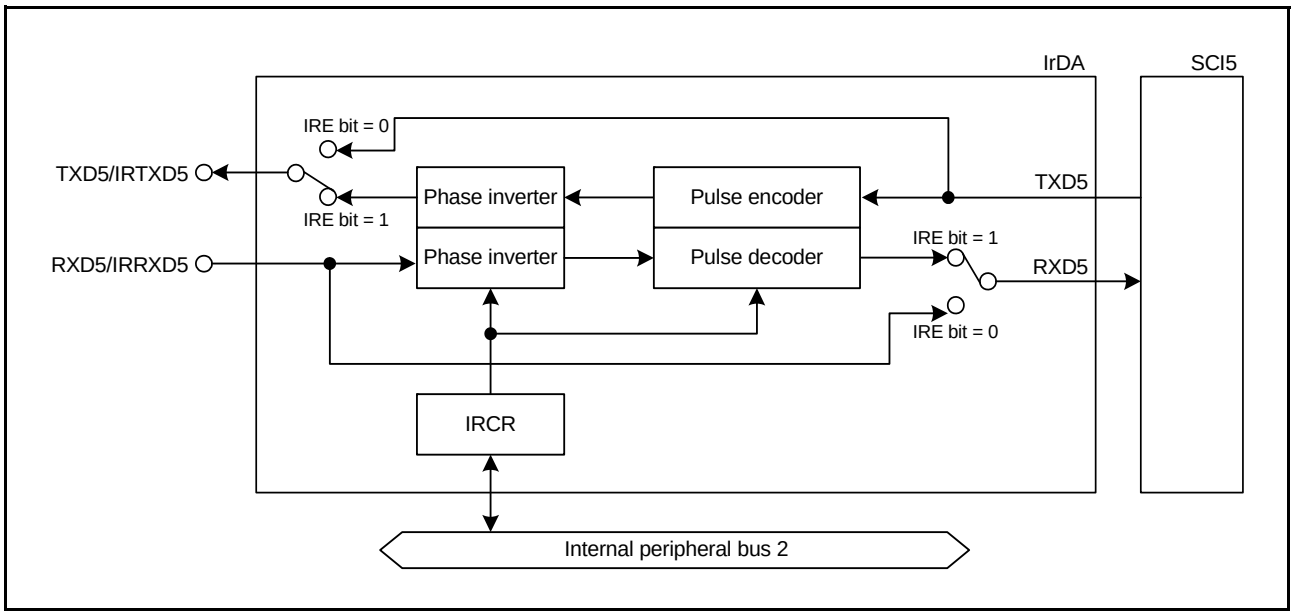


Figure 28.1 Block Diagram Showing Cooperation between IrDA and SCI5

Table 28.1 IrDA Interface Input Pins

Pin Name	I/O	Function
IRTXD5	Output	Outputs data to be transmitted.
IRRXD5	Input	Inputs received data.

28.2 Register Descriptions

28.2.1 IrDA Control Register (IRCR)

Address(es): 0008 8410h

	b7	b6	b5	b4	b3	b2	b1	b0
	IRE	IRCKS[2:0]			IRTXINV	IRRXINV	—	—
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
b2	IRRXINV	IRRXD Polarity Switching	0: IRRXD input is used as received data as is. 1: IRRXD input is used as received data after the polarity is inverted.	R/W
b3	IRTXINV	IRTXD Polarity Switching	0: Data to be transmitted is output to IRTXD as is. 1: Data to be transmitted is output to IRTXD after the polarity is inverted.	R/W
b6 to b4	IRCKS[2:0]	IrDA Clock Select	b6 b4 0 0 0: B × 3/16 (B = bit rate) 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	IRE	IrDA Enable	0: Serial I/O pins are used for normal serial communication. 1: Serial I/O pins are used for IrDA data communication.	R/W

Note 1. The IRCR register values are retained in module stop function mode, sleep mode, all-module clock stop mode, and software standby mode.

IRRXINV Bit (IRRXD Polarity Switching)

This bit inverts the logic level of the IRRXD input. When inverted, the high-level pulse width set by the IRCKS[2:0] bits applies to the low-level pulse width.

IRTXINV Bit (IRTXD Polarity Switching)

This bit inverts the logic level of the IRTXD output. When inverted, the high-level pulse width set by the IRCKS[2:0] bits applies to the low-level pulse width.

IRCKS[2:0] Bits (IrDA Clock Select)

These bits set the high-level pulse width during IRTXD output encoding when the IrDA function is enabled.

IRE Bit (IrDA Enable)

This bit selects either normal serial communication or IrDA data communication as the function of the serial I/O pins.

28.3 Operation

28.3.1 IrDA Interface Setting Procedure

Set the IrDA interface operation according to the following procedure.

(1) Set the corresponding pins to IRTXD5 and IRRXD5 by the pin function select register (Pmn.PFS = 1010b) of the multi-function pin controller (MPC).
(2) Set the general I/O port to the port mode by the port mode register (PORTm.PMR = 0) of the I/O port.
(3) Specify the IrDA function by the IRCR register.
(4) Set the SCI5-related registers of the serial communications interface.

28.3.2 Transmission

In transmission, the signals output from the SCI5 (UART frames) are converted to the IR frame data through the IrDA interface (see Figure 28.2). When the IRCR.IRTXINV bit is 0 and serial data is 0, high-level pulses with 3/16 the width of the bit rate (1-bit width period) are output (initial setting). The high-level pulse width can be changed by setting the IRCR.IRCKS[2:0] bits. The standard prescribes that the minimum high-level pulse width should be 1.41 µs and the maximum high-level pulse width be (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When the peripheral module clock PCLK is 20 MHz, the high-level pulse width can be 1.41 µs to 1.6 µs. When serial data is 1, no pulses are output.

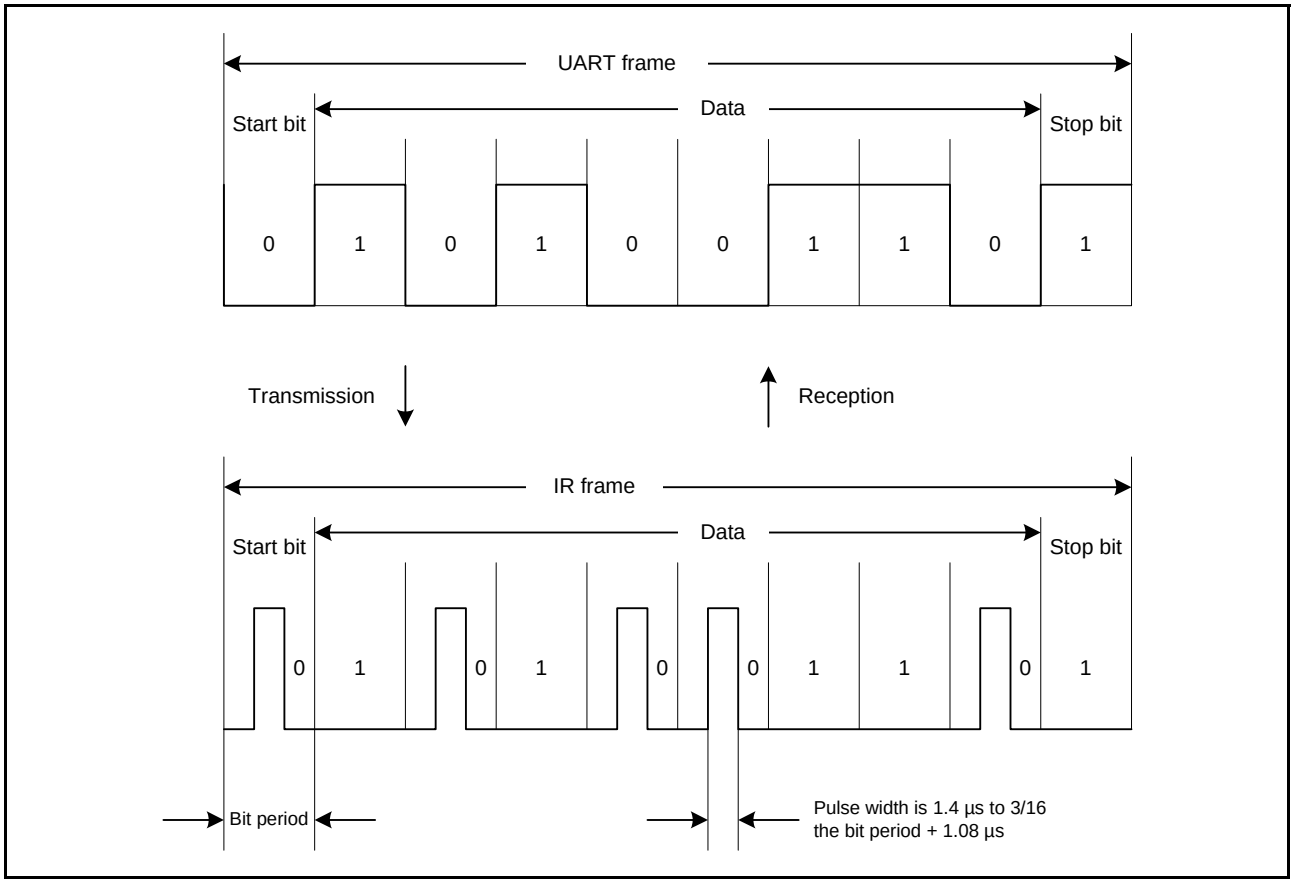


Figure 28.2 IrDA Transmission/Reception

28.3.3 Reception

In reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI5. Low-level data is output when the IRCR.IRRXINV bit is 0 and a high-level pulse is detected; high-level data is output when no pulse is detected for a 1-bit period. Note that pulses shorter than the minimum pulse width of 1.41 µs are not recognized.

28.3.4 Selecting High-Level Pulse Width

When the pulse width should be shorter than bit rate × 3/16 for transmission, IRCKS[2:0] bit setting (minimum pulse width) shown in Table 28.2 can be used. In the table, the corresponding operating frequencies and bit rates of this module are also shown.

Table 28.2 IRCKS[2:0] Bit Setting

Peripheral Module Operating Frequency PCLK (MHz)	Bit Rate (bps) (Upper Row)/Bit Period × 3/16 (µs) (Lower Row)					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
4.9152	011b	011b	011b	011b	011b	011b
5	011b	011b	011b	011b	011b	011b
6	100b	100b	100b	100b	100b	—
6.144	100b	100b	100b	100b	100b	—
7.3728	100b	100b	100b	100b	100b	—
8	100b	100b	100b	100b	100b	—
9.8304	100b	100b	100b	100b	100b	100b
10	100b	100b	100b	100b	100b	100b
12	101b	101b	101b	101b	101b	—
12.288	101b	101b	101b	101b	101b	—
14	101b	101b	101b	101b	101b	—
14.7456	101b	101b	101b	101b	101b	—
16	101b	101b	101b	101b	101b	—
16.9344	101b	101b	101b	101b	101b	—
17.2032	101b	101b	101b	101b	101b	—
18	101b	101b	101b	101b	101b	—
19.6608	101b	101b	101b	101b	101b	101b
20	101b	101b	101b	101b	101b	101b
25	110b	110b	110b	110b	110b	—

—: The bit rate cannot be set for the SCI5.

28.4 Usage Notes

28.4.1 Module Stop Function Setting

The IrDA can be enabled and disabled by setting the MSTPCRC.MSTPC20 bit. The IrDA is stopped after a reset. Registers can be accessed by enabling the IrDA. For details, refer to section 11, Low Power Consumption.

28.4.2 Minimum Pulse Width during Reception

Pulses shorter than the minimum pulse width of 1.41 μs are not recognized.

28.4.3 Asynchronous Reference Clock for SCI5

DB: The IrDA receives a clock with a frequency 16 times the bit rate from SCI5 and operates in conjunction with SCI5. When using the IrDA, set the SCI5.SEMR.ABCS bit to 0.

29. I2C Bus Interface (RIIC)

The RX220 Group has one I2C bus interface (RIIC module).
The RIIC module conforms with and provides a subset of the NXP I2C bus (Inter-IC-Bus) interface functions.

29.1 Overview

Table 29.1 lists the specifications of the RIIC, Figure 29.1 shows a block diagram of the RIIC, and Figure 29.2 shows an example of I/O pin connections to external circuits (I2C bus configuration example). Table 29.2 lists the I/O pins of the RIIC.

Table 29.1 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 set-up times, hold times, and bus-free times for the transfer rate
Transfer rate	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 slave-address settings can be made. • Seven- and ten-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 clock signal (SCL) 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 (WAIT function)
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) signal 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 time-out function is capable of detecting long-interval stop of the SCL (clock signal).
Noise removal	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 AL, NACK, time-out, a start condition including a restart condition, or a stop condition) Receive-data-full (including matching with a slave address) Transmit-data-empty (including matching with a slave address) Transmission complete

Table 29.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
Event link function	• Four sources: Error in transfer or occurrence of events (detection of AL, NACK, time-out, a start condition including a restart condition, or a stop condition) Receive-data-full (including matching with a slave address) Transmit-data-empty (including matching with a slave address) Transmission complete

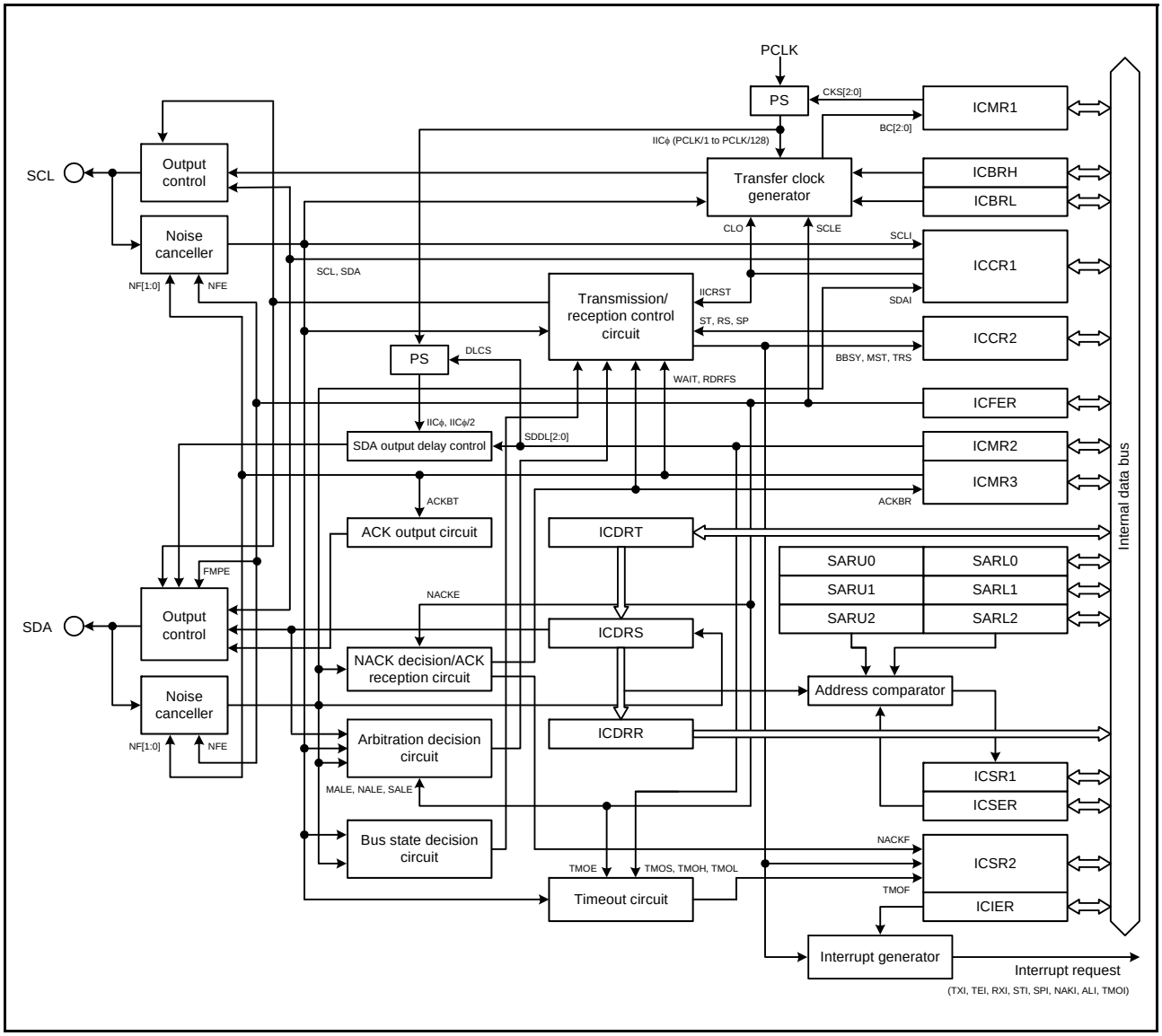


Figure 29.1 Block Diagram of RIIC

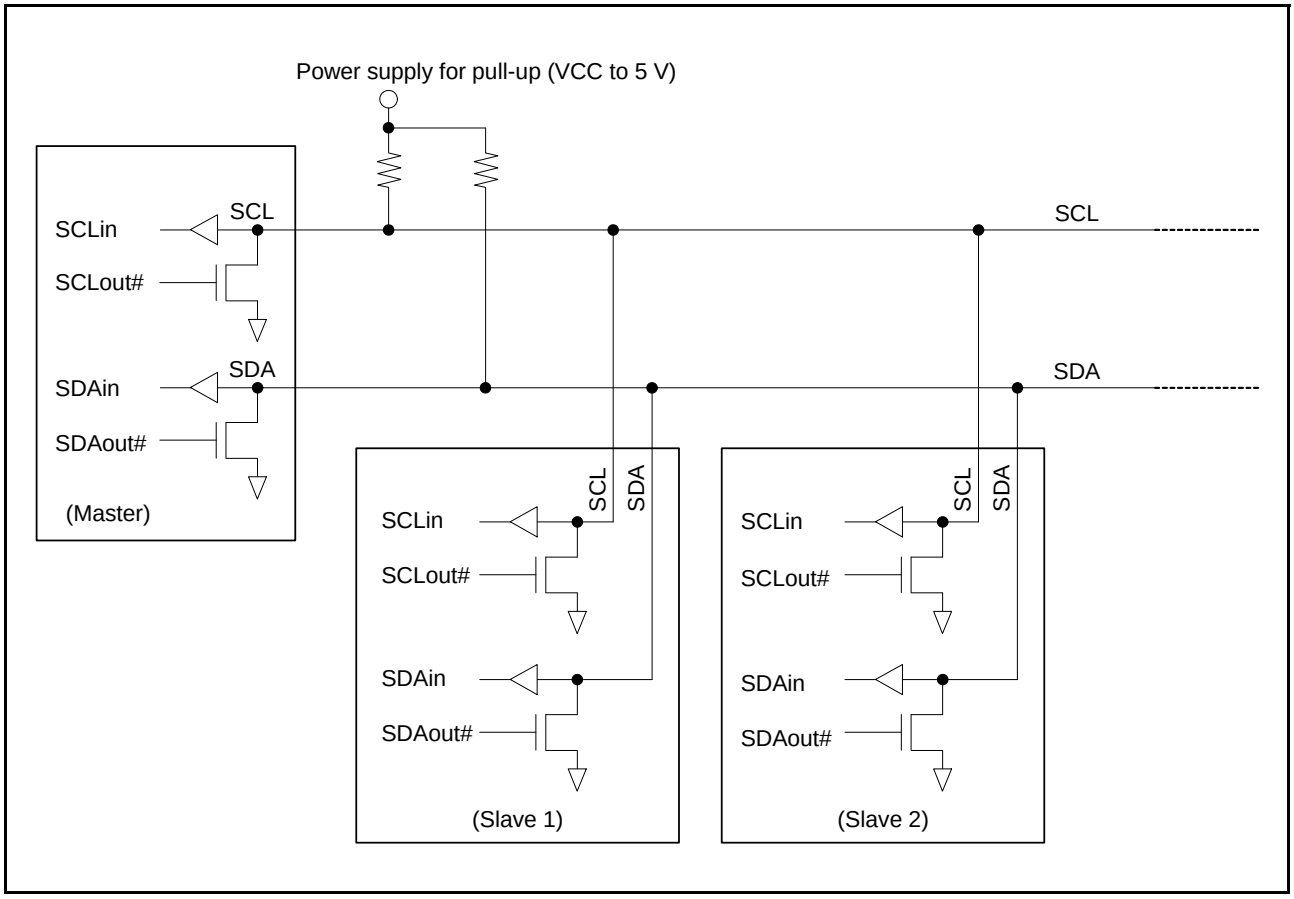


Figure 29.2 Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

The input level of the signals for RIIC is CMOS when I2C bus is selected (the ICMR3.SMBS bit = 0), or TTL when SMBus is selected (the ICMR3.SMBS bit = 1).

Table 29.2 Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL	I/O	RIIC0 serial clock I/O pin
	SDA	I/O	RIIC0 serial data I/O pin

29.2 Register Descriptions

Table 29.3 Register Allocation for 16-Bit Access

Address	Upper 8 Bits	Lower 8 Bits
0008 830Ah*1	RIIC0.TMOCNTU	RIIC0.TMOCNTL

Note 1. Same address with ones of the slave address registers, SARL0, SARU0. Care should be taken.

29.2.1 I2C Bus Control Register 1 (ICCR1)

Address(es): 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA line Monitor	0: SDA line is low. 1: SDA line is high.	R
b1	SCLI	SCL line Monitor	0: SCL line is low. 1: SCL line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA pin low. 1: The RIIC has released the SDA pin. • Write: 0: The RIIC drives the SDA pin low. 1: The RIIC releases the SDA pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCL pin low. 1: The RIIC has released the SCL pin. • Write: 0: The RIIC drives the SCL pin low. 1: The RIIC releases the SCL pin.	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: Bits SCLO and SDAO can be written. 1: Bits SCLO and SDAO 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: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)	R/W
b7	ICE	I2C Bus Interface Enable	0: Disable (SCL and SDA pins in inactive state) 1: Enable (SCL and SDA 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 SDA and SCL 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, repeated START 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, see section 29.11.2, 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 29.4 lists the resets of the RIIC.
The RIIC reset resets all registers including the BBSY flag in ICCR2 and internal states of the RIIC, and the internal reset resets the bit counter (BC[2:0] bits in ICMR1), 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, see section 29.14, Reset States.
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 SCL pin and SDA 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 SCL 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 29.4 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Reset the BC[2:0] bits in ICMR1, and the ICSR1, ICSR2, 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 SCL and SDA pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 29.4, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL and SDA 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 SCL and SDA pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL or SDA 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.

29.2.2 I2C Bus Control Register 2 (ICCR2)

Address(es): 0008 8301h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition 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 MTWP bit in ICMR1 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).

For details on the start condition issuance, see section 29.10, 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
- 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).
Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

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) and the MST bit is set to 1 (master mode).
For details on the restart condition issuance, see section 29.10, 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 or 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 the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, 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) and the MST bit is set to 1 (master mode).
For details on the stop condition issuance, see section 29.10, 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 or 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).
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 the TRS bit is automatically changed to the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. 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 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 the value for master mode or slave mode by detection or issuing of a start 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)

The BBSY flag indicates whether the I2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared 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)

29.2.3 I2C Bus Mode Register 1 (ICMR1)

Address(es): 0008 8302h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Selection	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 MST and TRS bits in ICCR2. 1: Enables writing to the MST and TRS bits in ICCR2.	R/W

Note 1. Set the BCWP bit to 0 to rewrite the BC[2:0] bits. The BC[2:0] bits must be rewritten by using the MOV instruction.

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 SCL 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 SCL line is at a low level.

The values of the BC[2:0] bits return 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

29.2.4 I2C Bus Mode Register 2 (ICMR2)

Address(es): 0008 8303h

	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 Selection	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count is disabled while the SCL line is at a low level. 1: Count is enabled while the SCL line is at a low level.	R/W
b2	TMOH	Timeout H Count Control	0: Count is disabled while the SCL line is at a high level. 1: Count is enabled while the SCL 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 Selection	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 Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (TMOE bit = 1 in ICFER). 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 SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.
For details on the timeout function, see section 29.11.1, 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 SCL line is held low when the timeout function is enabled (TMOE bit = 1 in ICFER).

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 SCL line is held high when the timeout function is enabled (TMOE bit = 1 in ICFER).

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).

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.
For details on this function, see section 29.5, Facility for Delaying SDA Output.

Note: • 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.

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])

29.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): 0008 8304h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Selection	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.	W*1
b5	RDRFS	RDRF Flag Set Timing Selection	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL 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 SCL 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 Selection	0: The I2C bus is selected. 1: The SMBus is selected.	R/W

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.
Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

ICMR3 has functions to send/receive acknowledge, to select the RDRF set timing in RIIC receive operation, and to control WAIT operation.

Note: • Set the noise range to be filtered out by the noise filter within a range less than the SCL 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)

Note: • The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.
When the RDRFS bit is 0, the SCL 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 SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL 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 receive data buffer (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 SCL 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 Selection)

Setting this bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

29.2.6 I2C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h

	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 MST and TRS bits in ICCR2 automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the MST and TRS bits in ICCR2 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, see section 29.11.1, 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, see section 29.8.2, 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 cleared to 0 (SCL synchronous circuit not 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 SCL 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 cleared to 0 except for checking the output of the set transfer rate.

29.2.7 I2C Bus Status Enable Register (ICSER)

Address(es): 0008 8306h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in 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 received slave address and 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 cleared 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 cleared 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 cleared 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, see section 29.7.3, 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 cleared to 0, the received slave address is ignored even if it matches the host address.

29.2.8 I2C Bus Interrupt Enable Register (ICIER)

Address(es): 0008 8307h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Enable	0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Enable	0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Enable	0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Enable	0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Enable	0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt request (RXI) is disabled. 1: Receive data full interrupt request (RXI) is enabled.	R/W
b6	TEIE	Transmit End Interrupt Enable	0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.	R/W

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Enable)

This bit is used to enable or disable receive data full interrupt requests (RXI) when the RDRF flag in ICSR2 is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (TEI) when the TEND flag in ICSR2 is set to 1. An TEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (TXI) when the TDRE flag in ICSR2 is set to 1.

29.2.9 I2C Bus Status Register 1 (ICSR1)

Address(es): 0008 8308h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL0 while the FS bit in SARU0 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU0) and the following address matches the SARL0 value while the FS bit in SARU0 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL0 match determination frame.)	R(/W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL1 while the FS bit in SARU1 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU1) and the following address matches the SARL1 value while the FS bit in SARU1 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL1 match determination frame.)	R(/W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected. • This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL2 while the FS bit in SARU2 is 0 (7-bit address format selected). • This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU2) and the following address matches the SARL2 value while the FS bit in SARU2 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL2 match determination frame.)	R(/W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected. • This bit is set to 1 when the received slave address matches the general call address (all 0).	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) (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 cleared 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 cleared 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 cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[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 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 enabled)
  This flag is cleared 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)

[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 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 enabled)
  This flag is cleared 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 enabled)
  This flag is cleared 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)

[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 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 enabled)
  This flag is cleared 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

29.2.10 I2C Bus Status Register 2 (ICSR2)

Address(es): 0008 8309h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R(/W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R(/W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R(/W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R(/W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R(/W) *1
b5	RDRF	Receive Data Full Flag	0: 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)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.
[Setting condition]

- When the SCL line state remains unchanged for the period specified by bits TMOH, TMOL, and TMOS in ICMR2 with the TMOE bit in ICFER set to 1 (timeout detection enabled) in master mode or in the slave specification state.

[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)

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 SDA 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 bit 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 SDA 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 SDA line is driven low while the internal SDA output is at a high level (the SDA 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 SDA 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 SDA 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 SDA 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 bit after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Table 29.5 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 SDA 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)

[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)

[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)

[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, clear the NACKF flag to 0.

RDRF Flag (Receive Data Full)

[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 cleared 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)

[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)

[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
  a. When the MST bit in ICCR2 is set to 1 after a start condition (or a restart condition) is detected
  b. When the RIIC enters transmit mode from receive mode
  c. When 1 is written to while the ICMR1.MTWP bit is 1
- When the received slave address matches while the TRS bit is 1

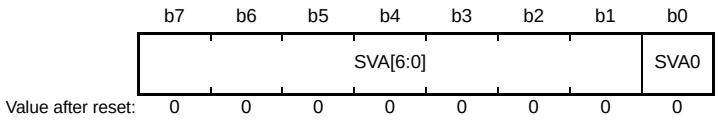
[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is cleared to 0
  a. When a stop condition is detected
  b. When the RIIC enters receive mode from transmit mode
  c. When 0 is written to while the ICMR1.MTWP bit is 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 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.

29.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah RIIC0.SARL1 0008 830Ch RIIC0.SARL2 0008 830Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	The least significant bit (LSB) of a 10-bit slave address is set. • When the FS bit in SARUy is 0 (7-bit address format), this bit is invalid. • When the FS bit in SARUy is 1 (10-bit address format), this bit is the LSB of the lower 8-bit address (combined with the SVA[6:0] bits) of a 10-bit slave address.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set. • When the FS bit in SARUy is 0 (7-bit address format), these bits form a 7-bit slave address. • When the FS bit in SARUy is 1 (10-bit address format), these bits form the lower 8-bit address (combined with the SVA0 bit) of a 10-bit slave address.	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 eight 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 eight 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.

29.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh RIIC0.SARU1 0008 830Dh RIIC0.SARU2 0008 830Fh

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

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Selection	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. • When the SARUy.FS bit is 0 (7-bit address format), these bits are invalid. • When the SARUy.FS bit is 1 (10-bit address format), these bits form the upper two bits of a 10-bit slave address.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Selection)

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 two 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.

29.2.13 I2C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): 0008 8310h

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

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 (see section 29.8, Automatically 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])

29.2.14 I2C Bus Bit Rate High-Level Register (ICBRH)

Address(es): 0008 8311h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-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 + SCL line rising time [tr] + SCL line falling time [tf]}
Duty cycle = {SCL line rising time [tr]*2 + (ICBRH + 1) / IICϕ} / {SCL line falling time [tf]*2 + (ICBRL + 1) / IICϕ}

Note 1. IICϕ = PCLK × Division ratio
Note 2. The SCL line rising time [tr] and SCL 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 29.6 lists examples of ICBRH/ICBRL settings.

Table 29.6 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)

Note: • ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I2C bus standard from NXP Semiconductors.

29.2.15 I2C Bus Transmit Data Register (ICDRT)

Address(es): 0008 8312h

	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.

29.2.16 I2C Bus Receive Data Register (ICDRR)

Address(es): 0008 8313h

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

When 1 byte of data has been received, the received data is transferred from the I2C bus shift register (ICDRS) to 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.

29.2.17 I2C Bus Shift Register (ICDRS)

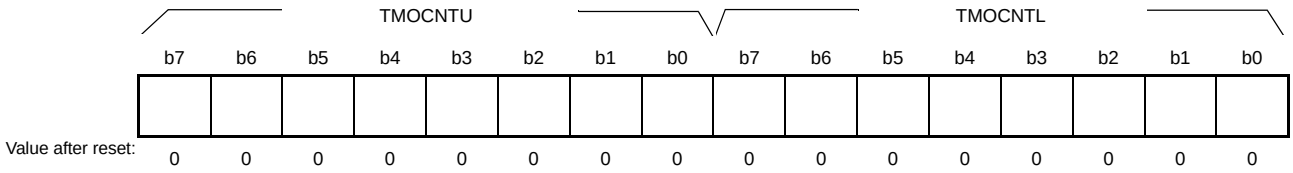
	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—

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 SDA pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.
ICDRS cannot be accessed directly.

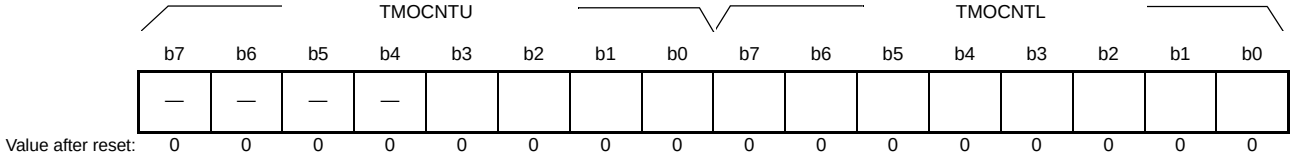
29.2.18 Timeout internal counter (TMOCNT)

Address(es): RIIC0.TMOCNTL 0008 830Ah, RIIC0.TMOCNTU 0008 830Bh

- 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. Value in timeout internal counter cannot be read. When 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), bits b7 to b4 are reserved bits. They are writable. However, the value written is disabled.
Note 2. Value in timeout internal counter cannot be read. When 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/TMOL bit (SCL rising edge/falling edge detection) are met.
The TMOCNTL and TMOCNTU counters can be accessed as 16-bit registers in 16-bit units.

29.3 Operation

29.3.1 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 29.3 shows the I2C bus format, and Figure 29.4 shows the I2C bus timing.

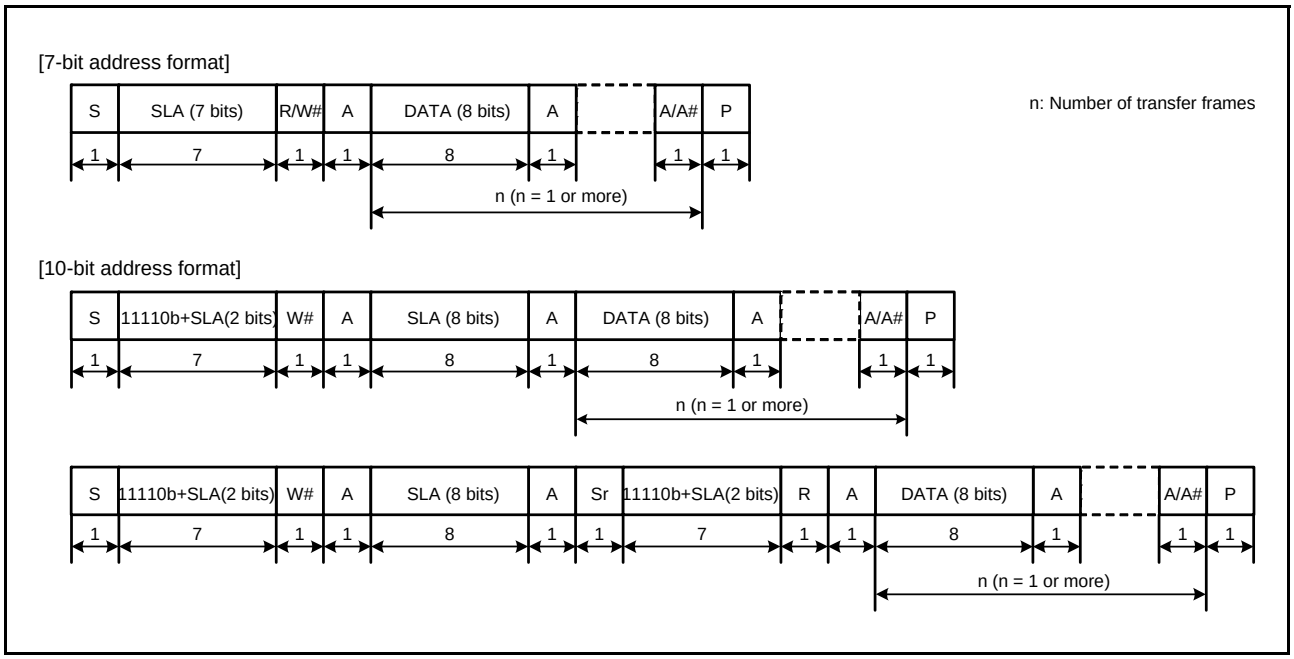


Figure 29.3 I2C Bus Format

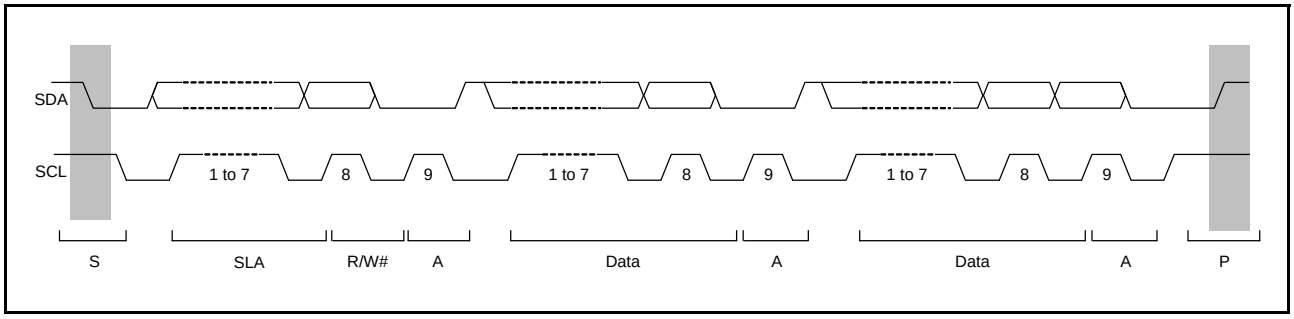


Figure 29.4 I2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL 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 SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

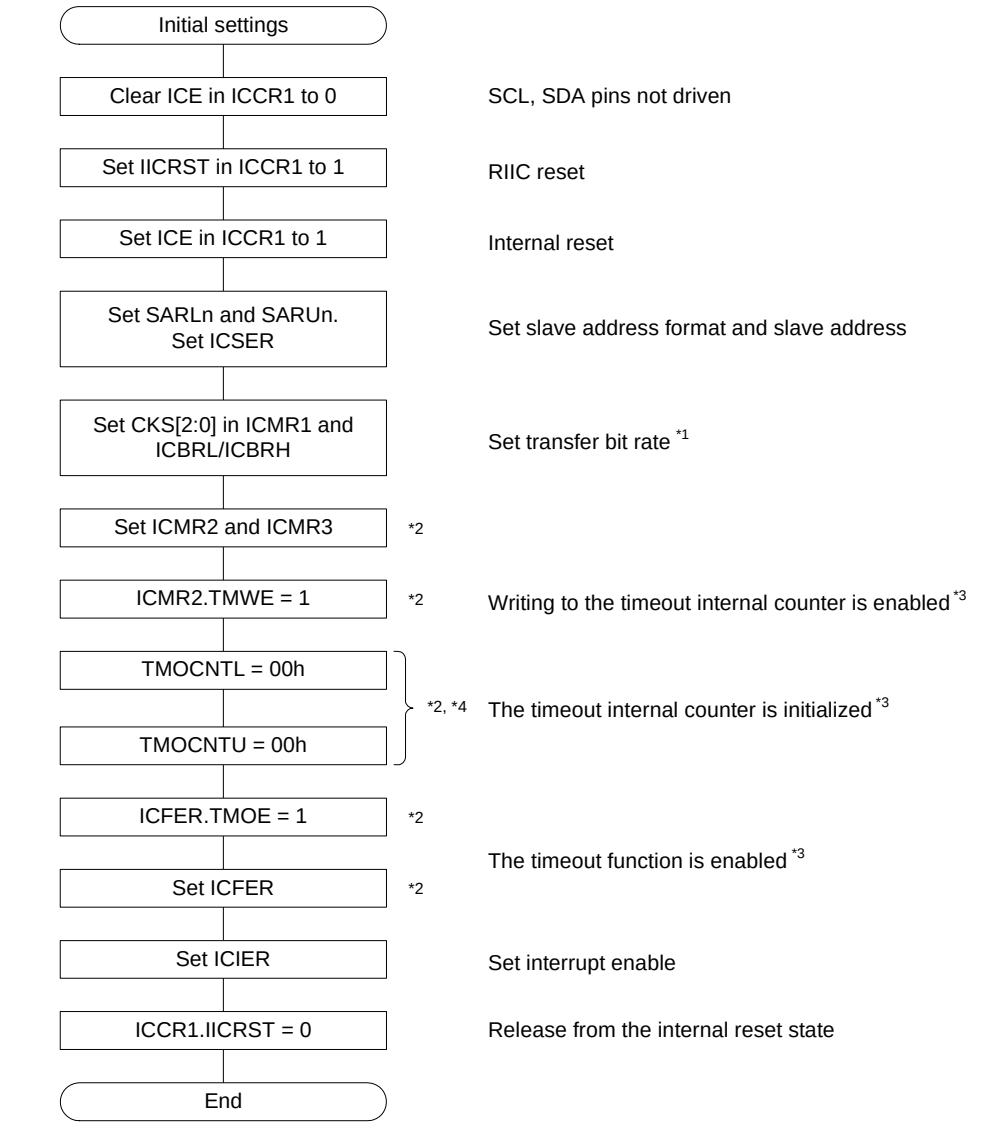
Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

29.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 29.5.



[Legend] n = 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. Those procedures need to be added only when using timeout function. Not needed when not using timeout function.
Note 4. The diagram above is for 8-bit access. For 16-bit access, set 0000h to the address listed in Table 29.3, Register Allocation for 16-Bit Access.

Figure 29.5 Example of RIIC Initialization Flowchart

29.3.3 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 29.6 shows an example of usage of master transmission and Figure 29.7 to Figure 29.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit cleared to 0 (SCL and SDA pins in inactive state). This initializes the various flags and internal state of ICSR1. 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 29.5). When the necessary register settings have been completed, clear the ICCR1.IICRST bit to 0 (to release the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(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 cleared 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 SDA 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.

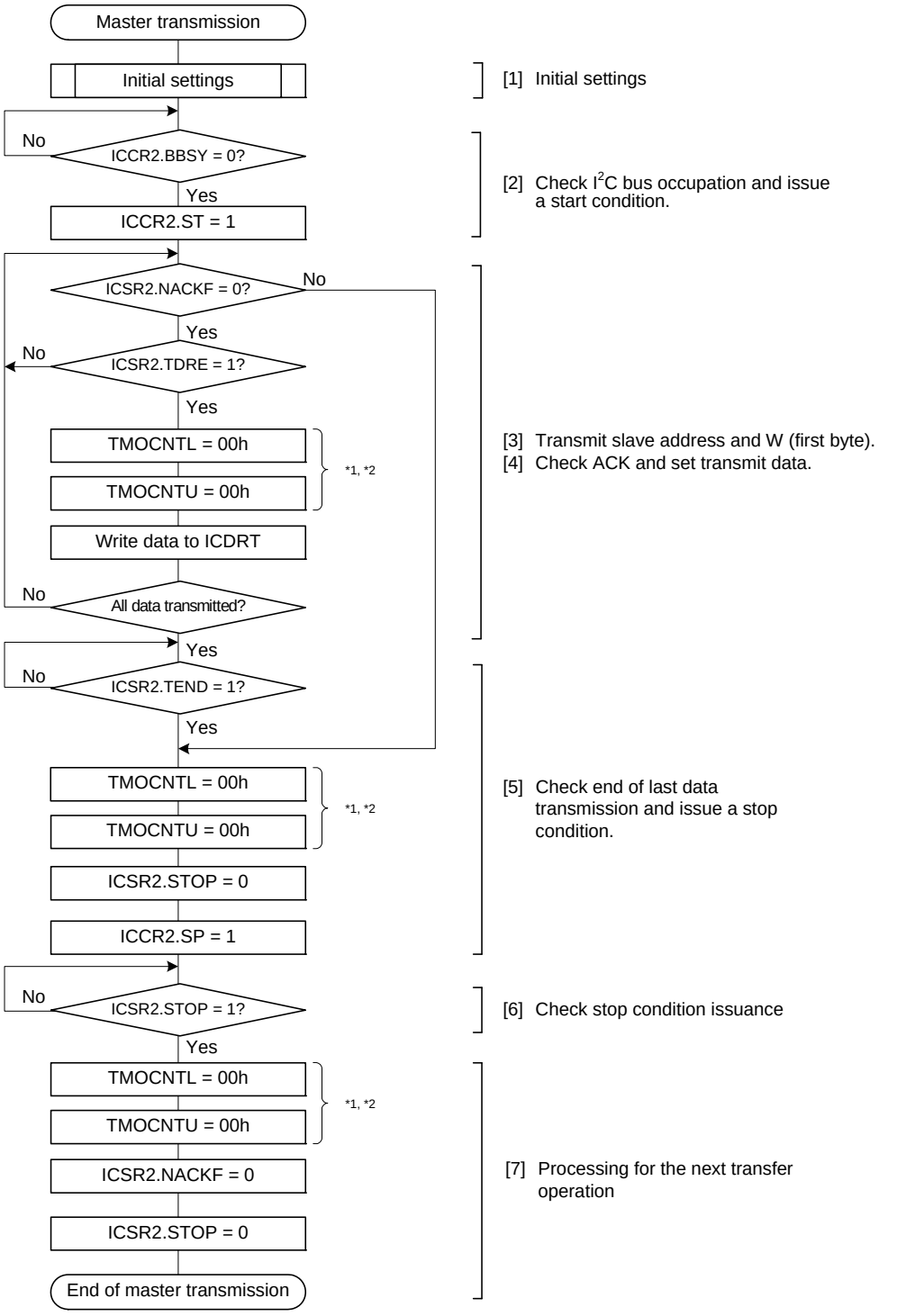
(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again 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 two higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the eight 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 SCL 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 clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically clears 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, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.



Note 1. Those procedures need to be added only when using timeout function. Not needed when not using timeout function.
Note 2. The diagram above is for 8-bit access. For 16-bit access, set 0000h to the address listed in Table 29.3, Register Allocation for 16-Bit Access.

Figure 29.6 Example of Master Transmission Flowchart

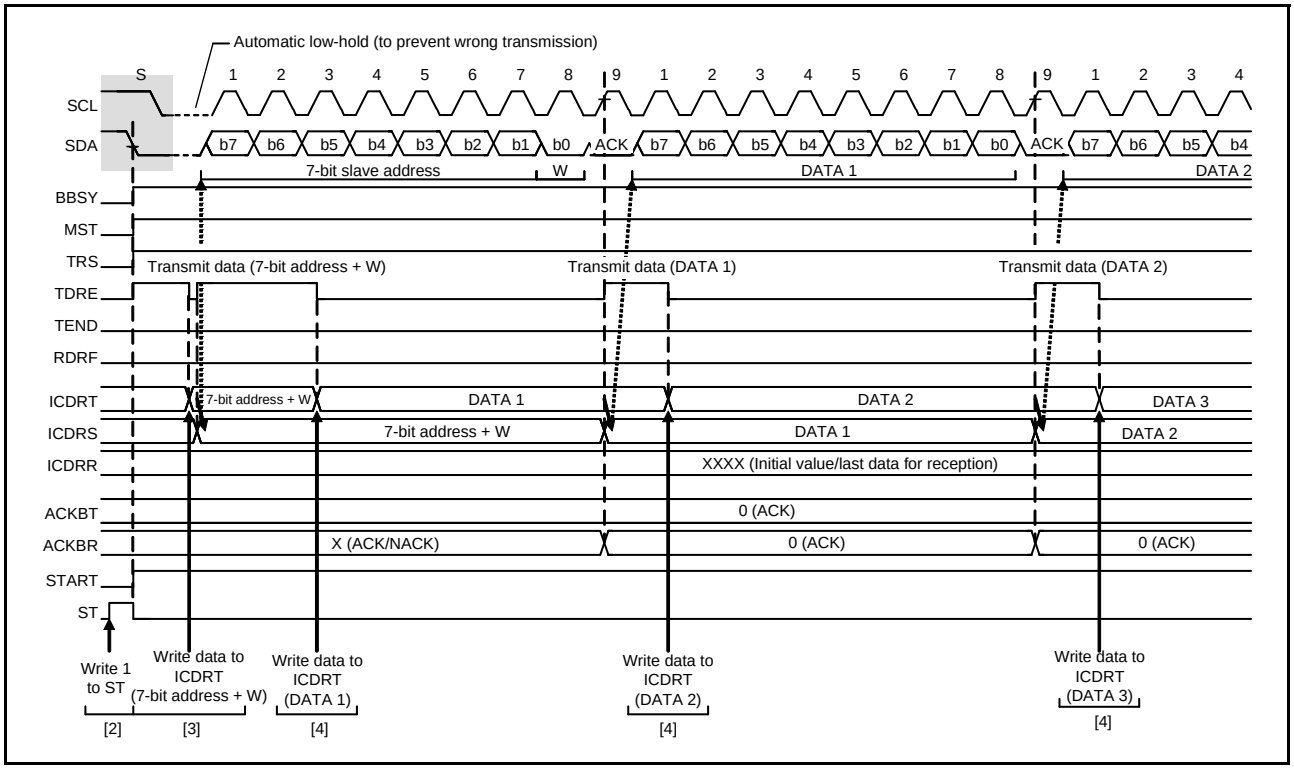


Figure 29.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

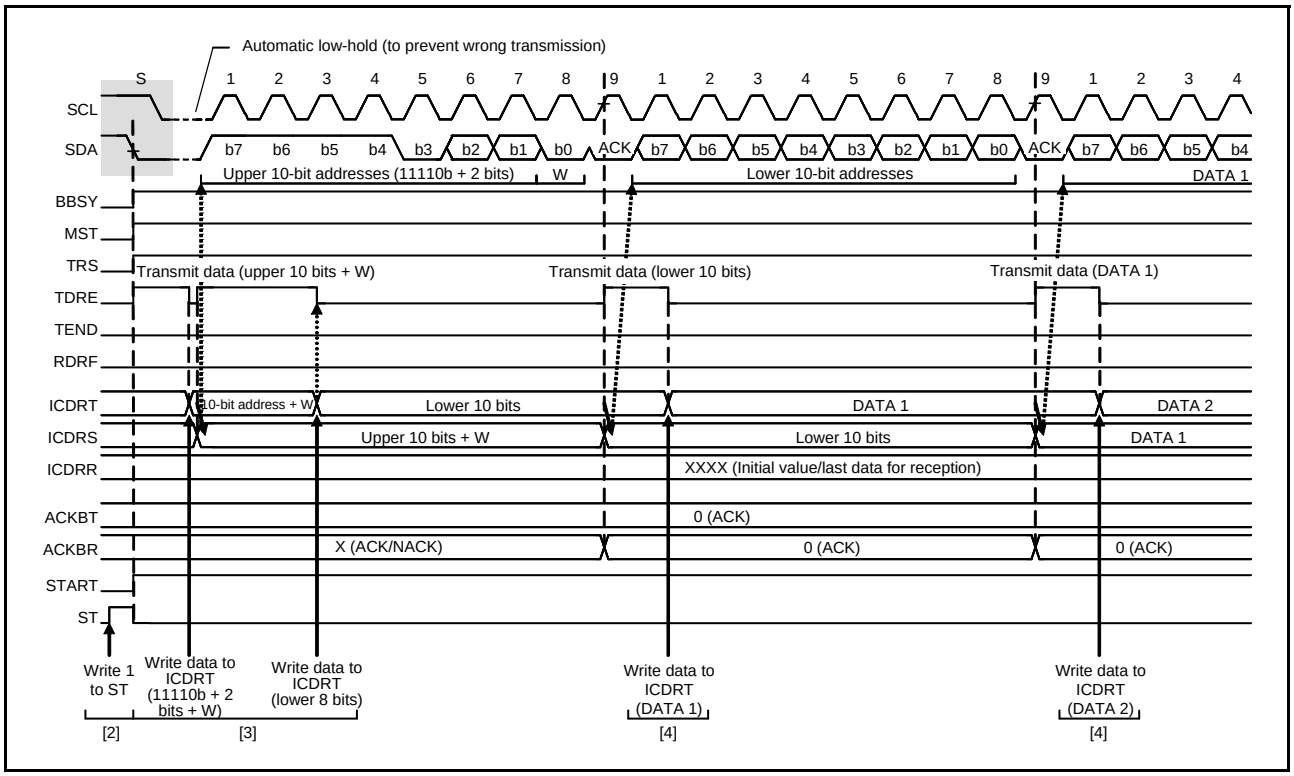


Figure 29.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

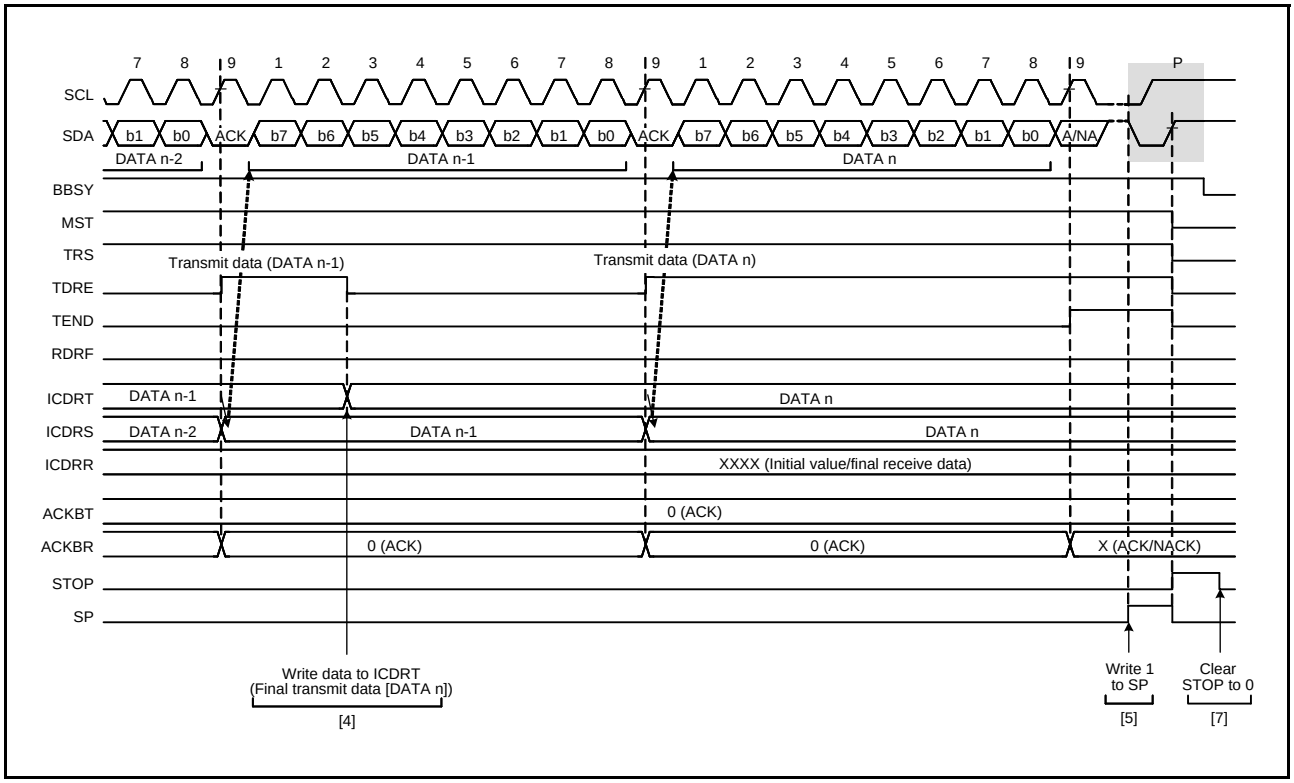


Figure 29.9 Master Transmit Operation Timing (3)

29.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, 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 29.11 shows an example of usage of master reception and Figure 29.12 and Figure 29.14 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit cleared to 0 (SCL and SDA pins in inactive state). This initializes the various flags and internal state of ICSR1. 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 29.5). When the necessary register settings have been completed, clear the ICCR1.IICRST bit to 0 (to release the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(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 cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA 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 cleared 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 cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared 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) signal 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 out ICDRR at this time will produce the received data, and the RDRF flag is automatically cleared 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 ACKBT bit of ICMR3. 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 SCL line to the low level on the rising 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 ACKBT bit in ICMR3 to 1 (NACK).

(7) After reading out 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 SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears 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 STOP flag in ICSR2 to 1.

(9) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

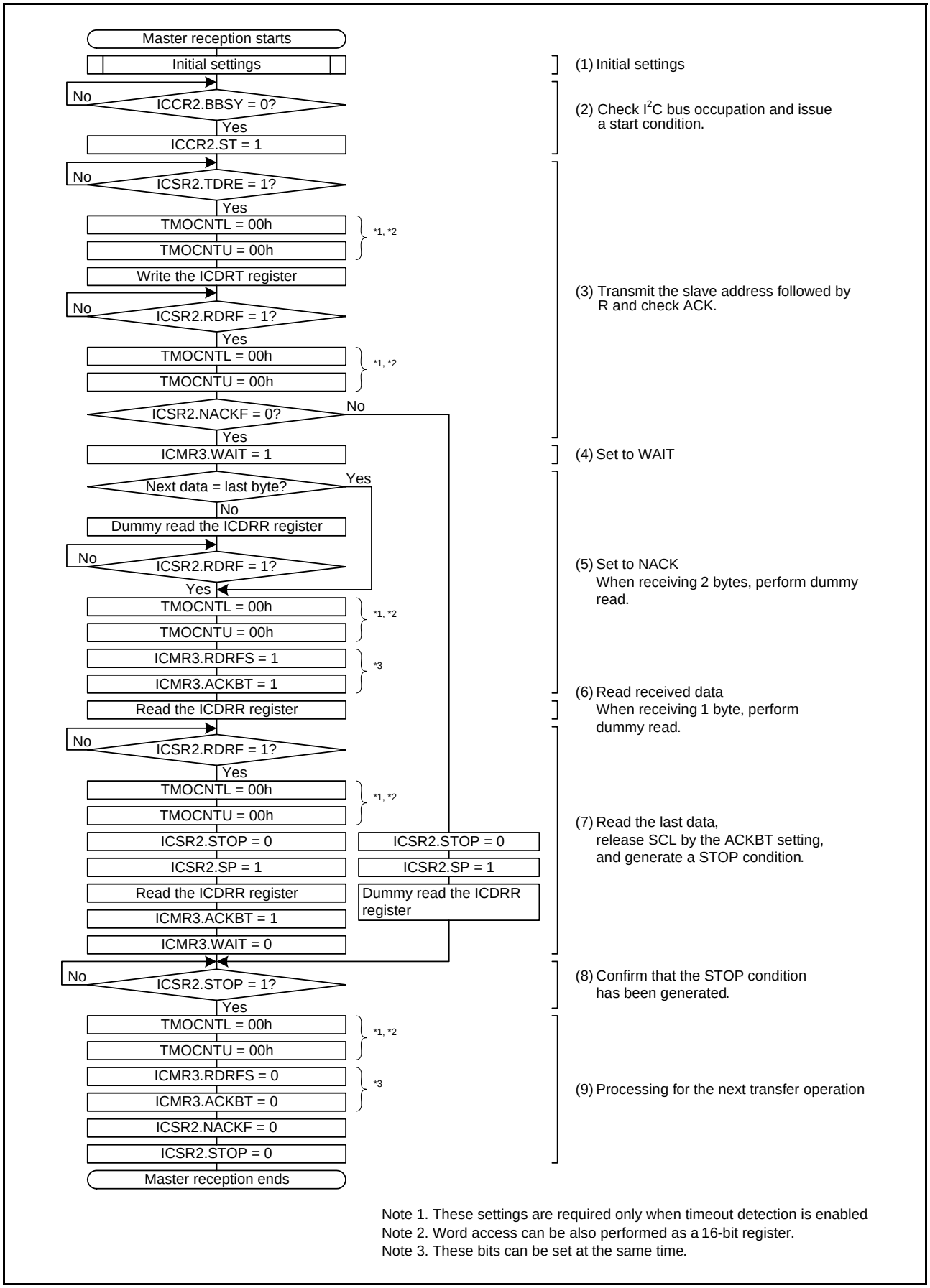
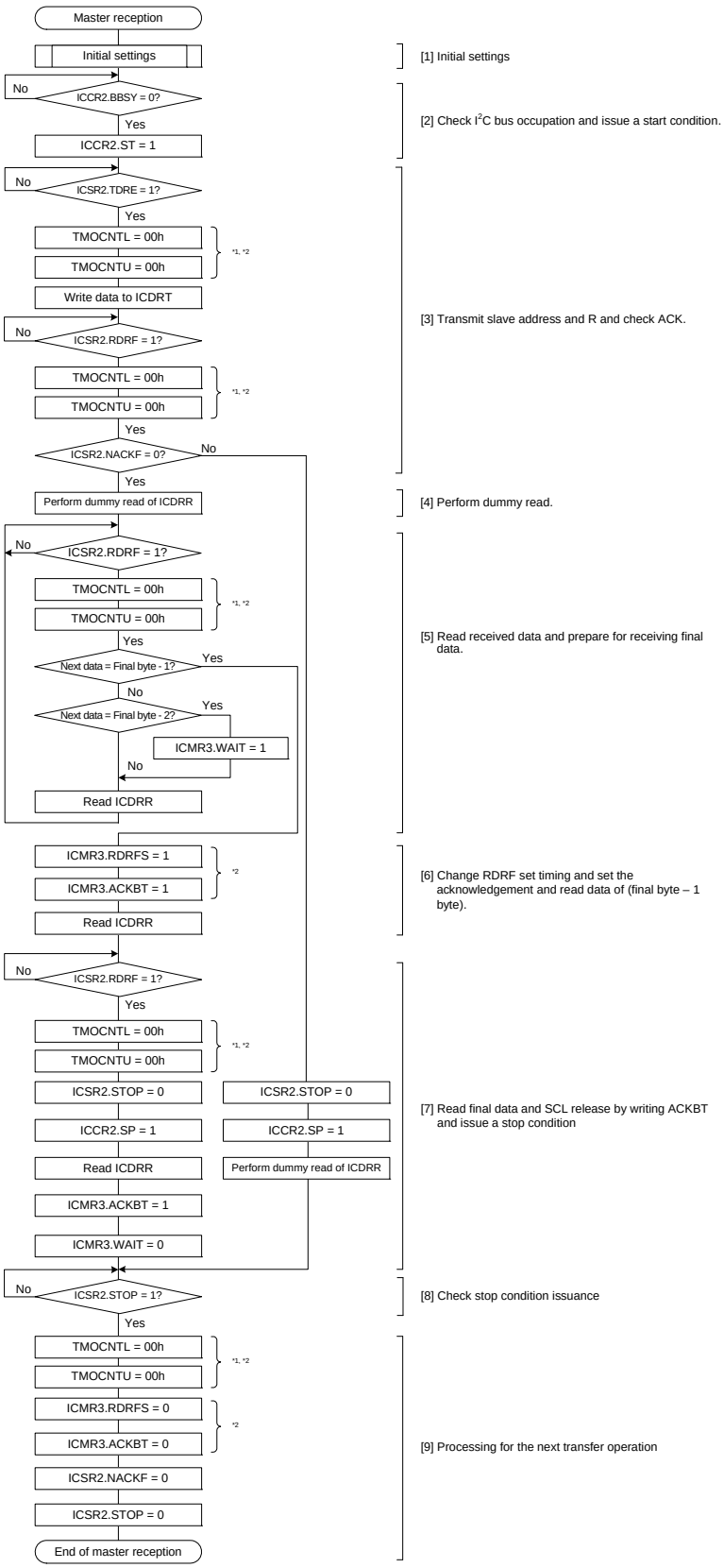


Figure 29.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)



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 29.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

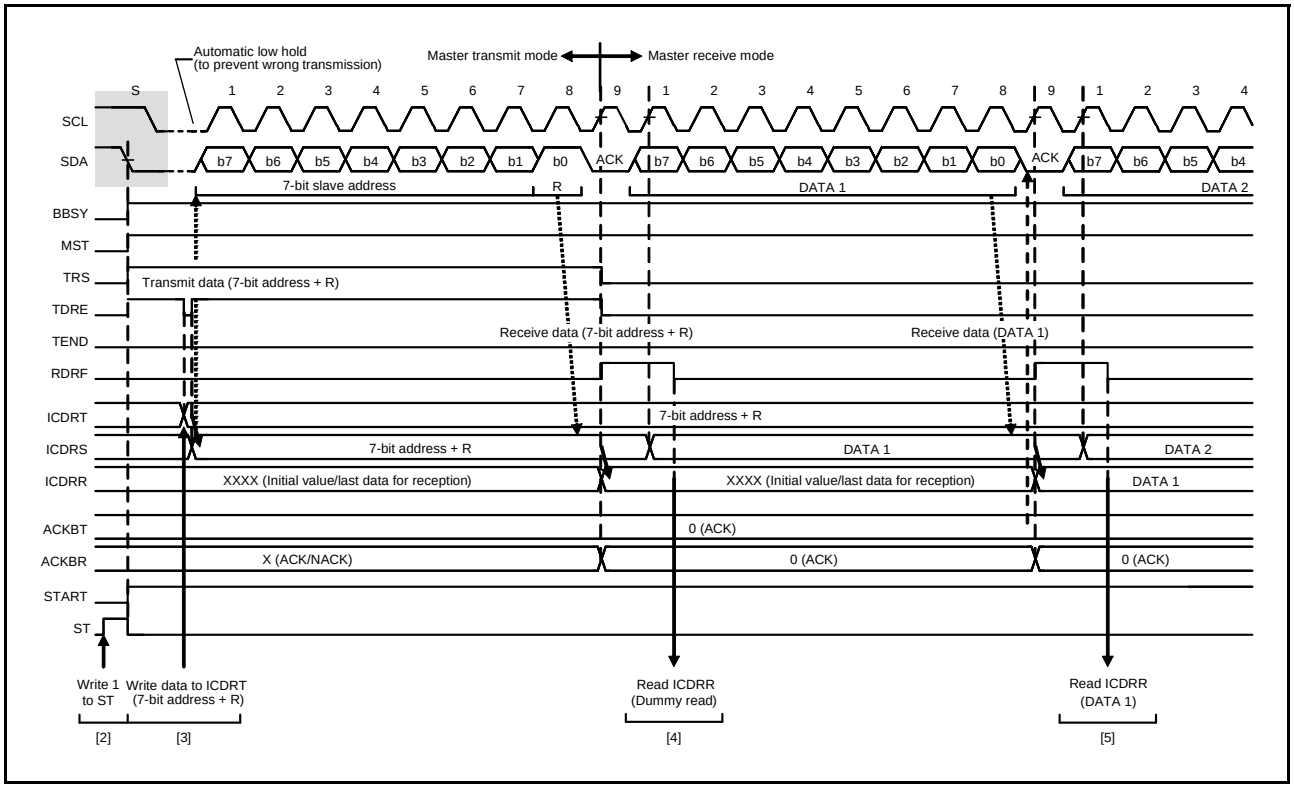


Figure 29.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

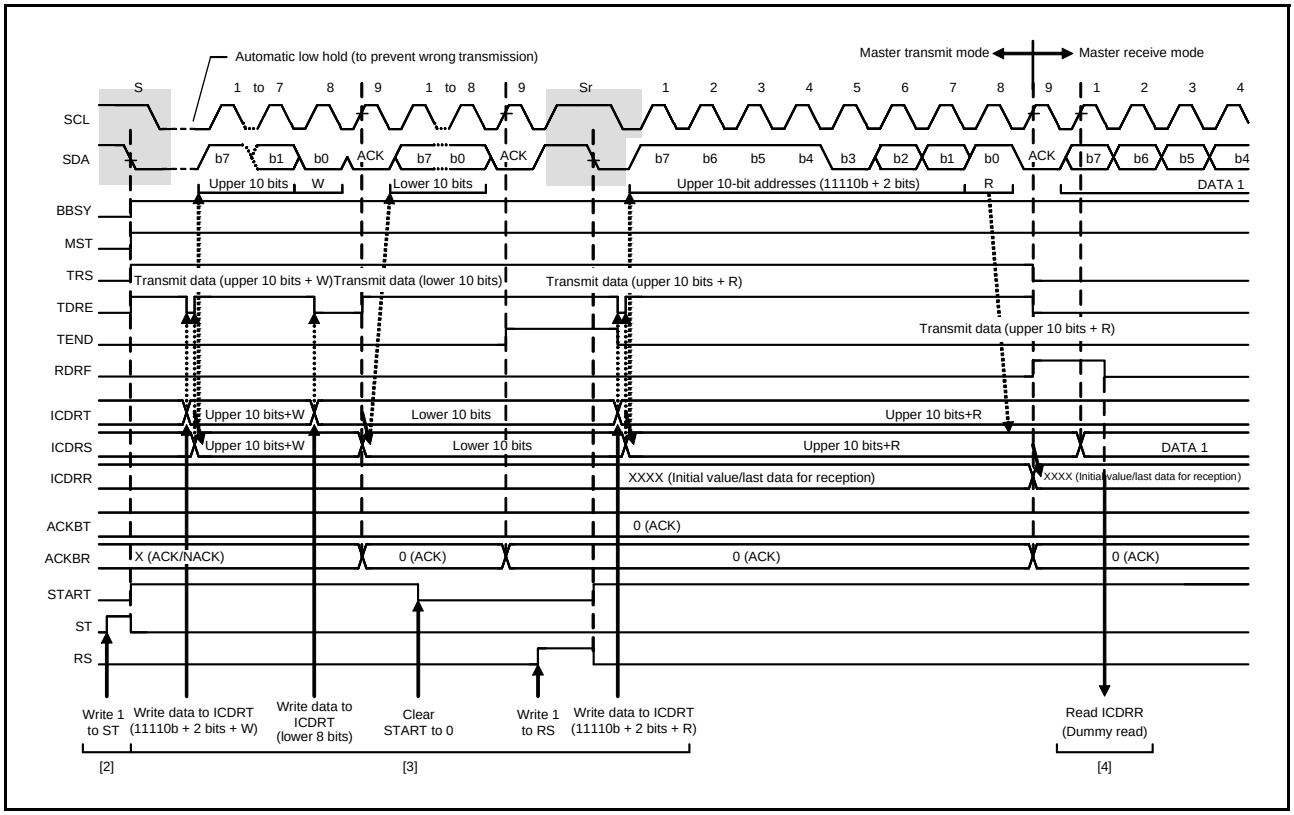


Figure 29.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

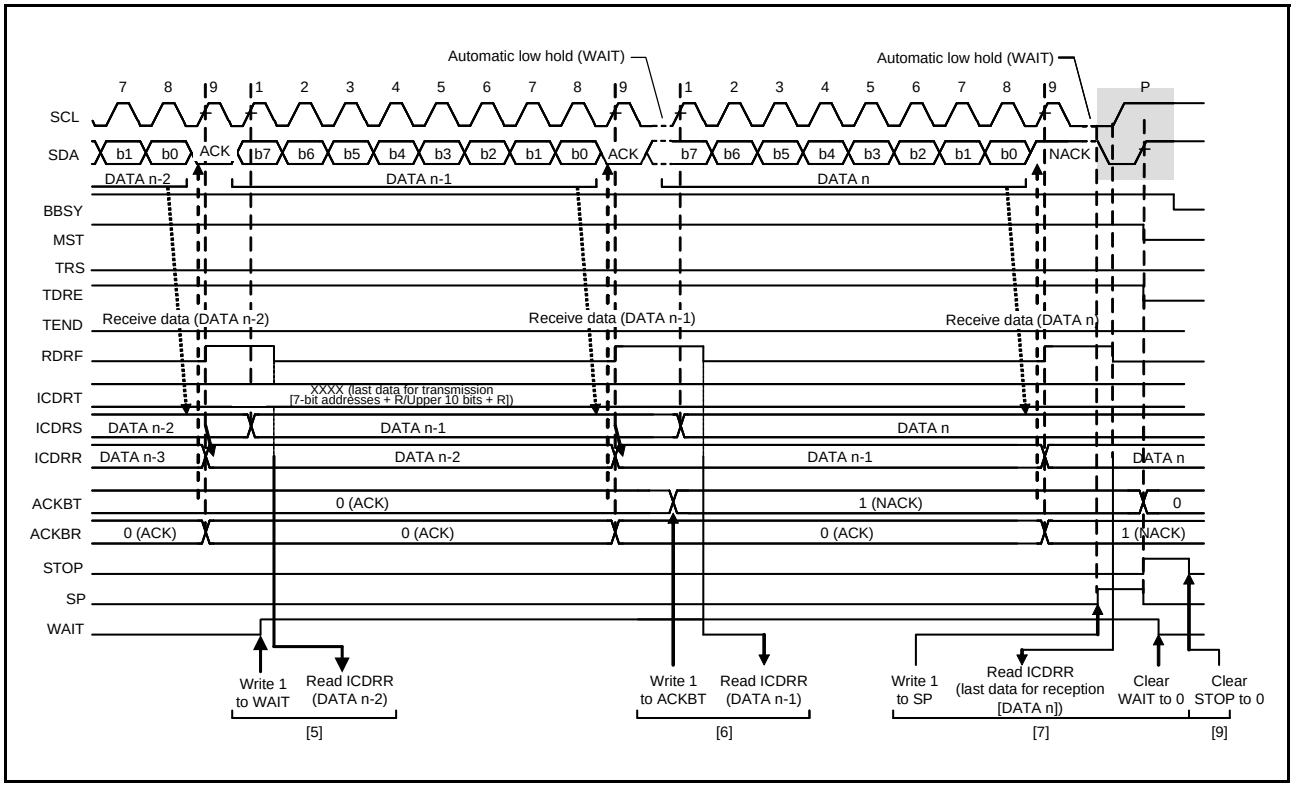
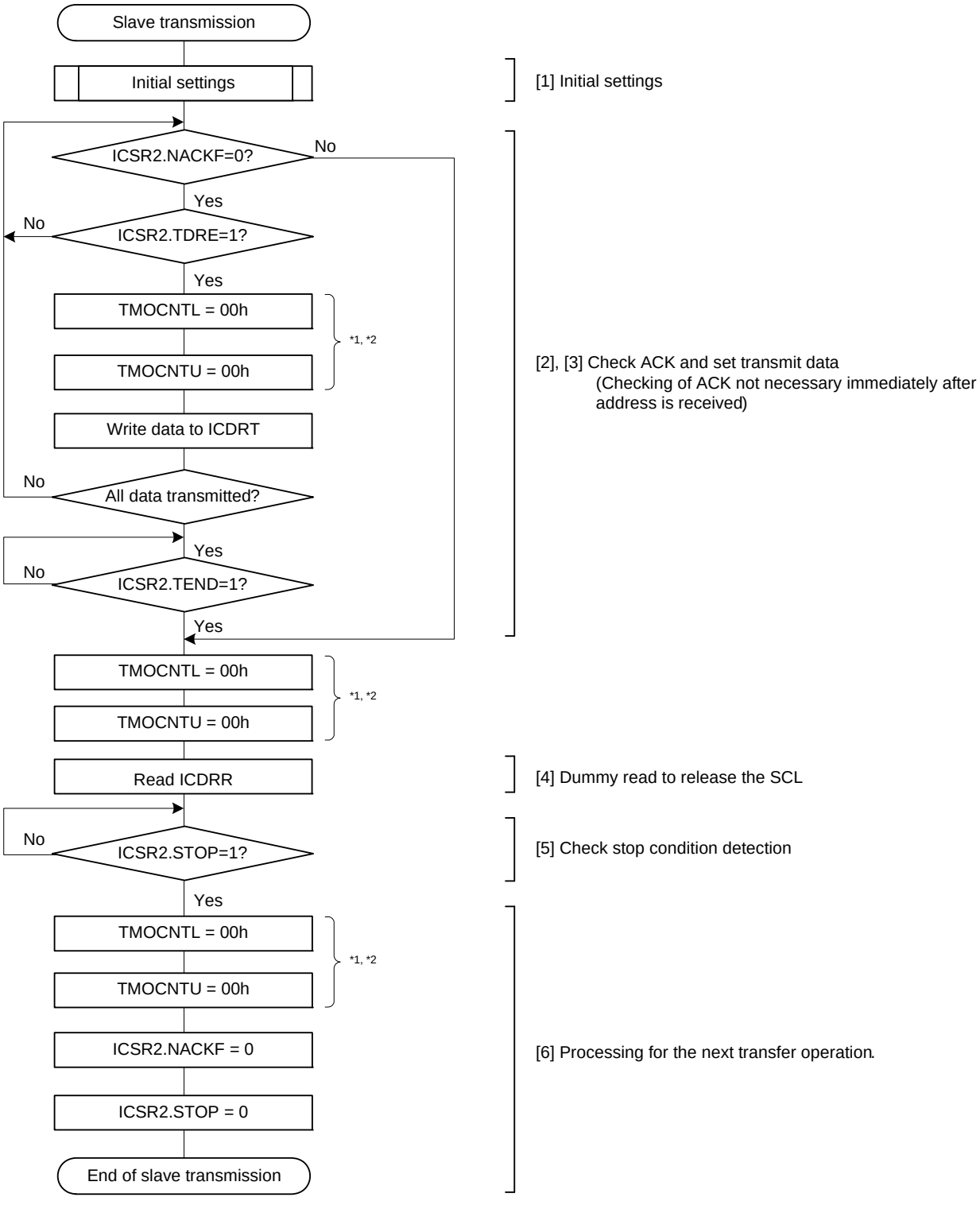


Figure 29.14 Master Receive Operation Timing (3) (when RDRFS = 0)

29.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.
Figure 29.15 shows an example of usage of slave transmission and Figure 29.16 and Figure 29.17 show the timing of operations in slave transmission.
The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in Figure 29.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 TRS bit and the TDRE flag in ICSR2 to 1.
(3) After the ICSR2.TEND 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 unit 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 SCL 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 SCL line.
(6) Upon detecting the stop condition, the RIIC automatically clears 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, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.



Note 1. Those procedures need to be added only when using timeout function. Not needed when not using timeout function.
Note 2. The diagram above is for 8-bit access. For 16-bit access, set 0000h to the address listed in Table 29.3, Register Allocation for 16-Bit Access.

Figure 29.15 Example of Slave Transmission Flowchart

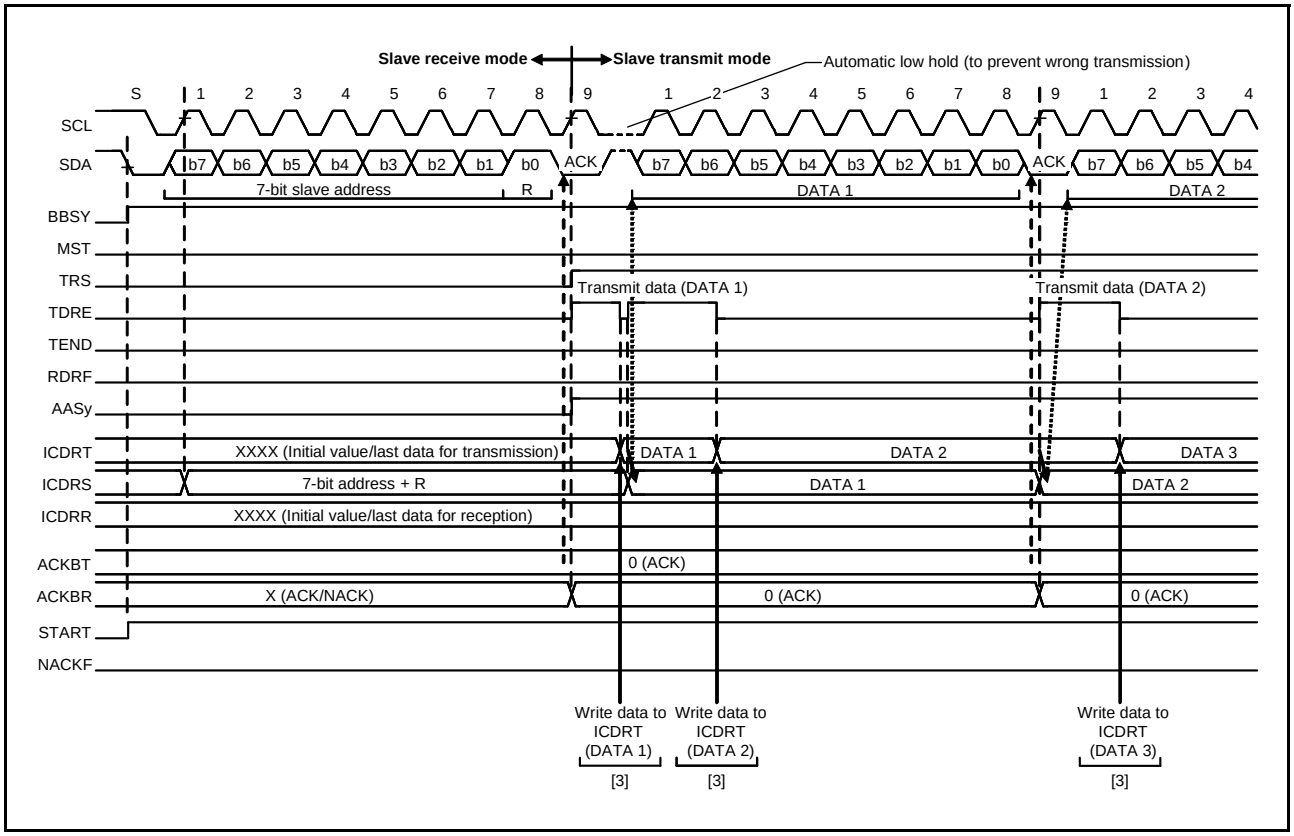


Figure 29.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

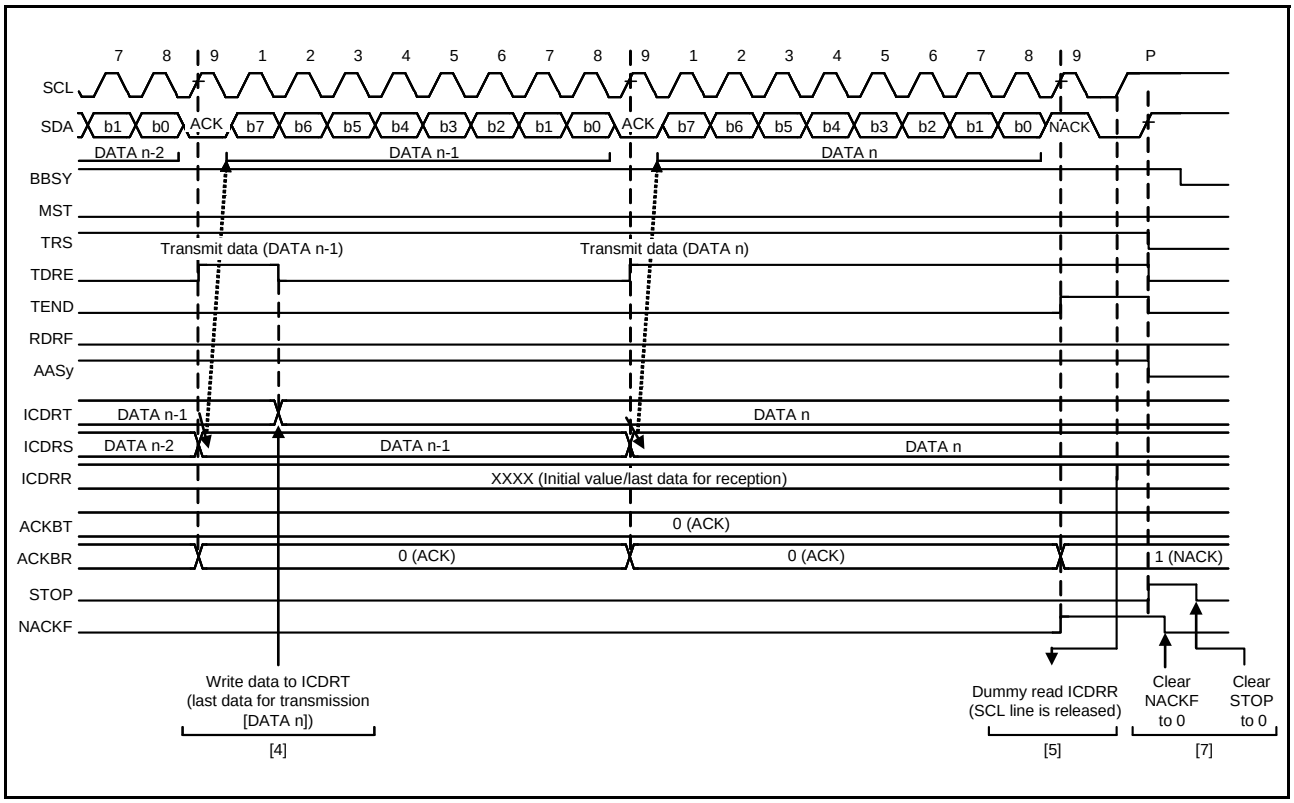


Figure 29.17 Slave Transmit Operation Timing (2)

29.3.6 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 29.18 shows an example of usage of slave reception and Figure 29.19 and Figure 29.20 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in Figure 29.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 eight bits when the 10-bit address format is selected).

(4) When ICDRR is read, the RIIC automatically clears 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 SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCL 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, clear the ICSR2.STOP flag to 0 for the next transfer operation.

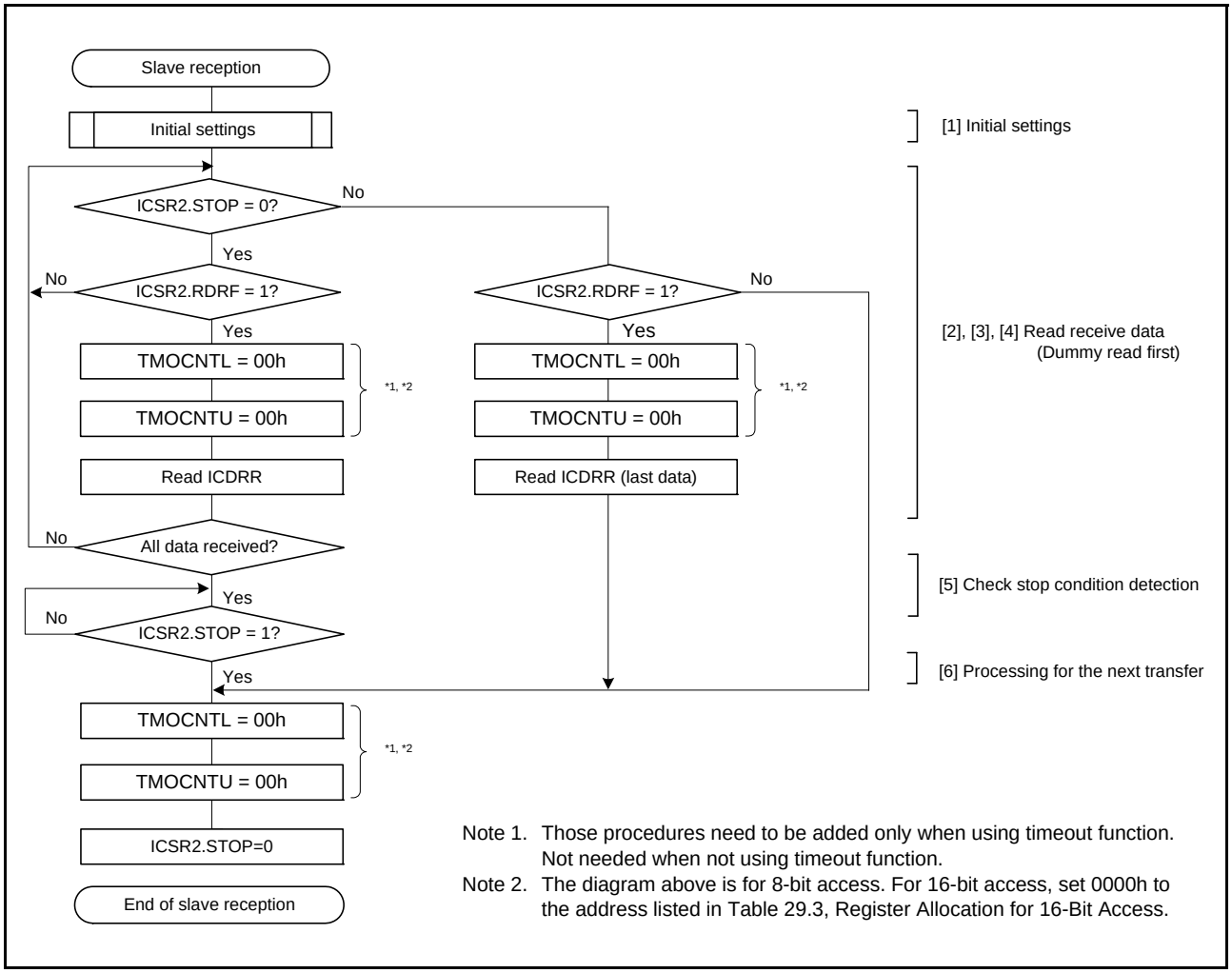


Figure 29.18 Example of Slave Reception Flowchart

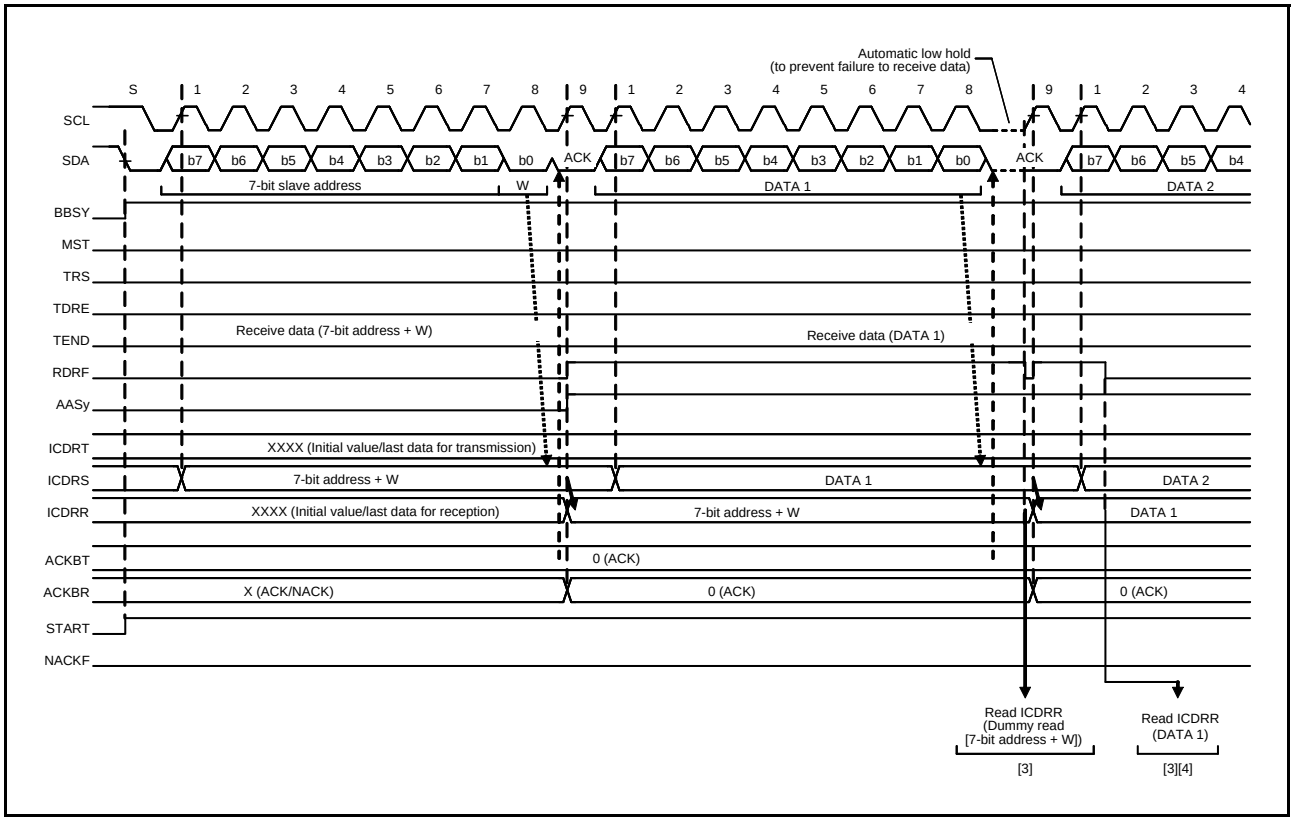


Figure 29.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

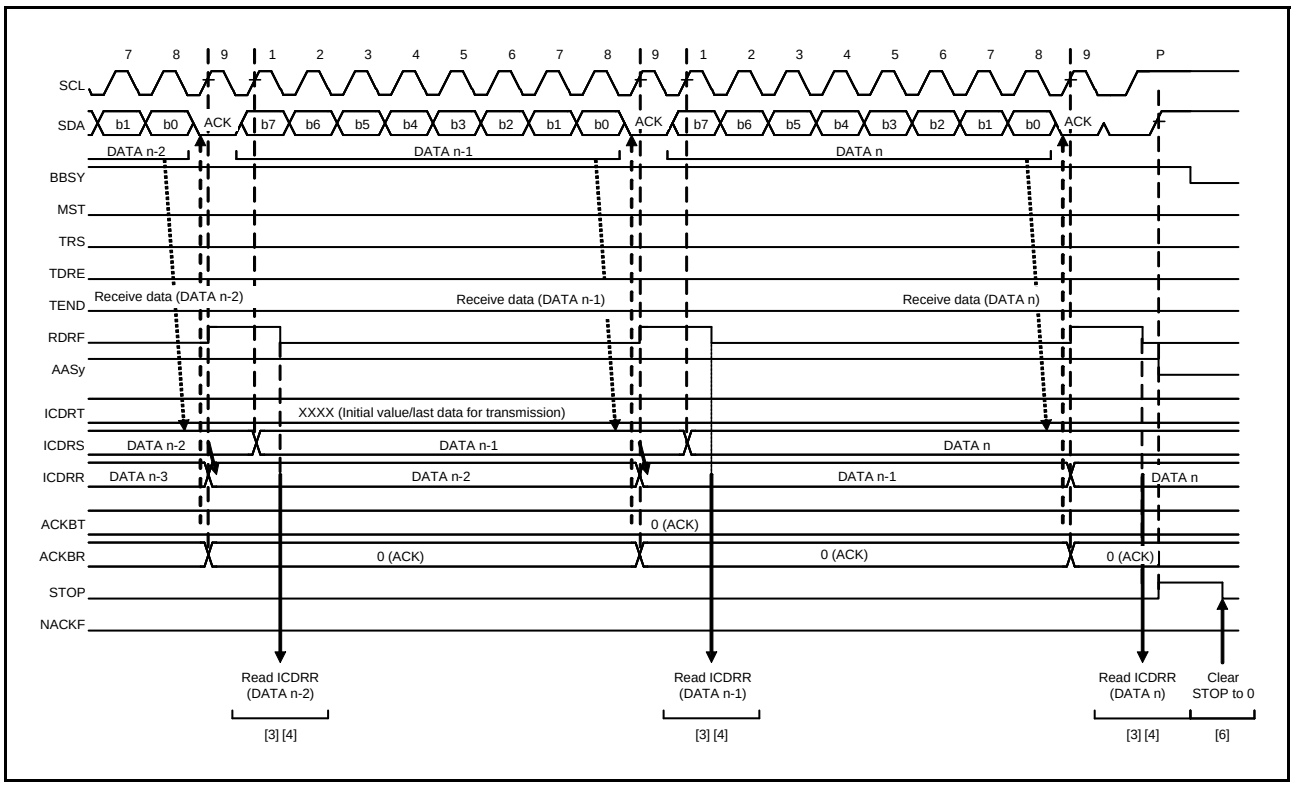


Figure 29.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

29.4 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal 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 SCL line while in master mode.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in ICBRH, and the level on the SCL 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 SCL 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 SCL 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 SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL 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.

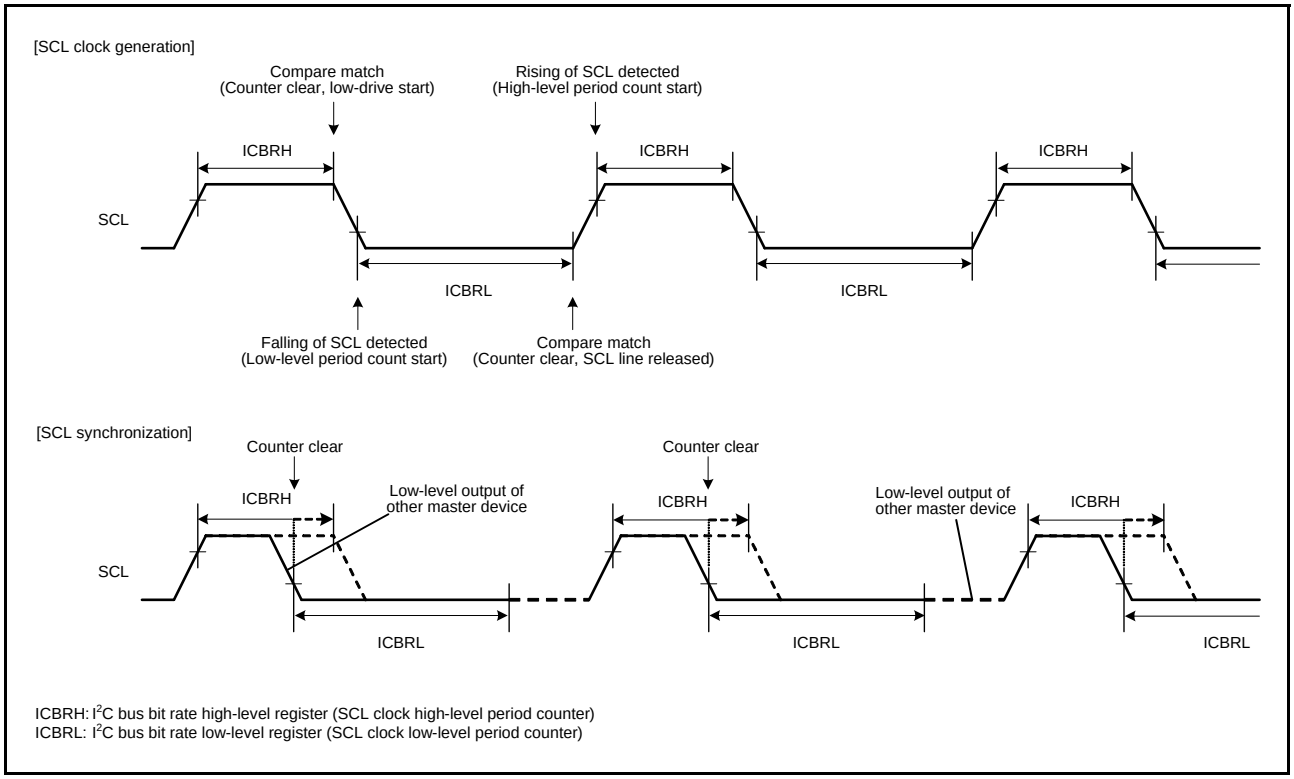


Figure 29.21 Generation and Synchronization of the SCL Signal from the RIIC

29.5 Facility for Delaying SDA Output

The RIIC module incorporates a facility 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 facility, 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) signal 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 facility 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 facility 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.

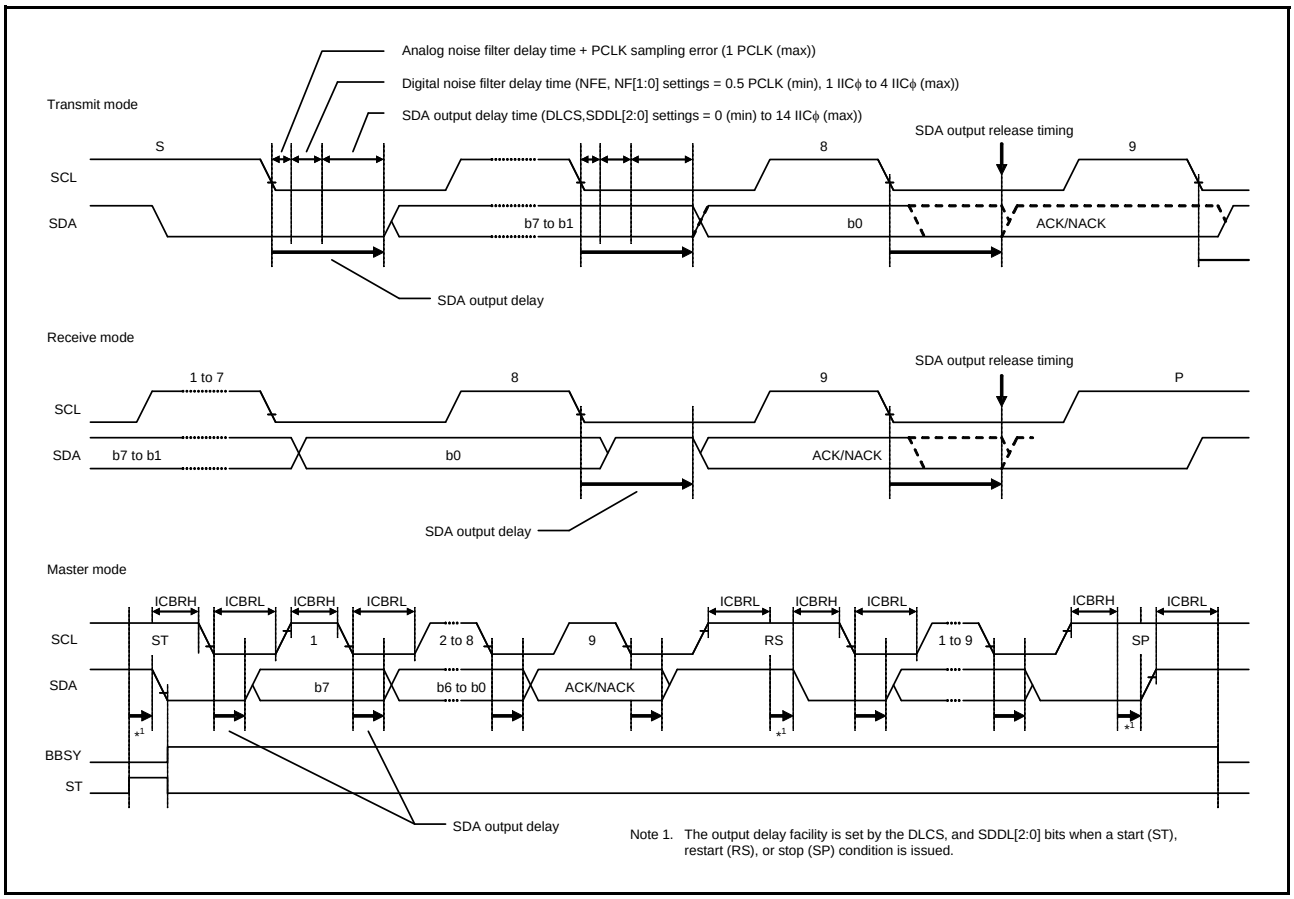


Figure 29.22 SDA Output Delay Facility

29.6 Digital Noise-Filter Circuits

The states of the SCL and SDA pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 29.23 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 SCL pin (or SDA 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 clearing the NFE bit in ICFER) and use only the analog noise-filter circuit.

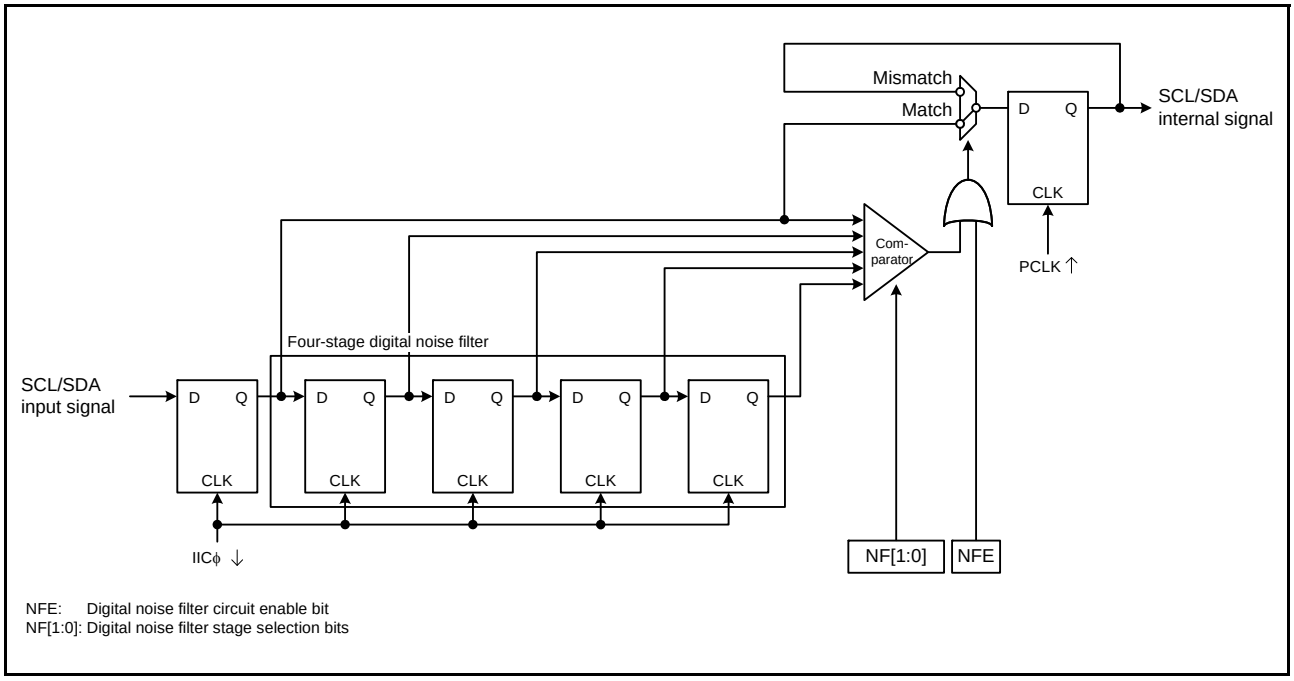


Figure 29.23 Block Diagram of Digital Noise Filter Circuit

29.7 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.

29.7.1 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 falling 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 29.24 to Figure 29.26 show the AASy flag set timing in three cases.

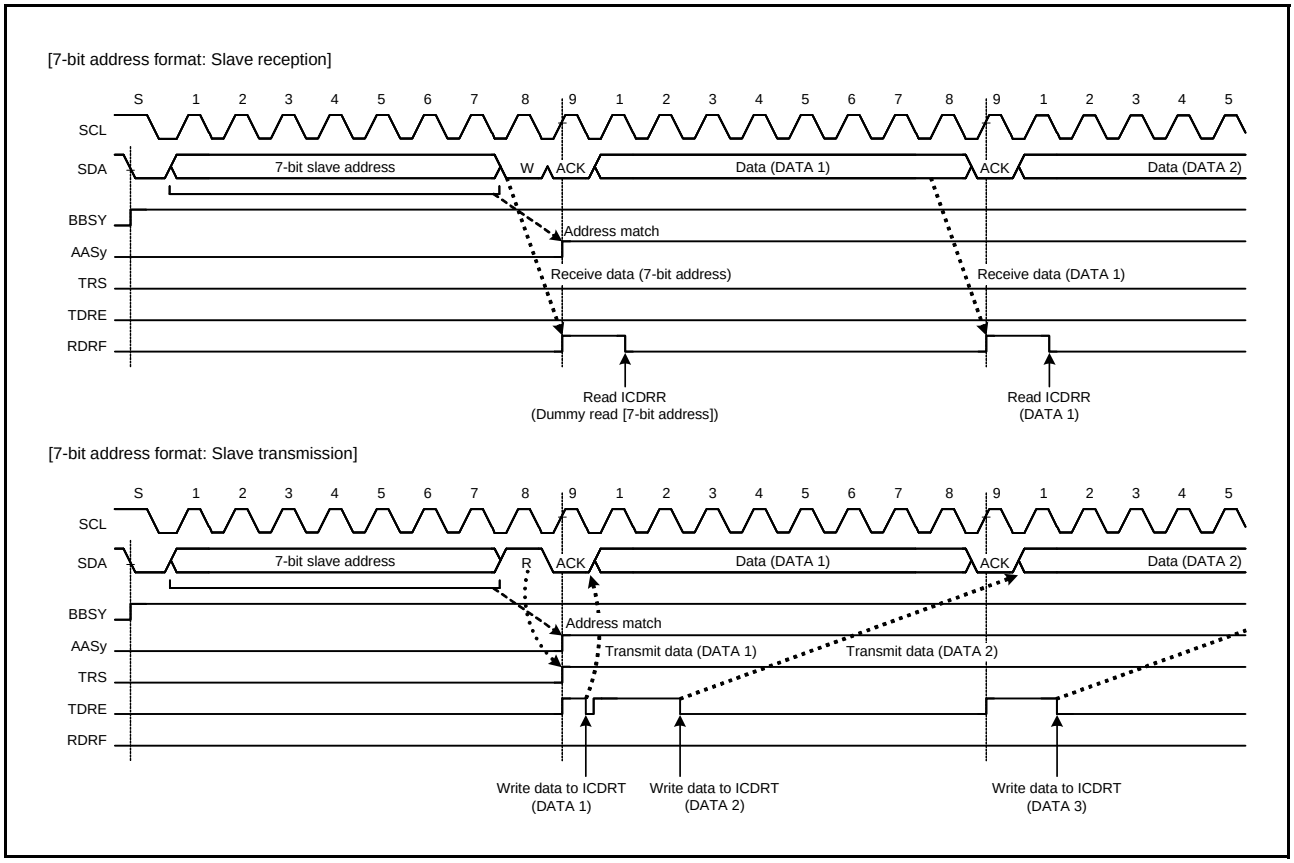


Figure 29.24 AASy Flag Set Timing with 7-Bit Address Format Selected

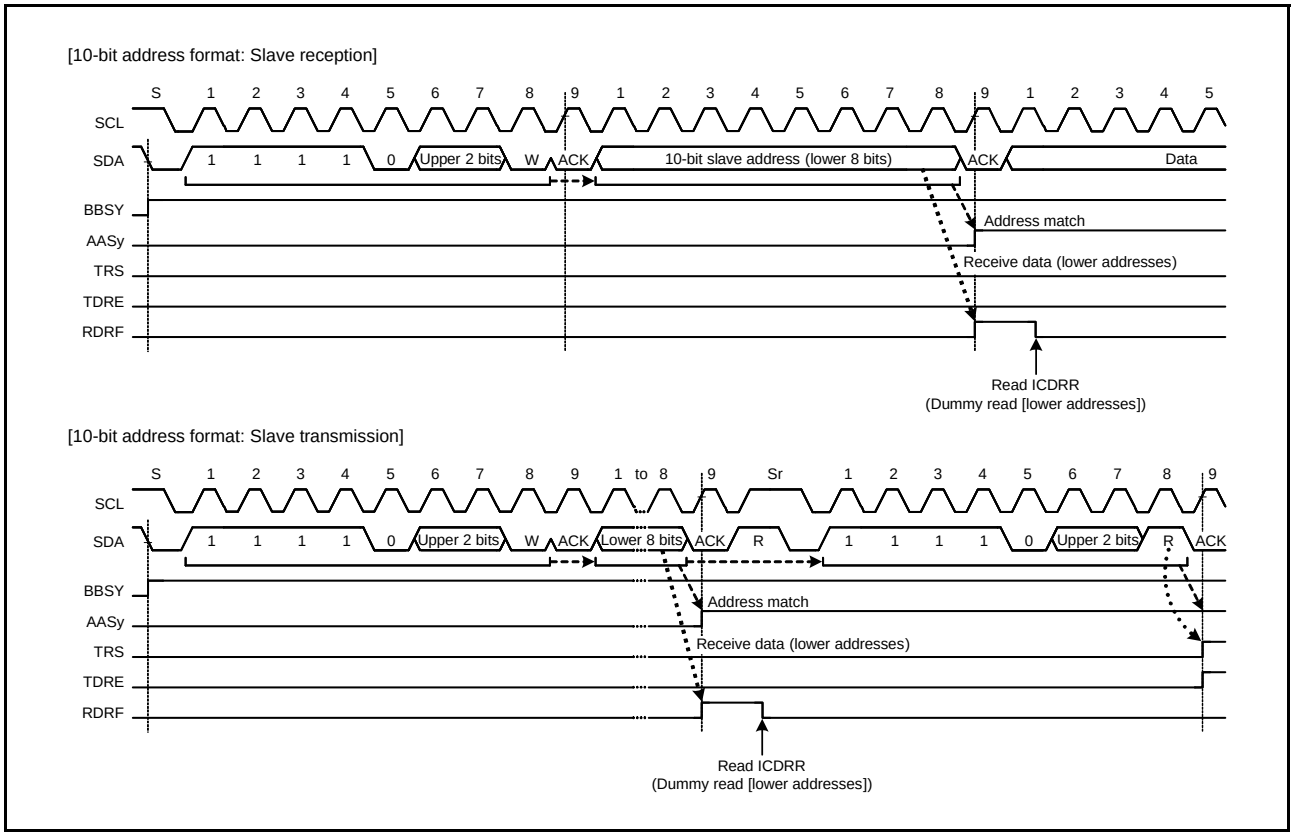


Figure 29.25 AASy Flag Set Timing with 10-Bit Address Format Selected

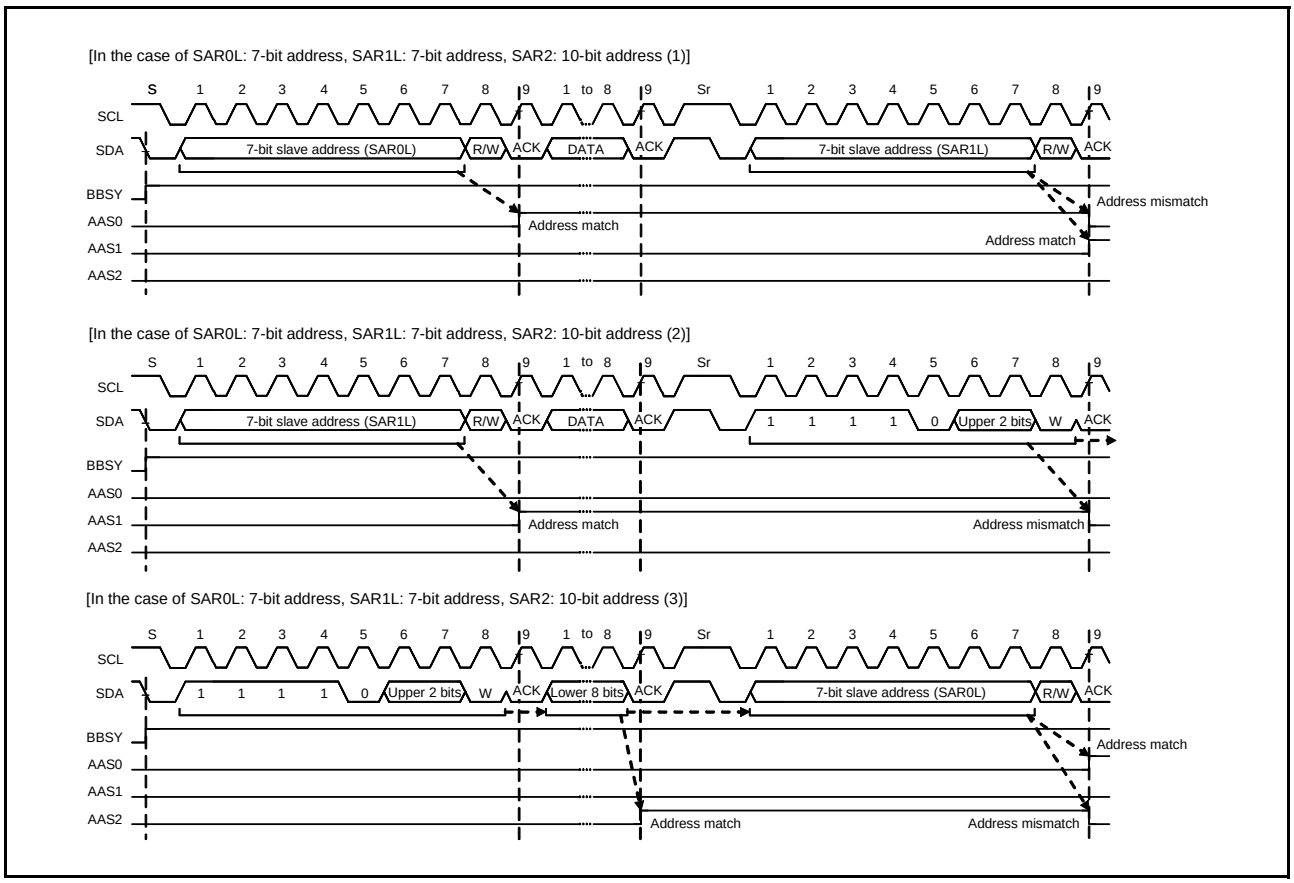


Figure 29.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

29.7.2 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 falling 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.

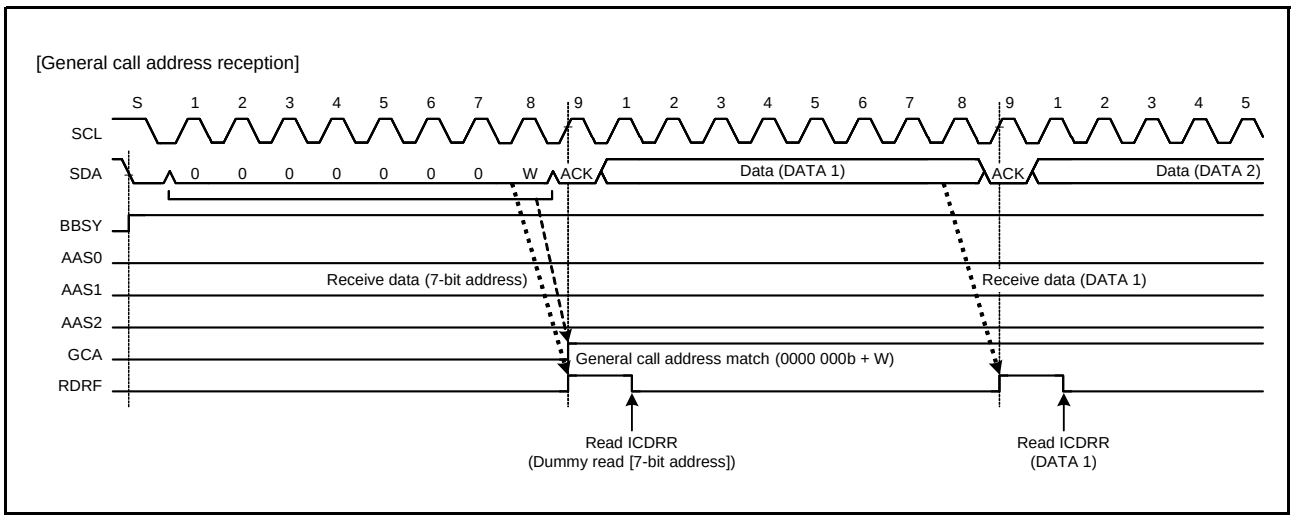


Figure 29.27 Timing of GCA Flag Setting during Reception of General Call Address

29.7.3 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 ninth 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 clears 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 (three 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.

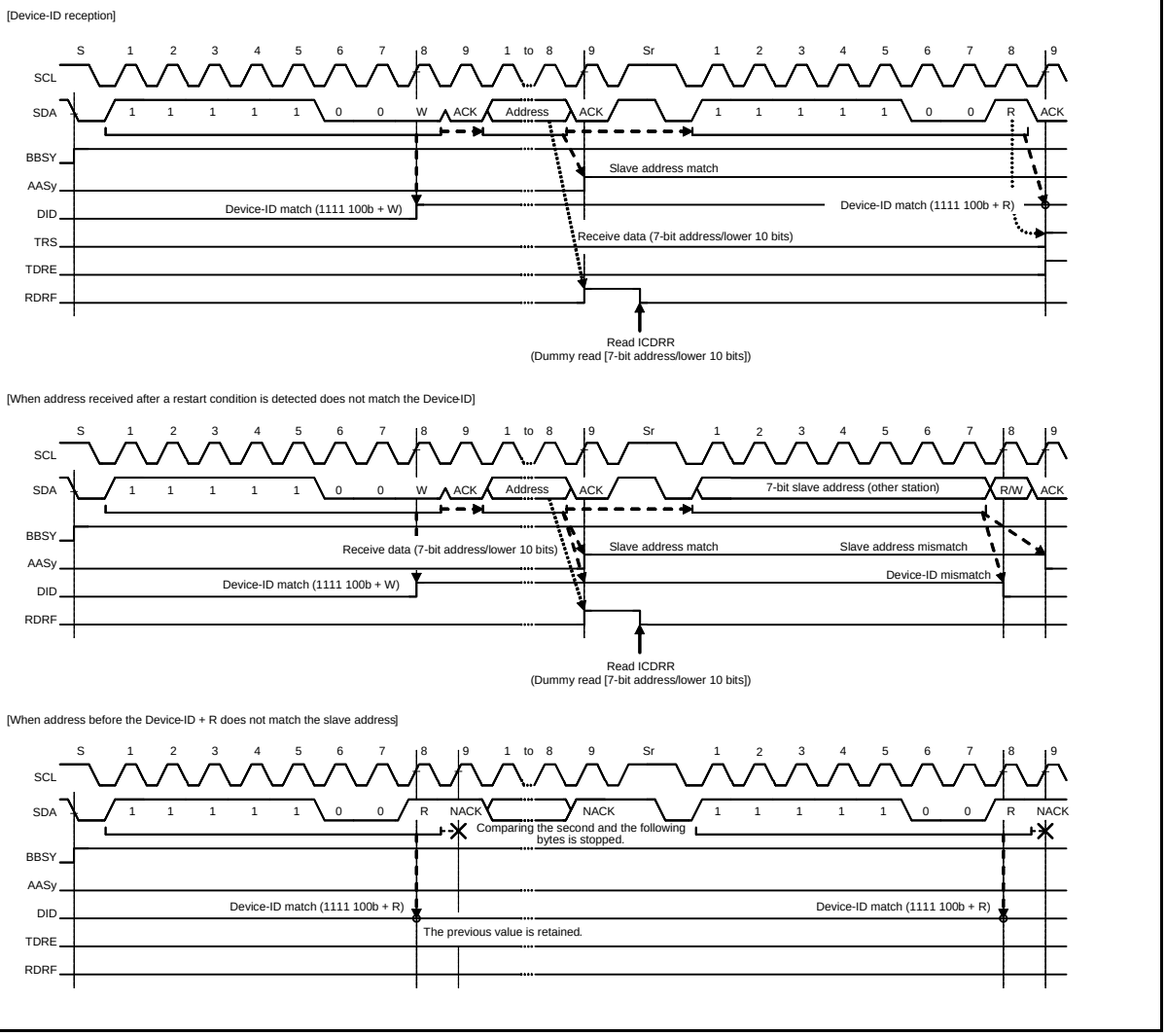


Figure 29.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

29.7.4 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 falling 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 transmit data empty interrupt (TXI) 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.

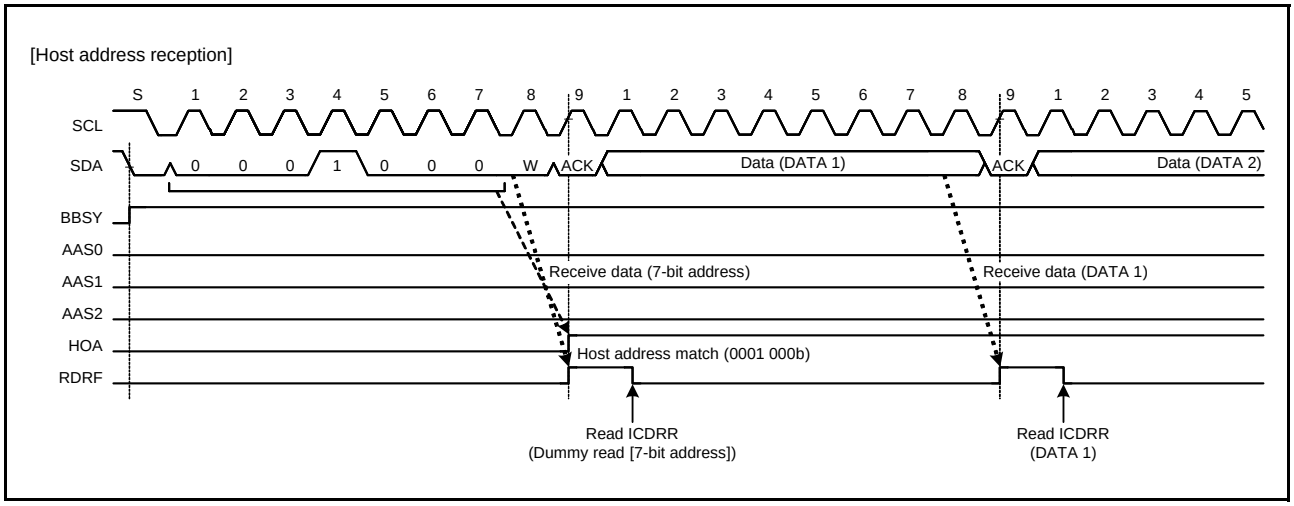


Figure 29.29 HOA Flag Set Timing during Reception of Host Address

29.8 Automatically Low-Hold Function for SCL

29.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCL 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 of one clock cycle 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

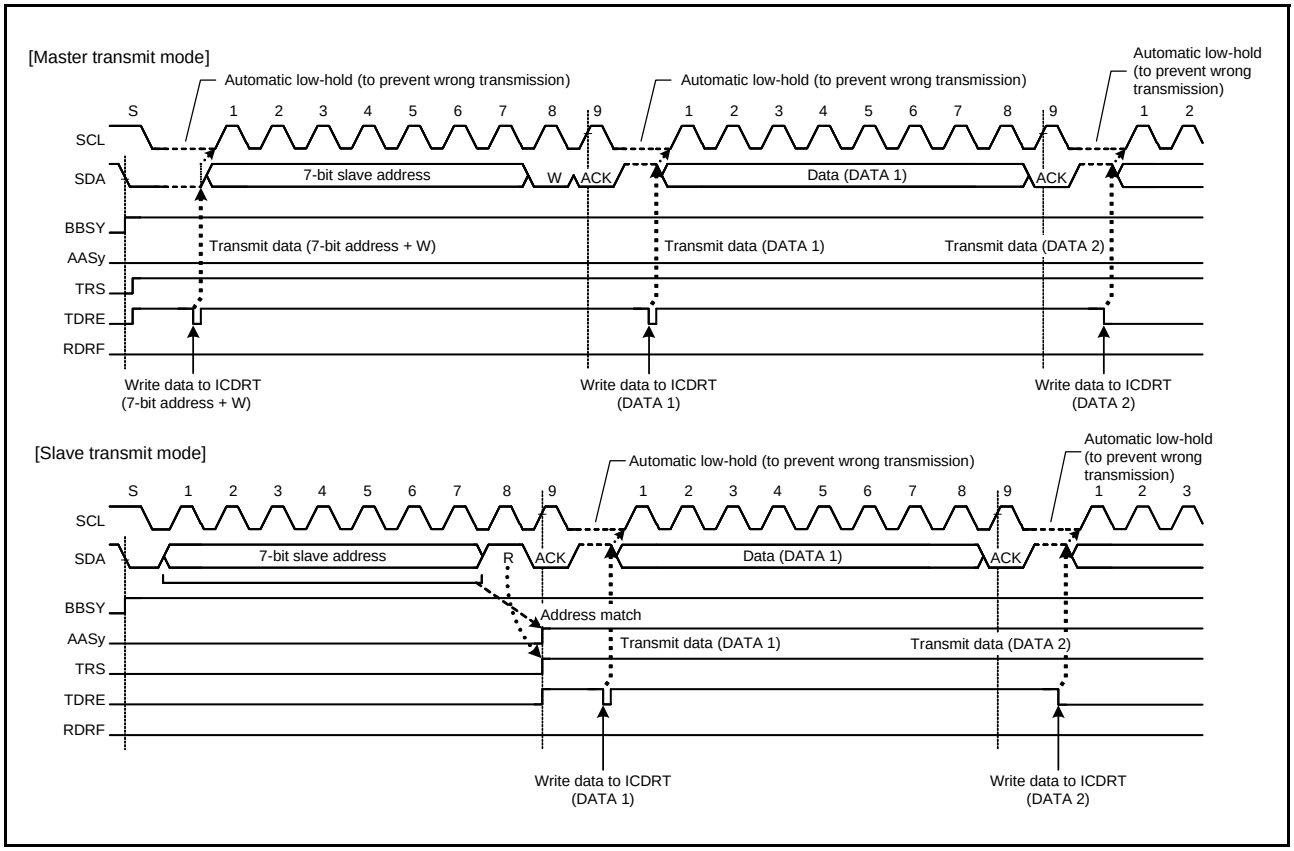


Figure 29.30 Automatic Low-Hold Operation in Transmit Mode

29.8.2 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 SDA 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 clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

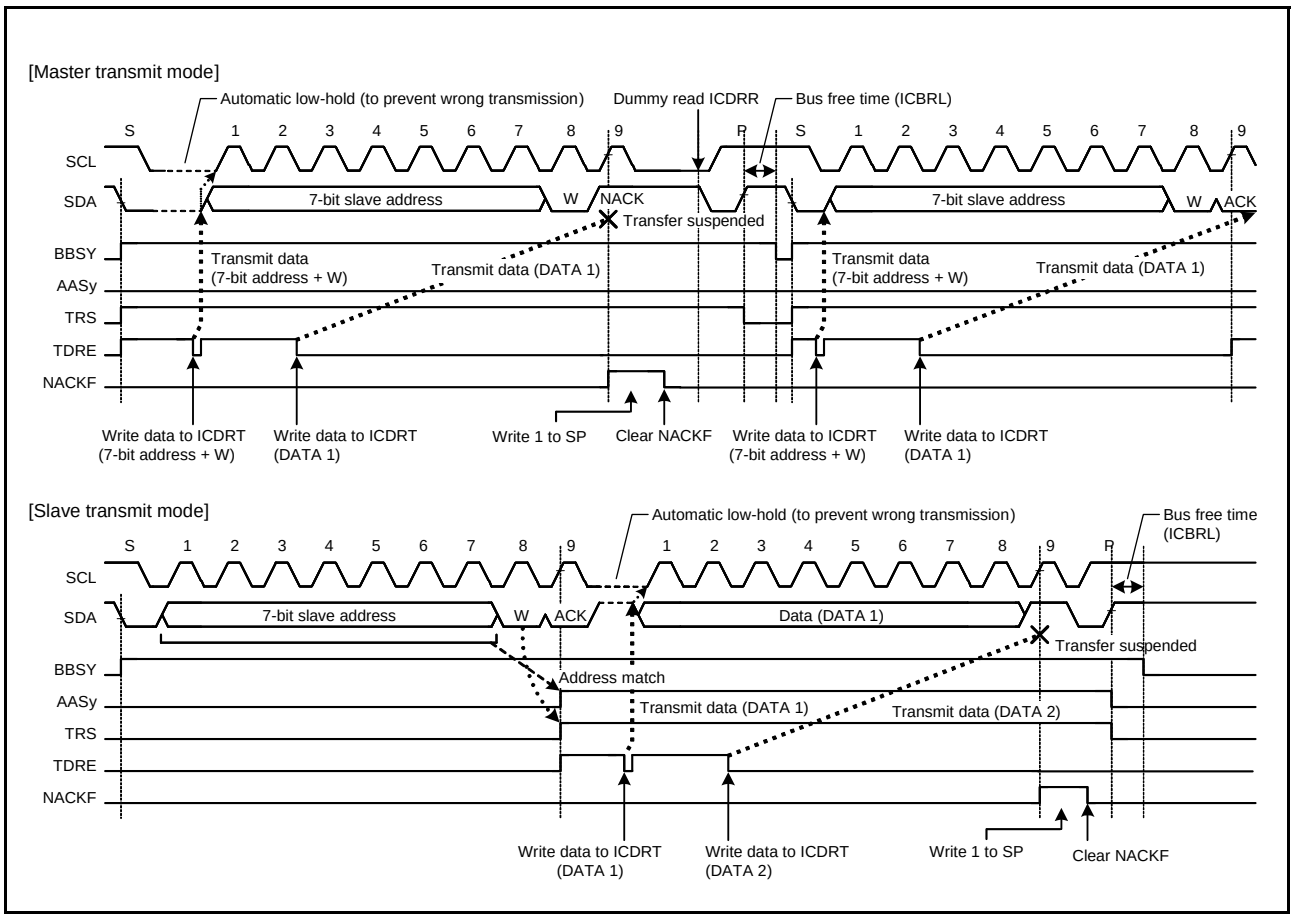


Figure 29.31 Suspension of Data Transfer when NACK is Received (NACKE = 1)

29.8.3 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 SCL 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 SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) One-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 one-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 SCL 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) One-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 one-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 SCL 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.

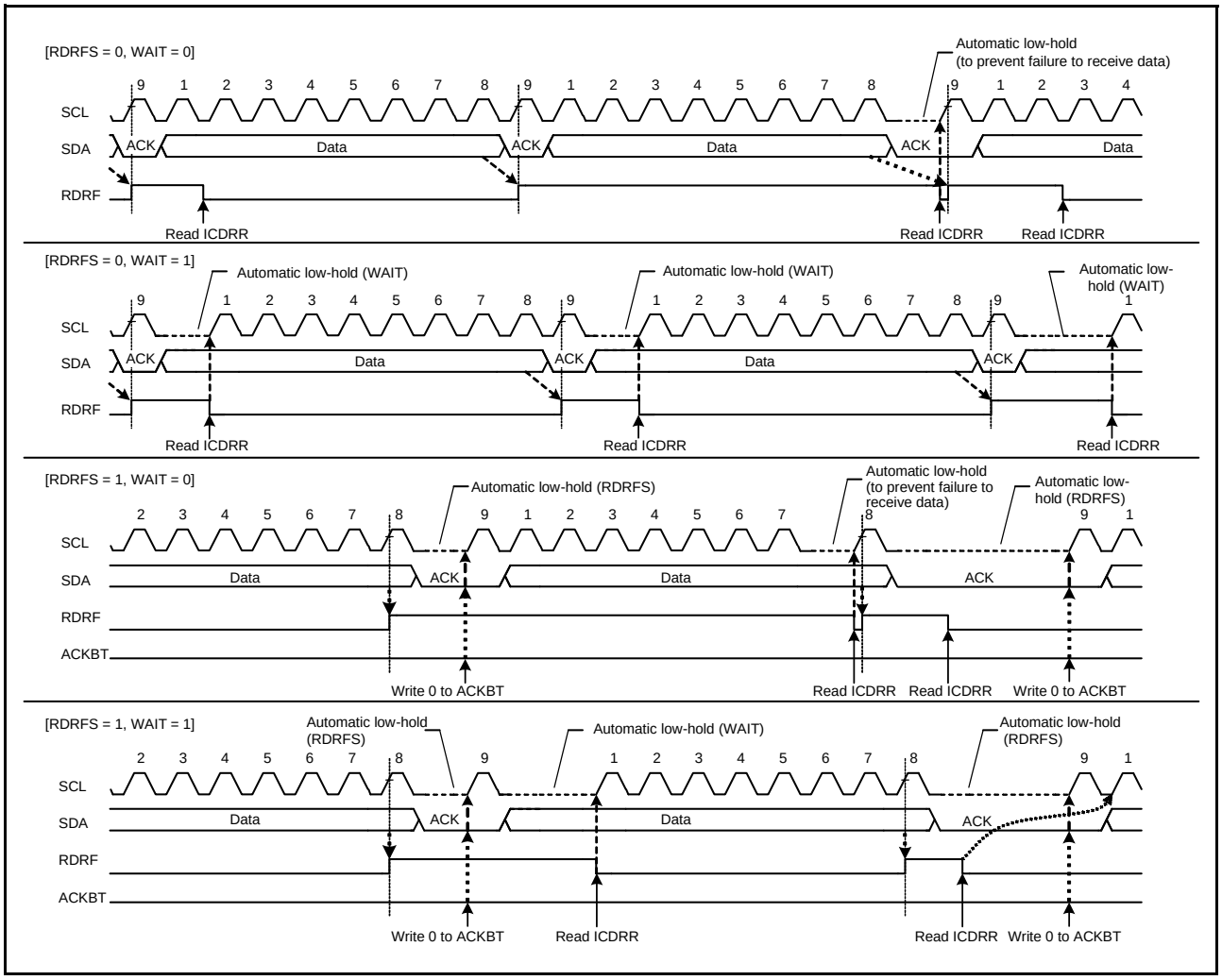


Figure 29.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

29.9 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.

29.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.
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 SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA 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 SDA line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was cleared 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 SDA line in master transmit mode (MST and TRS bits = 11b in ICCR2)

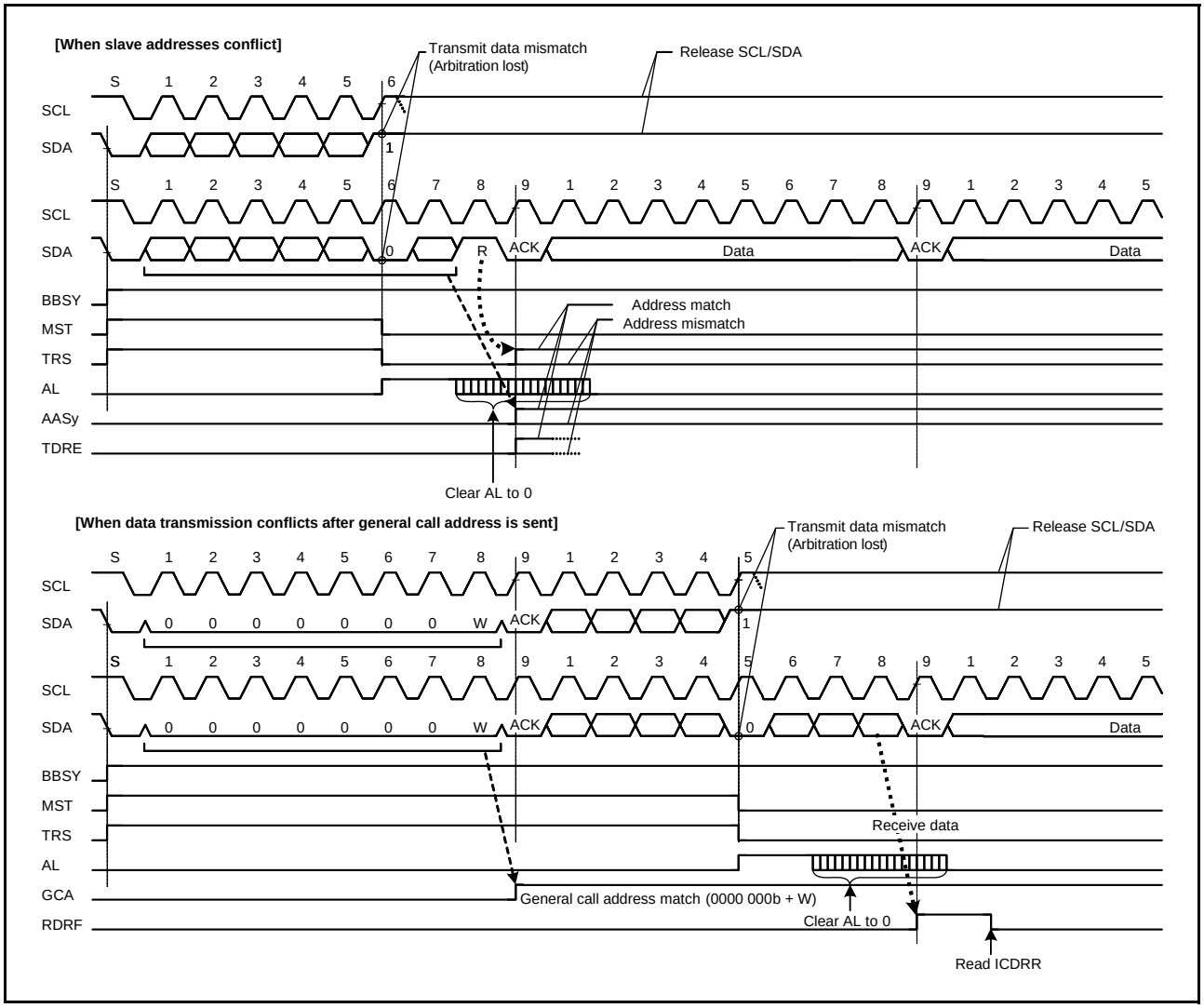


Figure 29.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

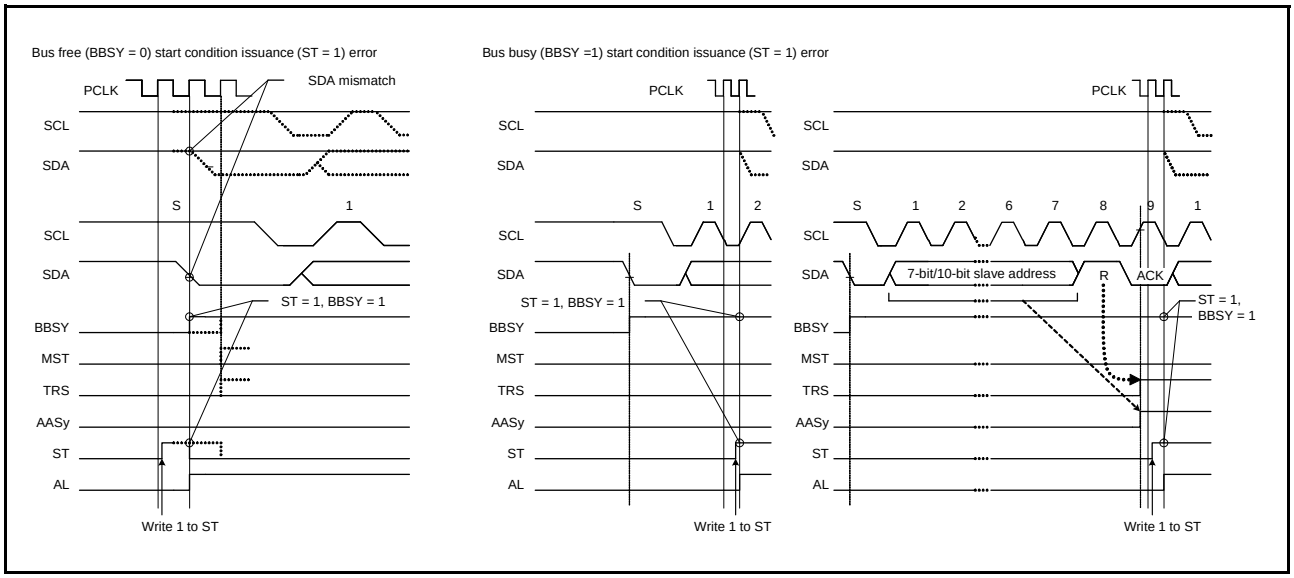


Figure 29.34 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

29.9.2 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 SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA 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 29.35 shows an example of arbitration-lost detection during transmission of NACK.

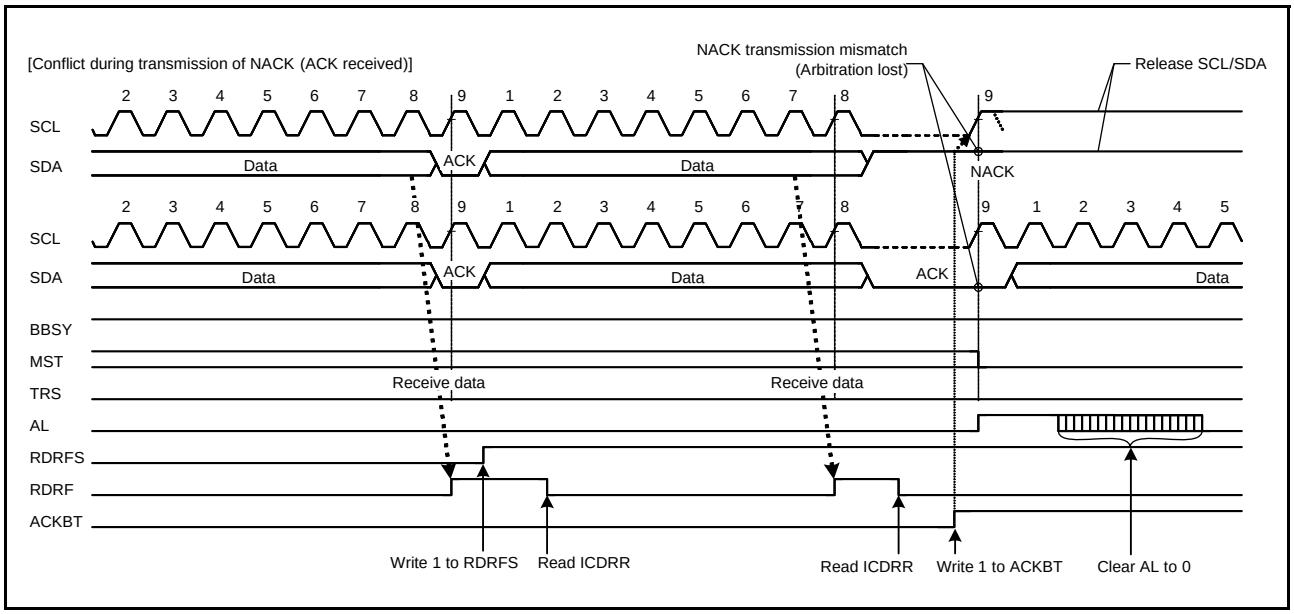


Figure 29.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

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 two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and 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 SDA line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

29.9.3 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 SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA 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 eliminates 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 SDA line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

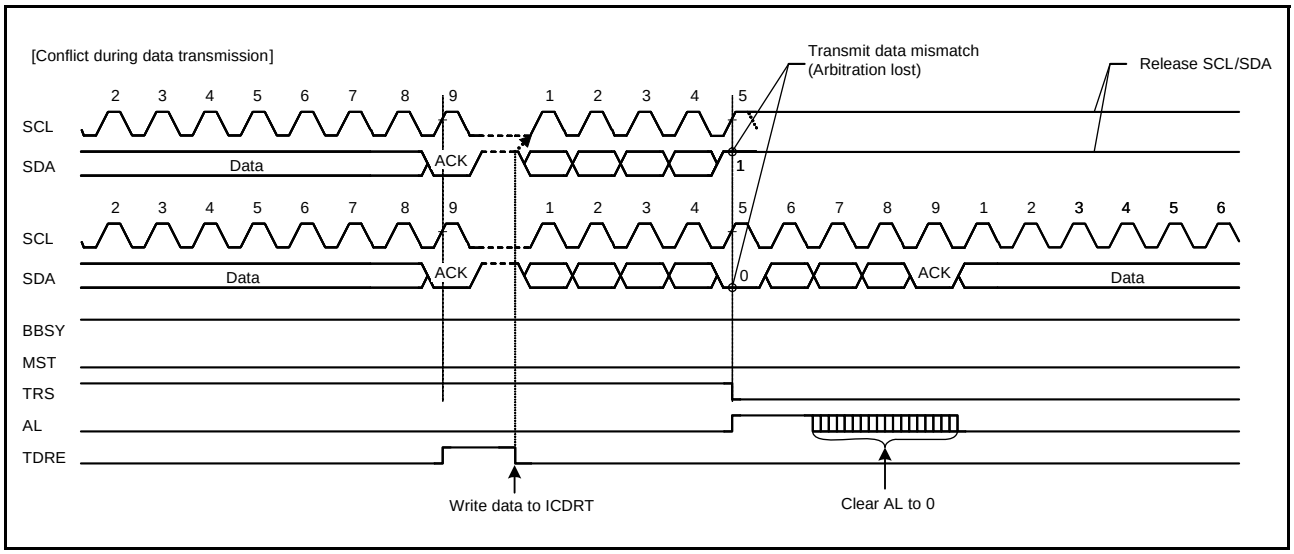


Figure 29.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

29.10 Start Condition/Restart Condition/Stop Condition Issuing Function

29.10.1 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). 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 SDA line low (high level to low level).
(2) Ensure the time set in ICBRH and the start condition hold time.
(3) Drive the SCL line low (high level to low level).
(4) Detect low level of the SCL line and ensure the low-level period of SCL line set in ICBRL.

29.10.2 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) 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 SDA line.
(2) Ensure the low-level period of SCL line set in ICBRL.
(3) Release the SCL line (low level to high level).
(4) Detect a high level of the SCL line and ensure the time set in ICBRL and the restart condition setup time.
(5) Drive the SDA line low (high level to low level).
(6) Ensure the time set in ICBRH and the restart condition hold time.
(7) Drive the SCL line low (high level to low level).
(8) Detect a low level of the SCL line and ensure the low-level period of SCL line set in ICBRL.

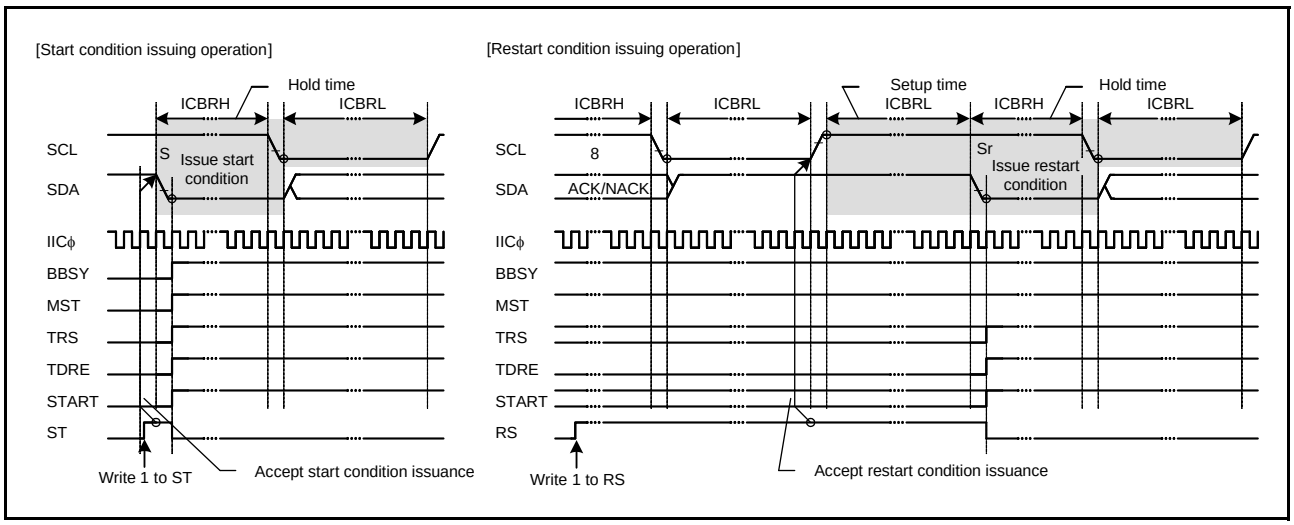


Figure 29.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

29.10.3 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) and the MST bit in ICCR2 is 1 (master mode).
A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in ICBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDA 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).

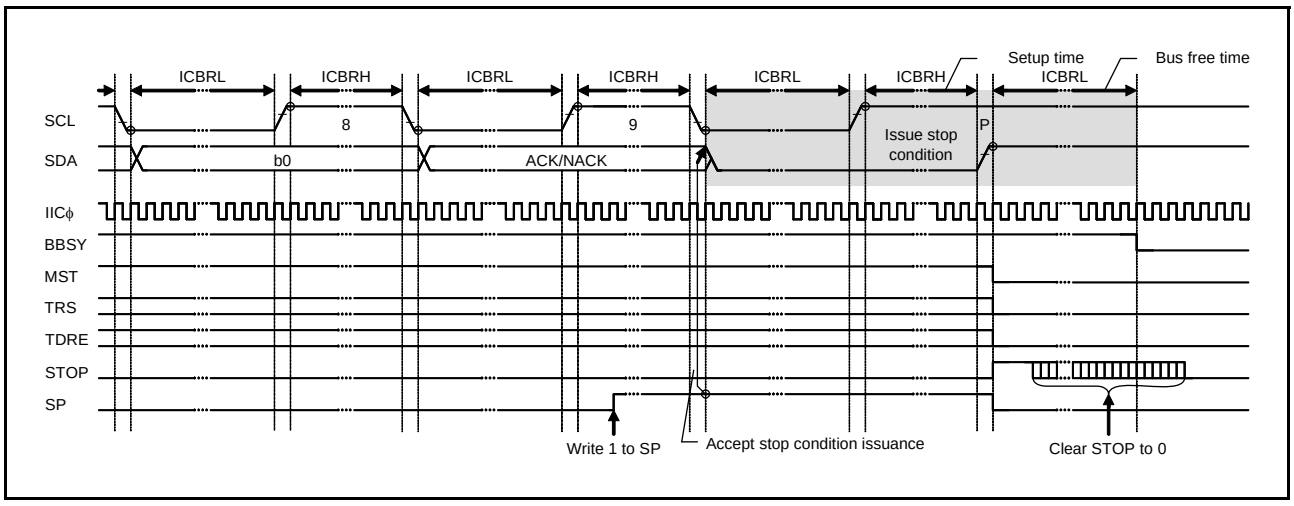


Figure 29.38 Stop Condition Issue Timing (SP Bit)

29.11 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 SCL line and/or SDA line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL 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, and the RIIC/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 SCL or SDA lines.

29.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is stuck low or high for a predetermined time. The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL 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 SCL 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 cleared to 0, the internal counter does not work.

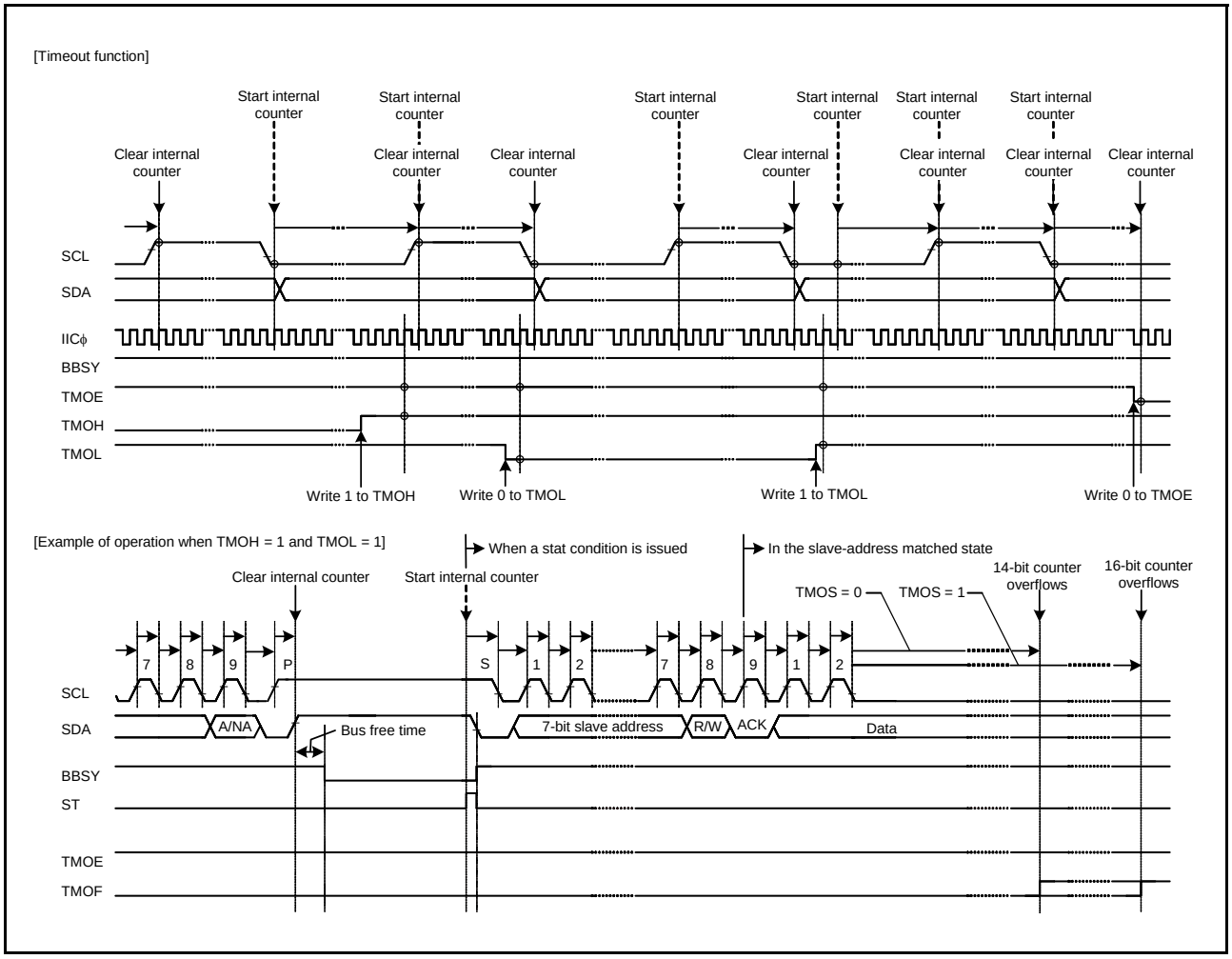


Figure 29.39 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

29.11.2 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 SDA 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 SDA 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) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA 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 cleared 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 SDA 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) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDA 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 cleared 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 ICCR1does not match the state of the SDA 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 SCL line low

Figure 29.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

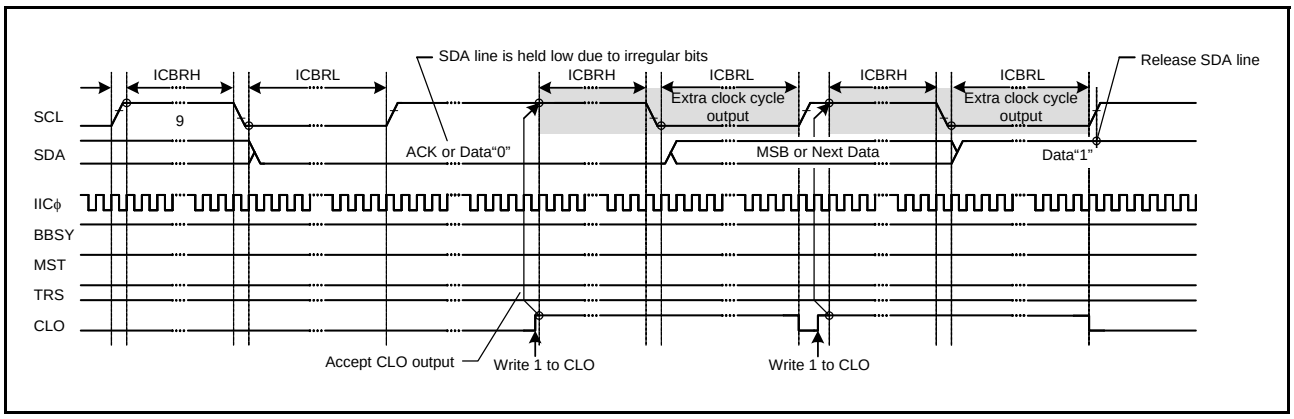


Figure 29.40 Extra SCL Clock Cycle Output Function (CLO Bit)

29.11.3 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 clear 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 SCL and SDA 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, see section 29.14, Reset States.

29.12 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. When 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.

29.12.1 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 SCL pin and SDA pin and make the SCL/SDA pin outputs high impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: TLOW: 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 SMBCLK 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 SMBCLK 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).

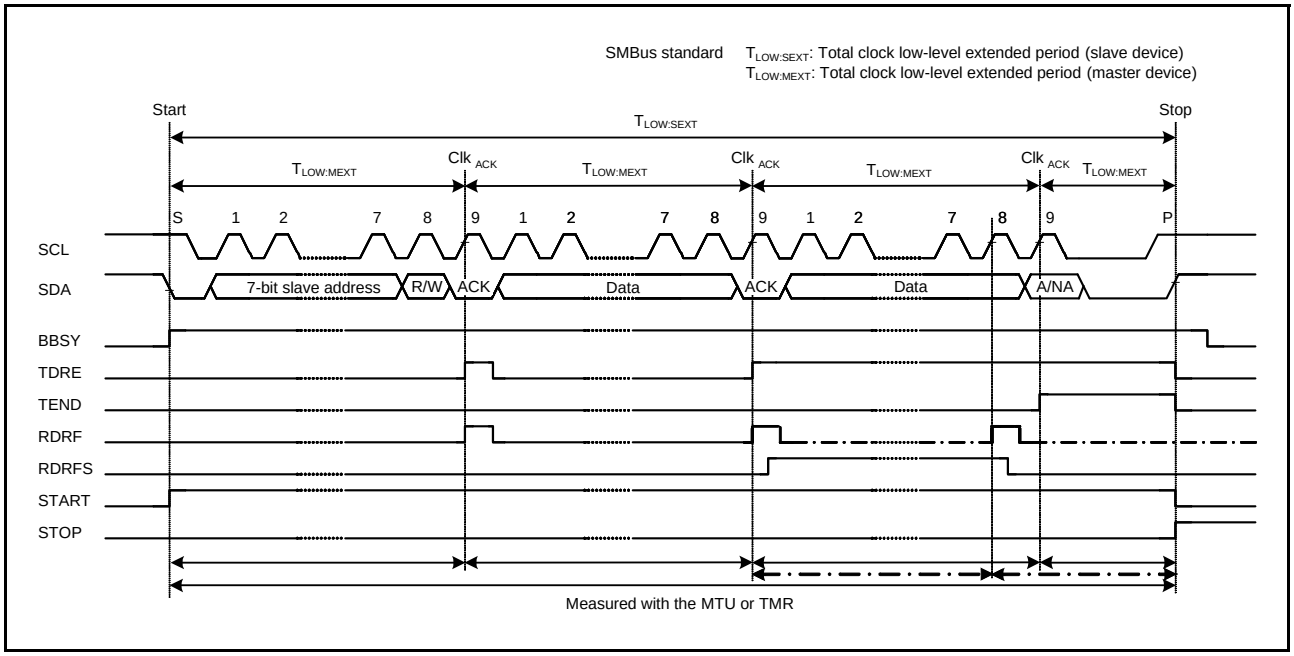


Figure 29.41 SMBus Timeout Measurement

29.12.2 Packet Error Code (PEC)

The RX220 Group 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, see section 31, CRC Calculator (CRC).

The PEC data in master transmit mode (master transmitter) 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 (master receiver) 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 SMBCLK clock cycle during reception of the final byte, and hold the SCL line low at the falling edge of the eighth clock cycle.

29.12.3 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the RX220 Group 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.

29.13 Interrupt Request

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 29.7 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of launching data transfer by the DTC or DMAC.

Table 29.7 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Launching	DMAC Launching	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	Receive Data Full	—	Possible	Possible		RDRF = 1 • RIE = 1
TXI	Transmit Data Empty	—	Possible	Possible		TDRE = 1 • TIE = 1
TEI	Transmit End	TEND	Not possible	Not possible	Low	TEND = 1 • TEIE = 1

Clear or mask the each flag during interrupt handling.

Notes on interrupt processing:

1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for TXI) is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
3. Since RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for RXI) is automatically cleared to 0 when data are read from ICDRR.
4. When using the TEI interrupt, clear the TEND flag in ICSR2 in the TEI interrupt processing.
   Note that the TEND flag in ICSR2 is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

29.13.1 Buffer Operation for TXI and RXI Interrupts

For the TXI and RXI interrupts, as well as the ICU.IRn.IR flag having the value 1 being a condition for issuing the interrupts, the interrupt request is retained within and not output by the ICU (the capacity for internally retaining the interrupts 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.

29.14 Reset States

The RIIC has chip reset, RIIC reset, and internal reset functions. Table 29.8 lists the scope of each reset and reset conditions.

Table 29.8 Reset Conditions

		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	Operation (retained)	Operation (retained)
	SCLO, SDAO		At a reset	At a reset		
	Others			Retained		
ICCR2	BBSY	At a reset	At a reset	Operation	Operation	Operation
	ST			At a reset	At a reset	Operation (retained)
	Others					At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Operation (retained)
	Others			Retained	Operation (retained)	
ICMR2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICMR3		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICFER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICIER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSR1		At a reset	At a reset	At a reset	Operation (retained)	At a reset
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Operation (retained)	At a reset
	START				Operation	
	STOP				Operation (retained)	Operation
	Others					Operation (retained)
SARL0, 1, 2 SARU0, 1, 2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICBRH, ICBRL		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRT		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRR		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRS		At a reset	At a reset	At a reset	Operation (retained)	Operation (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

29.15 Event Link Output

The event link controller (ELC) handles output of the following event signals.

(1) Transfer error event output

When a transfer error event occurs, the corresponding event signal can be output for another module via the ELC.

(2) Received-data full output

When a received data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmission-data empty output

When a transmission data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmission-completed output

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

29.15.1 Interrupt Handling and Event Linking

The RIIC module produces four kinds of interrupt: transfer-error (arbitration-lost detection, detection of NACK, detection of time-out, or detection of a stop condition) event, received data full, transmission data empty, and transmission completed 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 29.7.

29.16 Usage Notes

29.16.1 Setting Module Stop Function

Module stop state can be entered or canceled using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by clearing module stop state.
For details of module stop control register B, see section 11, Low Power Consumption.

29.16.2 Points to Note on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started (by setting the ICCR1.ICE bit to 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the ICU.IRn.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 ICU.IRn.IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Clear 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. Clear the ICU.IRn.IR flag to 0.

30. Serial Peripheral Interface (RSPI)

30.1 Overview

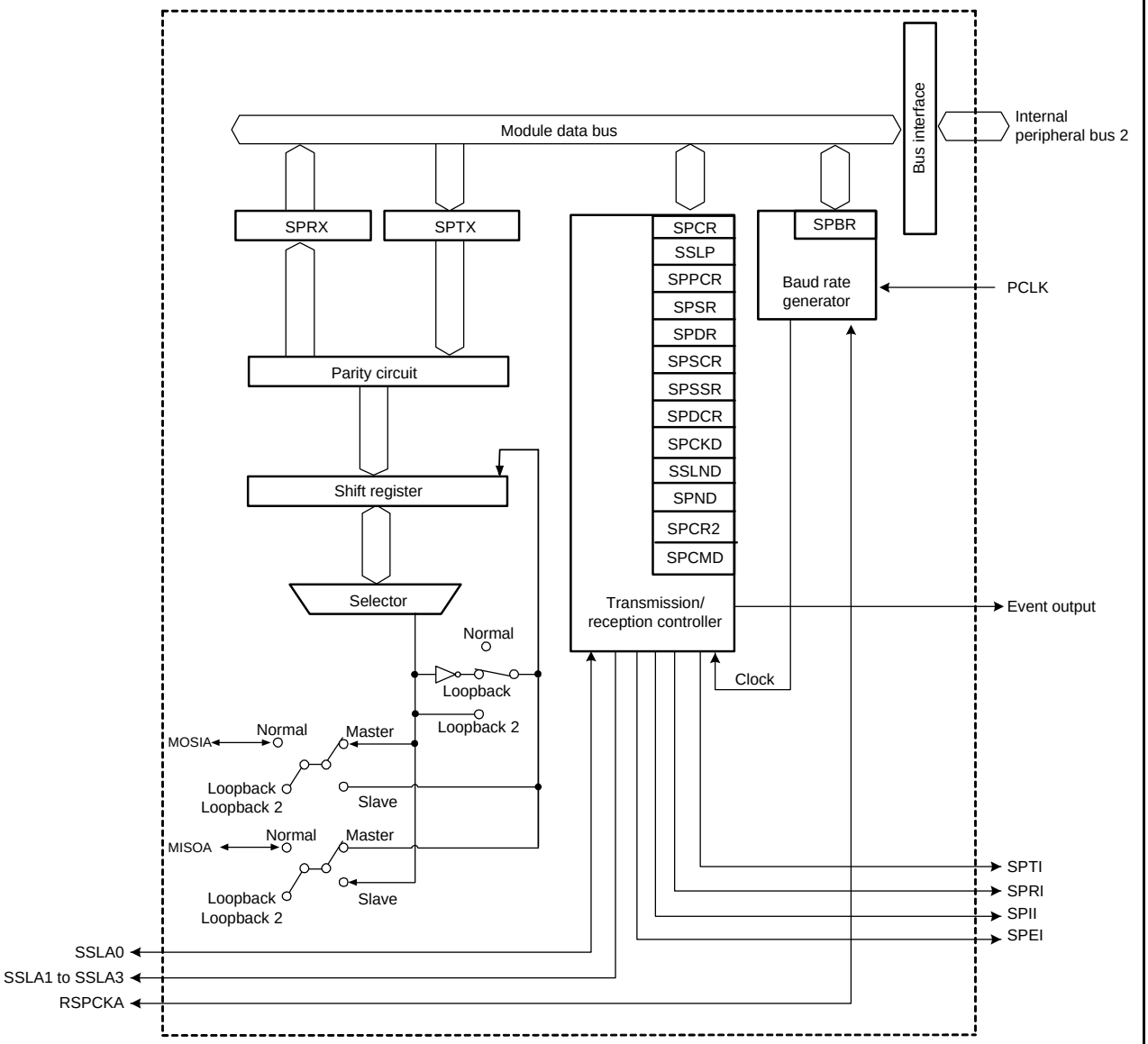
The RX220 Group includes one independent channels of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 30.1 lists the specifications of the RSPI, and Figure 30.1 shows a block diagram of the RSPI.
In this chapter, a lower-case letter i in pin and signal names indicates a value from 0 to 3, and a lower-case letter m in RSPI command register m (SPCMDm) indicates a value from 0 to 7.

Table 30.1 Specifications of RSPI (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (four-wire method) or clock synchronous operation (three-wire method). • Transmit-only operation is available. • Capable of serial communications in master/slave mode • Switching of the polarity of the serial transfer clock • Switching of the phase of the serial transfer clock
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 maximum divisor is 4096). • In slave mode, the externally input clock is used as the serial clock (the maximum frequency 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
Error detection	• Mode fault error detection • Overrun error detection • Parity error detection
SSL control function	• Four SSL signals (SSLA0 to SSLA3) for each channel • In single-master mode, SSLA0 to SSLA3 signals are output. • In multi-master mode: SSLA0 signal for input, and SSLA1 to SSLA3 signals for either output or unused. • In slave mode: SSLA0 signal for input, and SSLA1 to SSLA3 signals 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
Interrupt sources	• Maskable interrupt sources RSPI receive interrupt (receive buffer full) RSPI transmit interrupt (transmit buffer empty) RSPI error interrupt (mode fault, overrun, parity error) RSPI idle interrupt (RSPI idle)

Table 30.1 Specifications of RSPI (2/2)

Item	Description
Event linking	The following five types of events can be output to the event link controller. • Reception-buffer full event output • Transmission-buffer empty event output • Mode fault, overrun, or parity error event output • RSPI idle event output • Transmission-completed event output
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the RSPI • Loopback mode
Power consumption reducing function	Module stop state can be set.



SPCR: RSPI control register
SPCR2: RSPI control register 2
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPDR: RSPI data register
SPSCR: RSPI sequence control register
SPSSR: RSPI sequence status register
SPDCR: RSPI data control register
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negation delay register
SPND: RSPI next-access delay register
SPCMD: RSPI command registers 0 to 7 (eight registers)
SPBR: RSPI bit rate register
SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer
SPTI: RSPI transmit interrupt
SPRI: RSPI receive interrupt
SPII: SPI idle interrupt
SPEI: RSPI error interrupt

Figure 30.1 Block Diagram of RSPI

Table 30.2 lists the input and output pins used in the RSPI.
The RSPI automatically switches the input/output direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin (see section 30.3.2, Controlling RSPI Pins).

Table 30.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O pin
	MOSIA	I/O	Master transmit data I/O pin
	MISOA	I/O	Slave transmit data I/O pin
	SSLA0	I/O	Slave selection I/O pin
	SSLA1	Output	Slave selection output pin
	SSLA2	Output	Slave selection output pin
	SSLA3	Output	Slave selection output pin

30.2 Register Descriptions

30.2.1 RSPI Control Register (SPCR)

Address(es): 0008 8380h

	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 (four-wire method) 1: Clock synchronous operation (three-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	RSPI Transmit Interrupt Enable	0: Disables the generation of RSPI transmit interrupt requests 1: Enables the generation of RSPI transmit 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 Interrupt Enable	0: Disables the generation of RSPI receive interrupt requests 1: Enables the generation of RSPI receive interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, and SPCR.TXMD bits are changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (four-wire method) or clock synchronous operation (three-wire method).
The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The three pins RSPCKA, MOSIA, and MISOA 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 is not guaranteed 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 (see section 30.3.6, 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 (see section 30.3.8, Error Detection). In addition, the RSPI determines the input/output direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (see section 30.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

SPEIE Bit (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 (see section 30.3.8, Error Detection).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF bit is 1, the SPE bit cannot be set to 1. For details, refer to section 30.3.8, 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 30.3.9, Initializing RSPI. Furthermore, an RSPI transmission interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

30.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): 0008 8381h

	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 are not guaranteed.

30.2.3 RSPI Pin Control Register (SPPCR)

Address(es): 0008 8382h

	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 (reversed transmit data = receive data)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low. 1: The level output on the MOSIA pin during MOSI idling corresponds to high.	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations are not guaranteed.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (reverses) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

30.2.4 RSPI Status Register (SPSR)

Address(es): 0008 8383h

	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.

SPSR indicates the operating status of the RSPI. Writing to SPSR can only be performed under certain conditions.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- When a 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 writes the value 0 to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Neither condition 1 nor condition 2 is satisfied in master mode under the clearing conditions below.

Slave mode

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing conditions]

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)

The flag is cleared to 0 when the above first clearing condition is satisfied or all of the second to fourth clearing conditions are satisfied.

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 SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSL signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then writes the value 0 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 writes the value 0 to the PERF flag

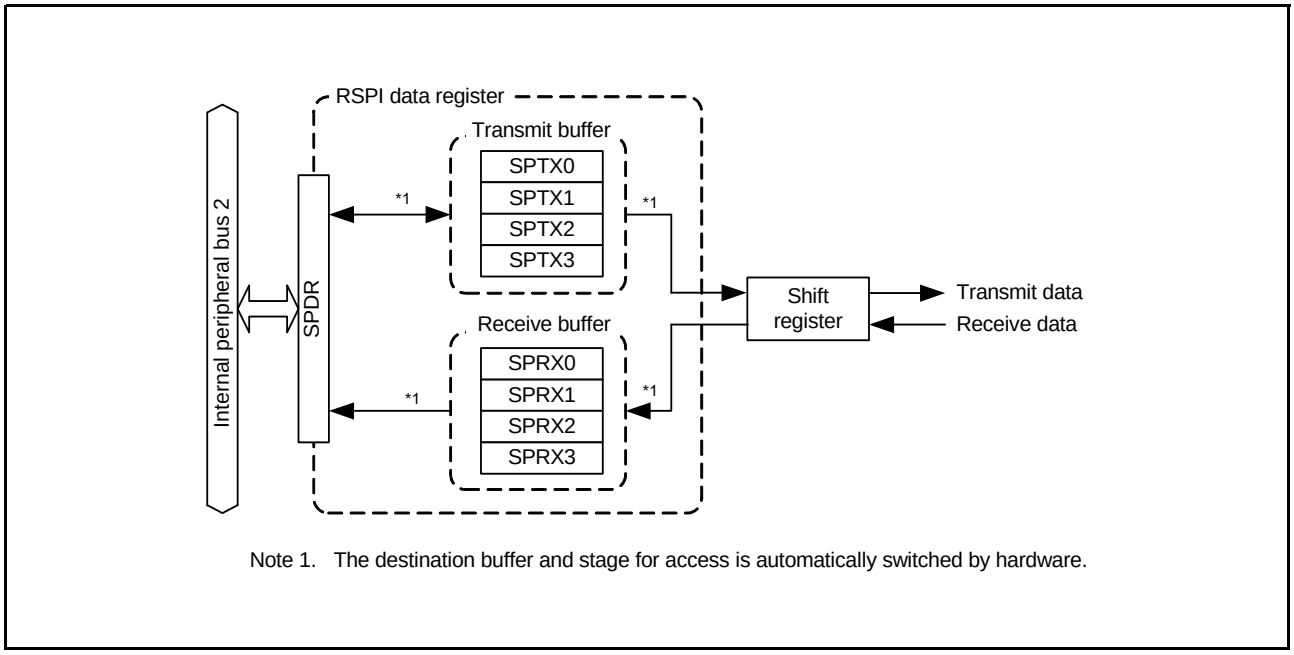
30.2.5 RSPI Data Register (SPDR)

Address(es): 0008 8384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
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
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
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. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 30.2 shows the Configuration of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 30.2 Configuration of SPDR

The transmit and receive buffers have four stages each. 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) 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 nine bits, the SPTXn[31:9] bits are stored in SPRXn[31:9] (and received data are stored in the SPRXn[8:0] 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 four-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 the SPDR register.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data is written to SPDR.
Figure 30.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

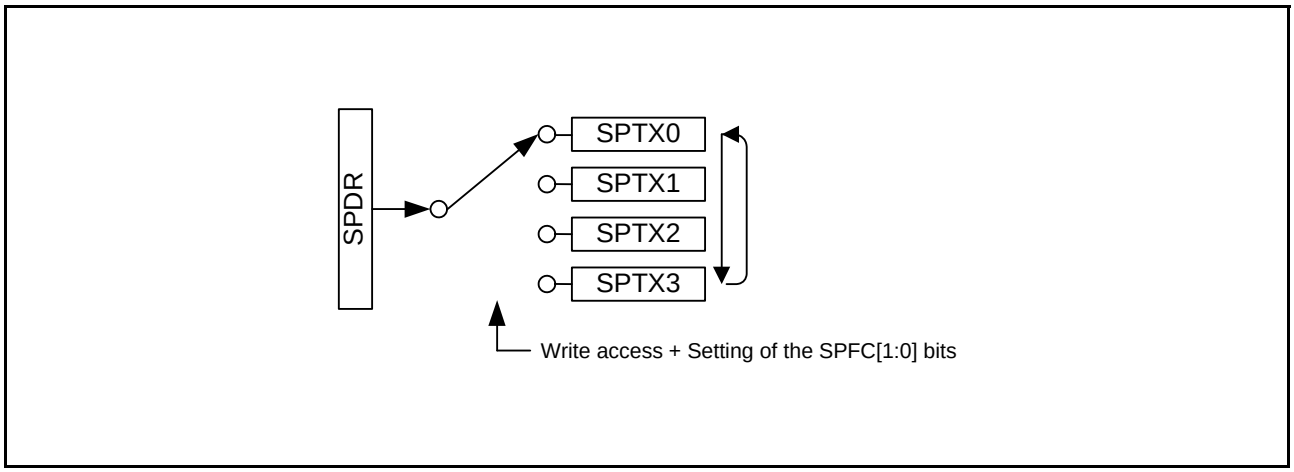


Figure 30.3 Configuration of SPDR (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the frame-number setting 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 RSPI transmission interrupt, write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Writing to the transmit buffer (SPTXn) before the next RSPI transmission interrupt breaks off write access to the SPDR.

(b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn; n = 0 to 3) or a transmit buffer (SPTXn; n = 0 to 3). The setting of the RSPI receive/transmit data selection bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The structure of the SPDR when it is read includes two independent pointers (receive buffer read pointer and transmit buffer read pointer). Reading SPDR causes automatic updating of the pointer so that it indicates the next stage of the buffer.

Figure 30.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

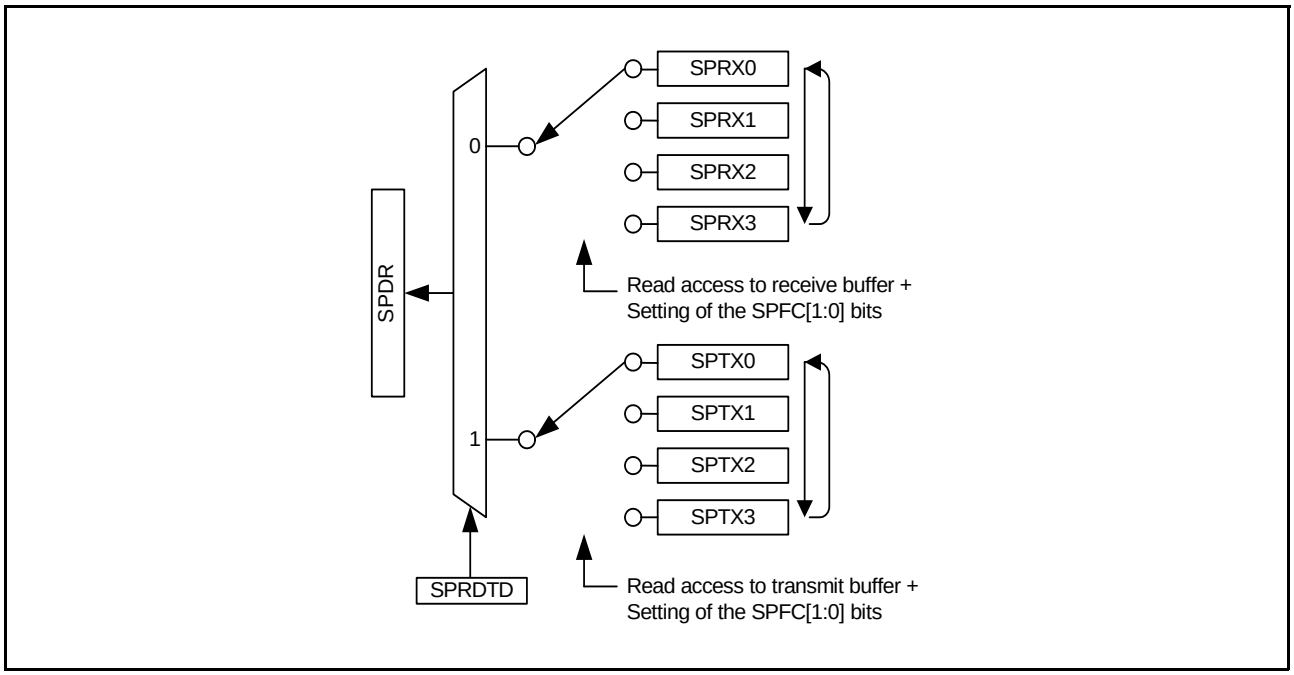


Figure 30.4 Configuration of SPDR (Reading)

The sequence for switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. When reading is from the transmit buffer, the value before that most recently written is read. Furthermore, only the read pointer for the buffer that is currently selected for reading by the setting of the RSPI receive/transmit selection bit in the RSPI data control register (SPDCR.SPRDTD) is updated, and the state of the read pointer for the other buffer is preserved.

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

If the transmit buffer is read in the interval after writing of the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) and generation of the RSPI transmission interrupt, and before the next RSPI transmission interrupt, all bits of the value read are 0.

The SPTXn buffer-reading pointer is cleared when the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) are written after generation of the RSPI transmission interrupt.

30.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): 0008 8388h

	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→… SPCMD0 to SPCMD7 to be referenced and the order in which they are referenced are 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 always 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 always referred to.

30.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): 0008 8389h

	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, see section 30.3.10.1, 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 bits 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, see section 30.3.8, Error Detection. For the RSPI's sequence control, see section 30.3.10.1, Master Mode Operation.

30.2.8 RSPI Bit Rate Register (SPBR)

Address(es): 0008 838Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
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 cannot be guaranteed.

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).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1)\, 2^{N}}$$

Table 30.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 30.3 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

30.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh

	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 Selection	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 reception interrupts, and start of transmission or generation of RSPI transmission interrupts. Table 30.4 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 are not guaranteed.

Table 30.4 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 Selection)

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 RSPI transmission interrupt.
For details, see section 30.2.5, 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 specification bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operation is not guaranteed.

30.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): 0008 838Ch

	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 b1 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 SSLAi 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 cannot be guaranteed.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

30.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): 0008 838Dh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

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.

30.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): 0008 838Eh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

30.2.13 RSPI Control Register 2 (SPCR2)

Address(es): 0008 838Fh

	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-Testing	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 bit or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

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 cleared to 0.

PTE Bit (Parity Self-Testing)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

30.2.14 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

	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 cannot be guaranteed.

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 (see section 30.2.8, 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 permits the execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
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 SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

30.3 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.

30.3.1 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 30.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 30.5 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
RSPCKA signal	Input	Output	Output/Hi-Z	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z	Input	Output
MISOA signal	Output/Hi-Z	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLA1 to SSLA3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity modification 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	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transfer data length	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	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	Supported	Supported	Supported	Supported
Receive buffer full detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Overrun error detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Parity error detection	Supported*2,*3	Supported*2,*3	Supported*2,*3	Supported*2,*3	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.

30.3.2 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 30.6 lists the relationship between pin states and bit settings. The I/O port settings should follow this relationship.

Table 30.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi for I/O ports = 0	ODRn.Bi for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKA	CMOS output	Open-drain output
	SSLA0 to SSLA3	CMOS output	Open-drain output
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLA0	Input	Input
	SSLA1 to SSLA3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOA	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKA	Input	Input
	SSLA0	Input	Input
	SSLA1 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKA	CMOS output	Open-drain output
	SSLA1 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKA	Input	Input
	SSLA1 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA	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 SSLA0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is cleared (= 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 30.7.

Table 30.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Always low
1	1	Always High

30.3.3 RSPI System Configuration Examples

30.3.3.1 Single Master/Single Slave (with This LSI Acting as Master)

Figure 30.5 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this LSI (master) are not used. The SSL input of the RSPI slave is fixed to the low level, and the RSPI slave is always maintained in a select state.*1
This LSI (master) always drives the RSPCKA and MOSIA. The RSPI slave always drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this LSI should be connected to the SSL input of the slave device.

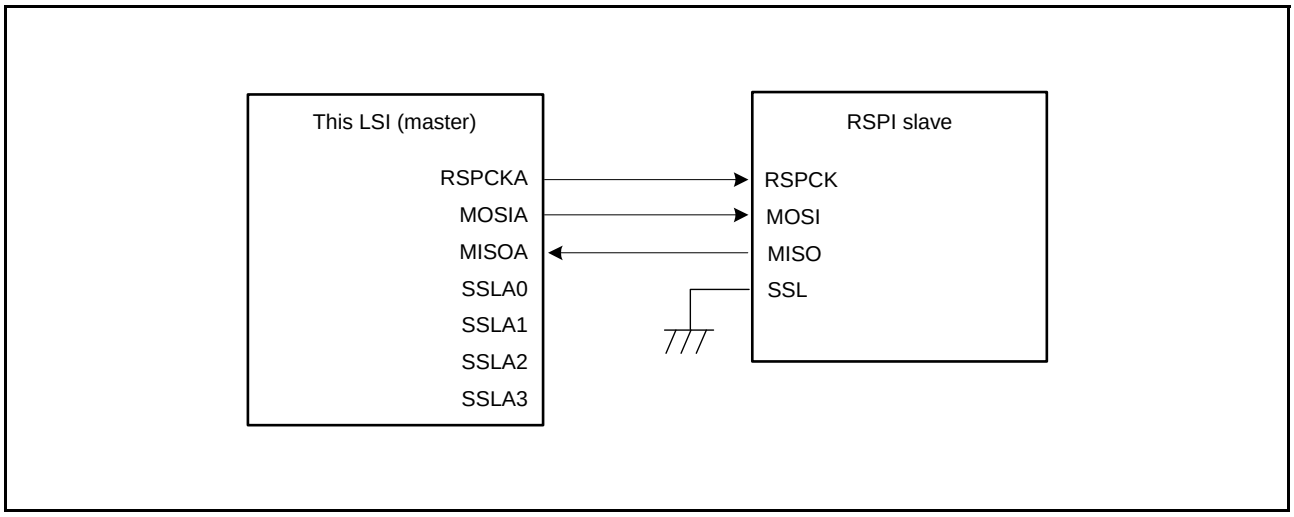


Figure 30.5 Single-Master/Single-Slave Configuration Example (This LSI = Master)

30.3.3.2 Single Master/Single Slave (with This LSI Acting as Slave)

Figure 30.6 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSLA0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this LSI (slave) is fixed to the low level, this LSI (slave) is always maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 30.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

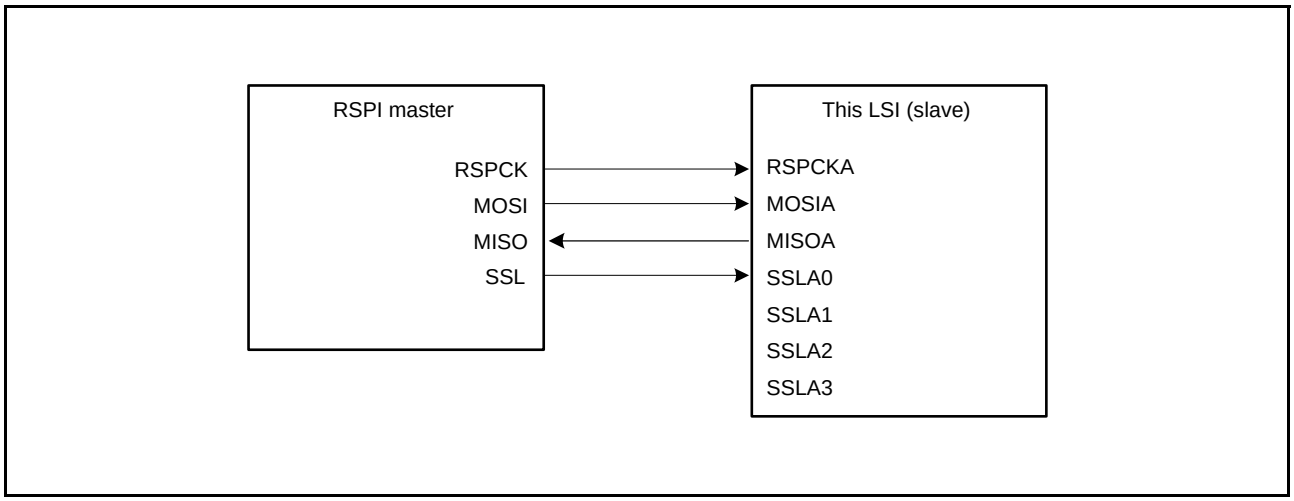


Figure 30.6 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0)

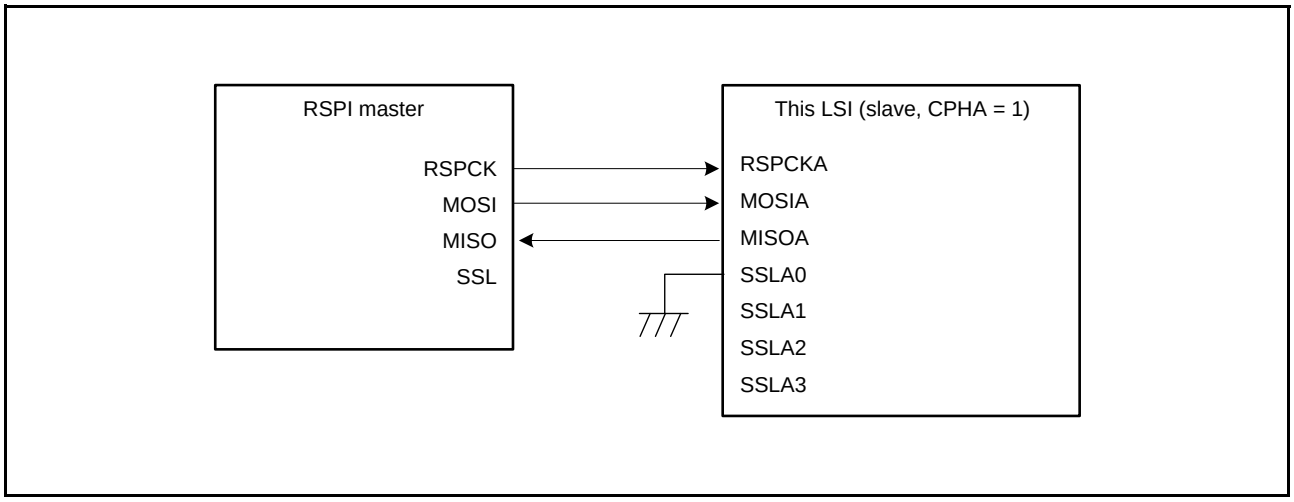


Figure 30.7 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

30.3.3.3 Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 30.8 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 30.8, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCKA and MOSIA outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISOA input of this LSI (master). SSLA0 to SSLA3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively.

This LSI (master) always drives RSPCK, MOSI, and SSLA0 to SSLA3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

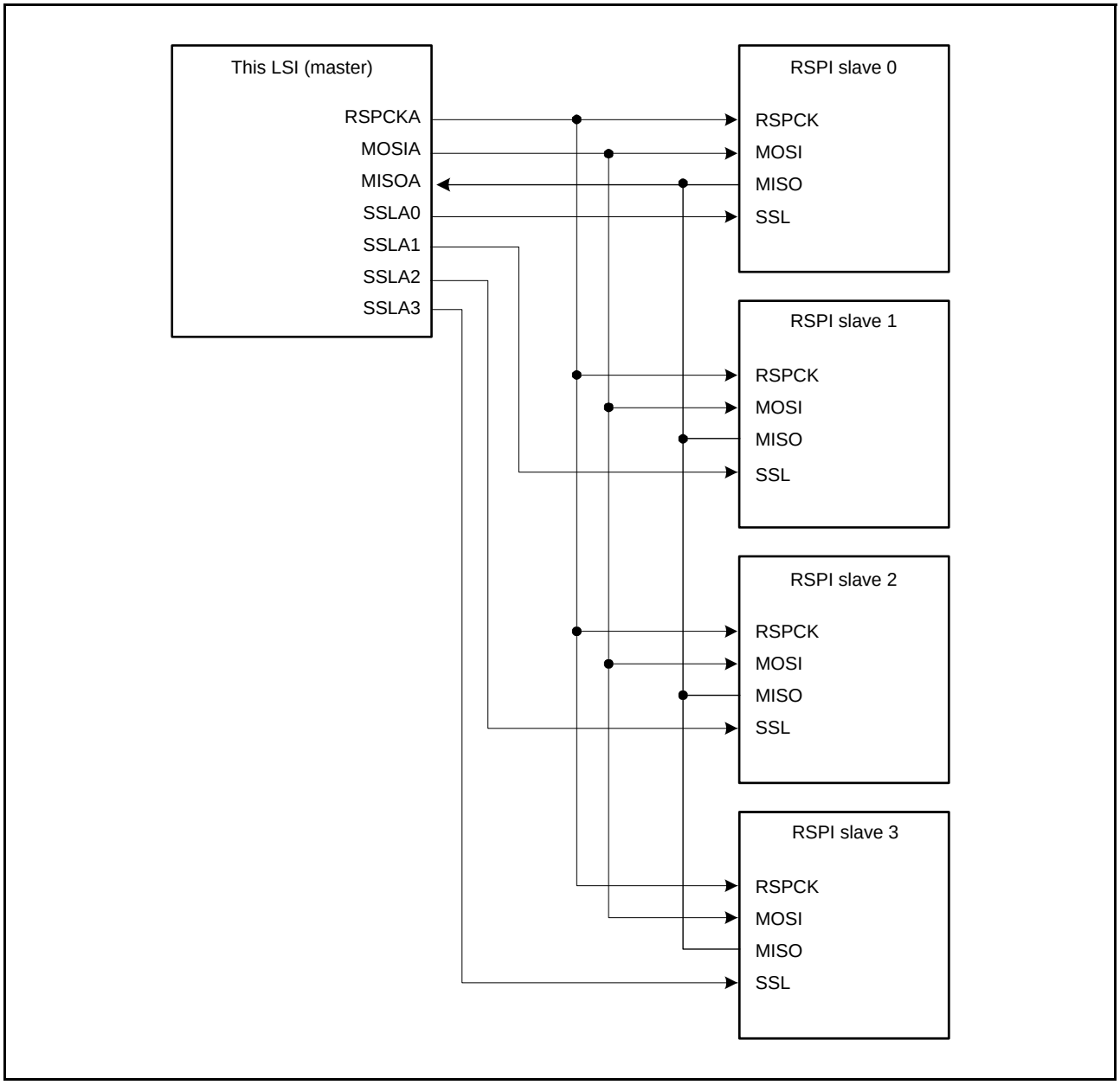


Figure 30.8 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

30.3.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 30.9 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of Figure 30.9, the RSPI system is comprised of an RSPI master and two LSIs (slave X and slave Y). The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCKA and MOSIA inputs of the LSIs (slave X and slave Y). The MISOA outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSLA0 inputs of the LSIs (slave X and slave Y), respectively.
The RSPI master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

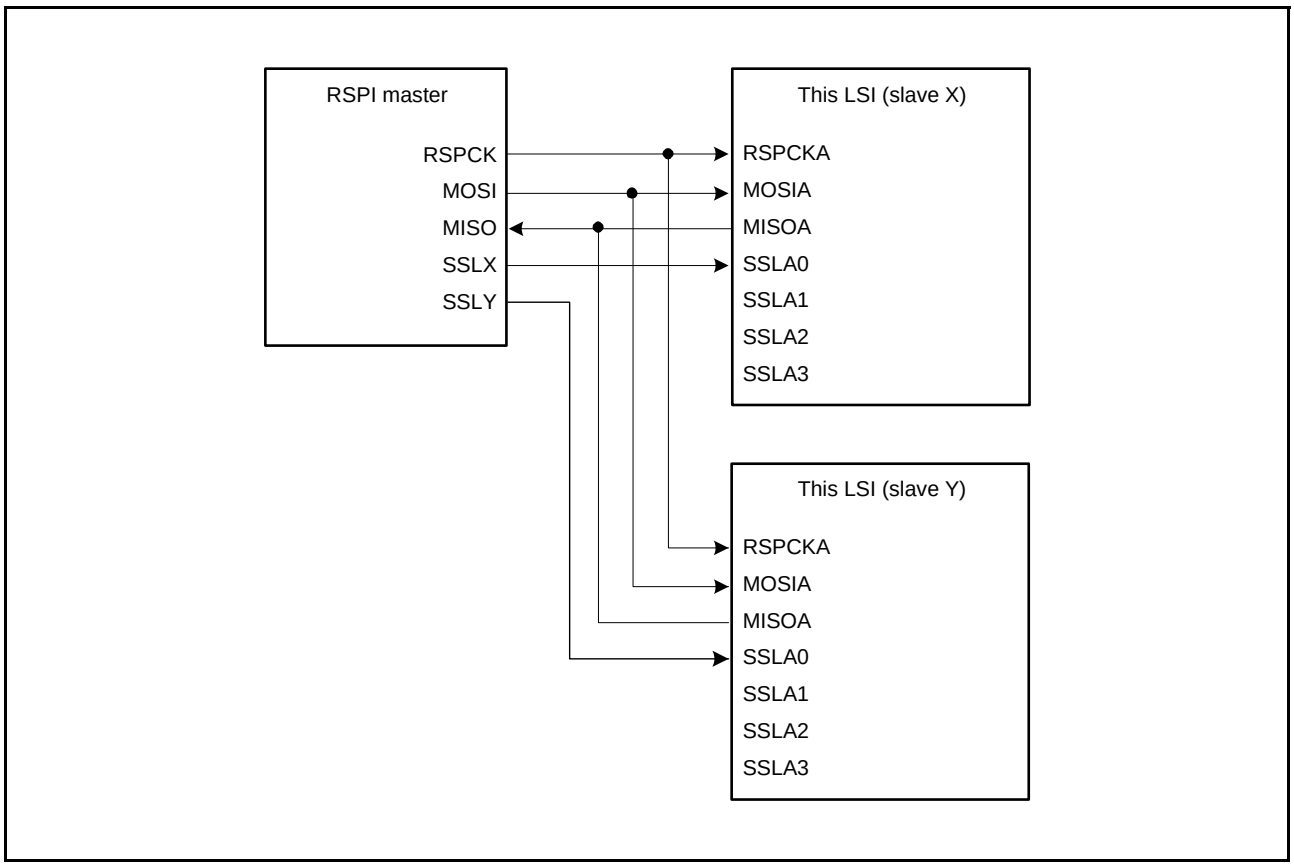


Figure 30.9 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

30.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 30.10 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 30.10, the RSPI system is comprised of two LSIs (master X and master Y) and two RSPI slaves (RSPI slave 1 and RSPI slave 2).

The RSPCKA and MOSIA outputs of the LSIs (master X and master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISOA inputs of the LSIs (master X and master Y). Any generic port Y output from this LSI (master X) is connected to the SSLA0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSLA0 input of this LSI (master X). The SSLA1 and SSLA2 outputs of the LSIs (master X and master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this LSI is not required.

This LSI drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this LSI detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

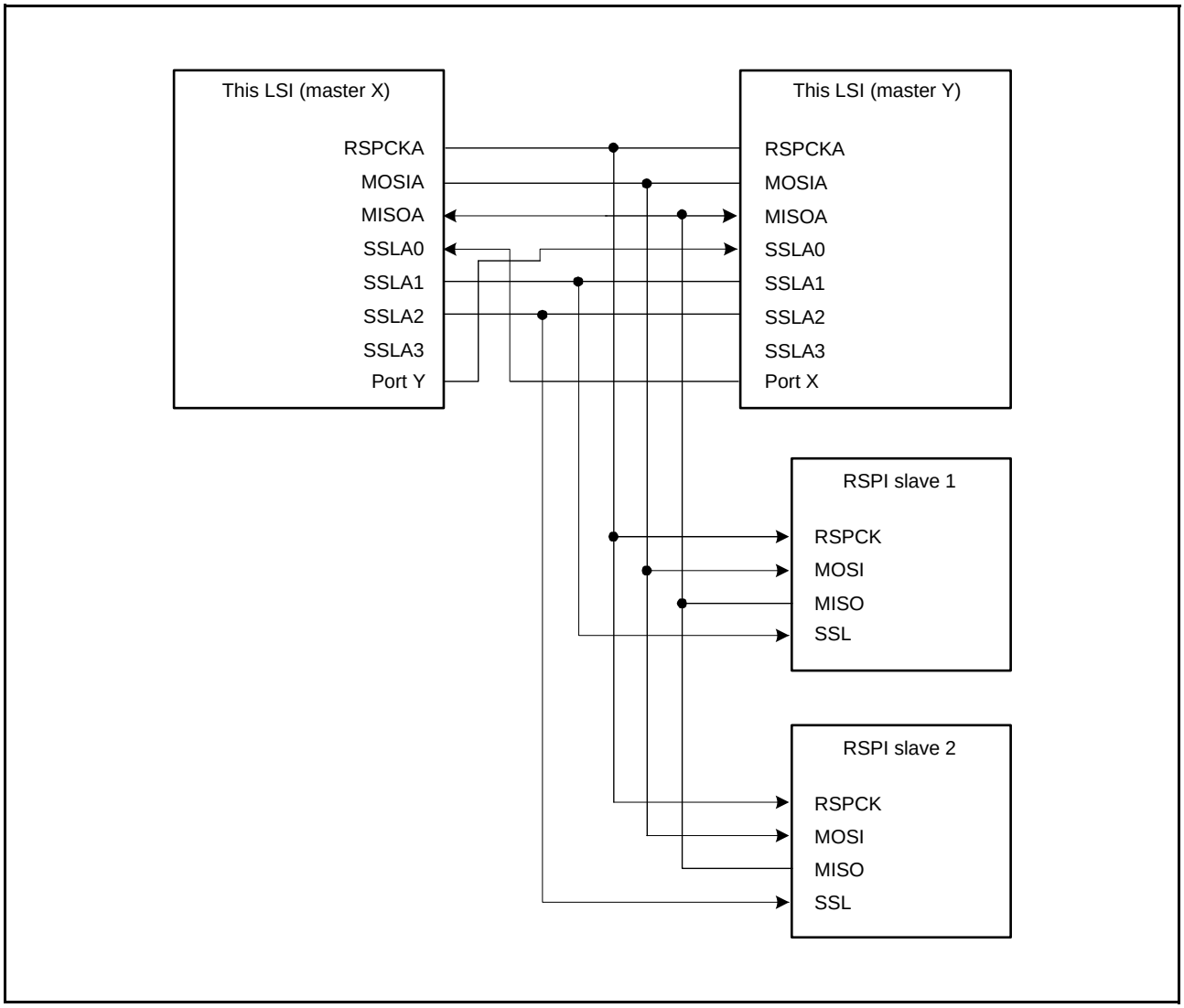


Figure 30.10 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

30.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master)

Figure 30.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this LSI (master) are not used.
This LSI (master) always drives the RSPCKA and MOSIA. The RSPI slave always drives the MISO.

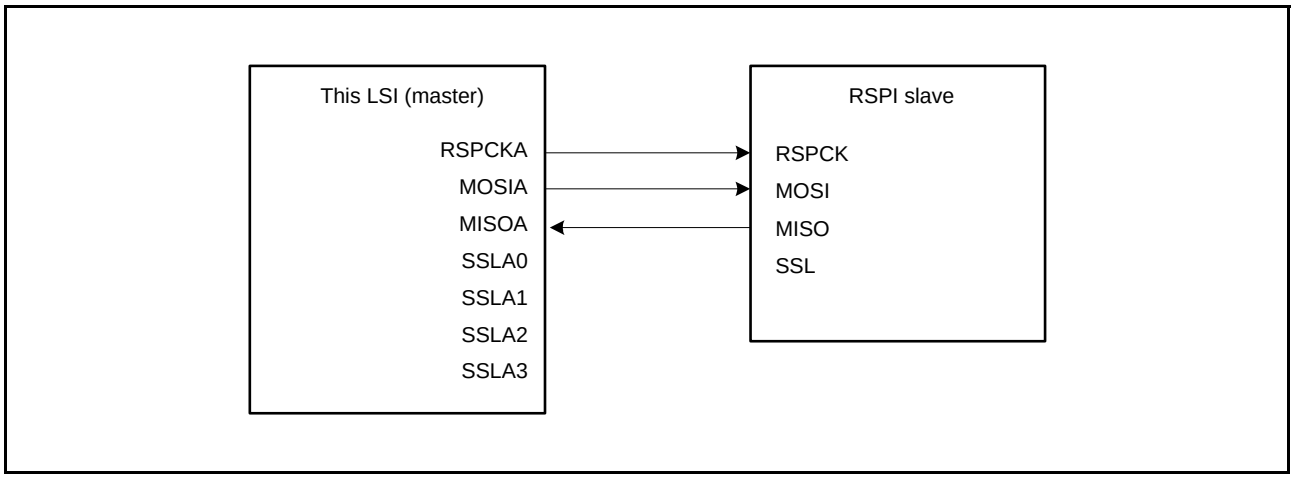


Figure 30.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Master)

30.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave)

Figure 30.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave (clock synchronous operation), this LSI (slave) always drives the MISOA and the RSPI master always drives the RSPCK and MOSI. In addition, SSLA0 to SSLA3 of this LSI (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this LSI (slave) can execute serial transfer.

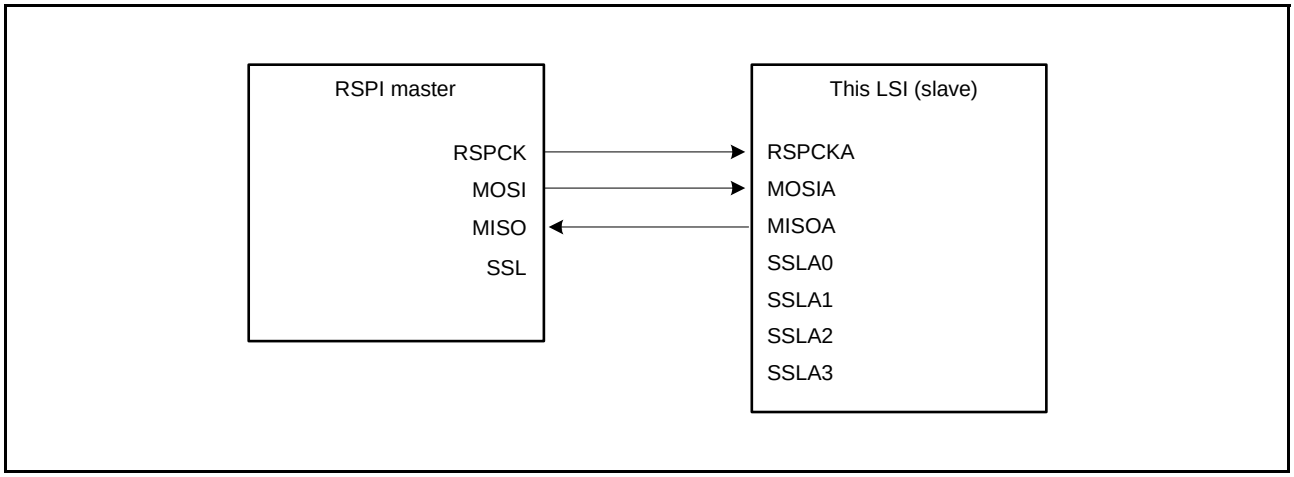


Figure 30.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Slave, CPHA = 1)

30.3.4 Data Format

The RSPI's data format depends on the settings in the RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit of RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit of 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 data length setting bits in the RSPI command register m (SPCMDm.SPB[3:0]).

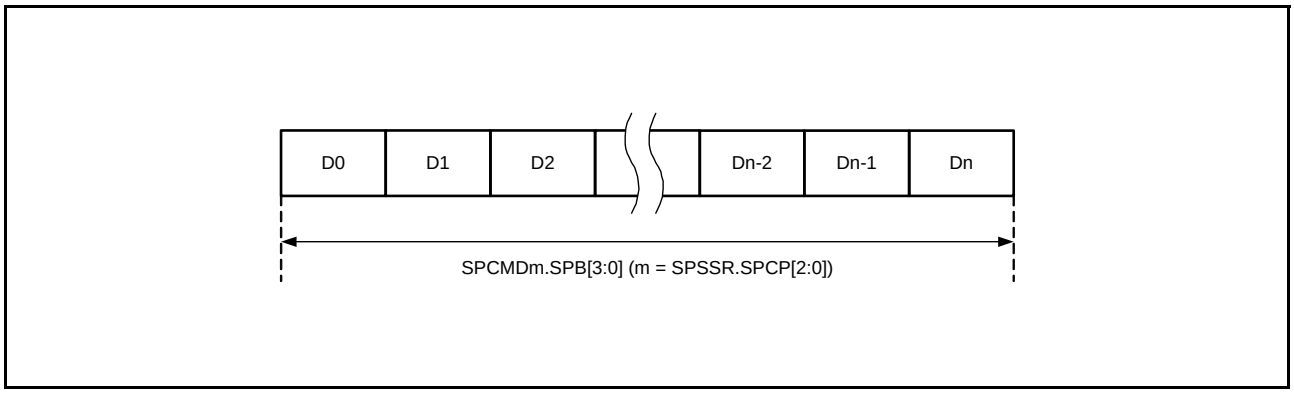


Figure 30.13 Outline of the Data Format (with Parity Disabled)

(b) With Parity Enabled

When parity is enabled, transmission or reception of data also proceeds with the length in bits selected in the SPCMDm.SPB[3:0] bits. In this case, however, the last bit is a parity bit.

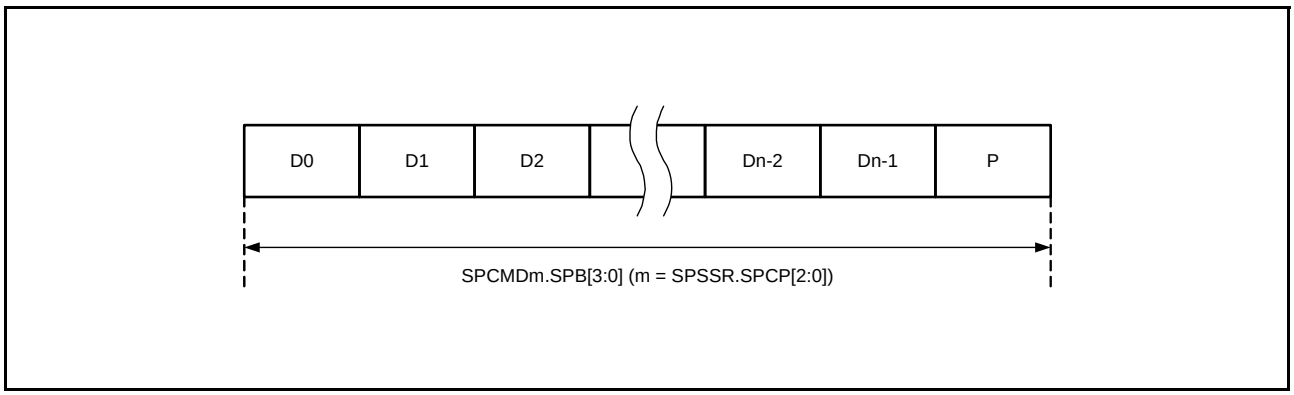


Figure 30.14 Outline of the Data Format (with Parity Enabled)

30.3.4.1 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 30.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB-first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

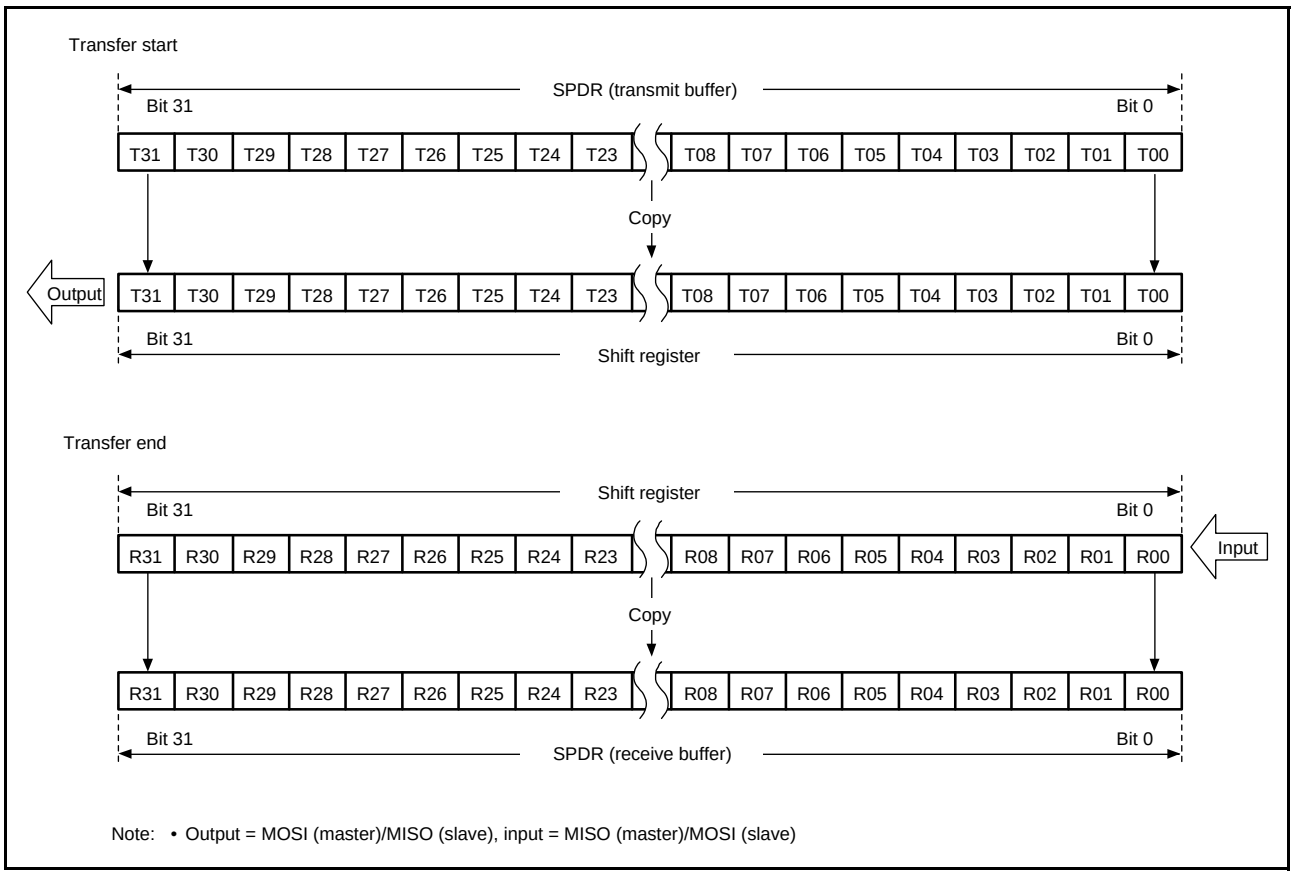
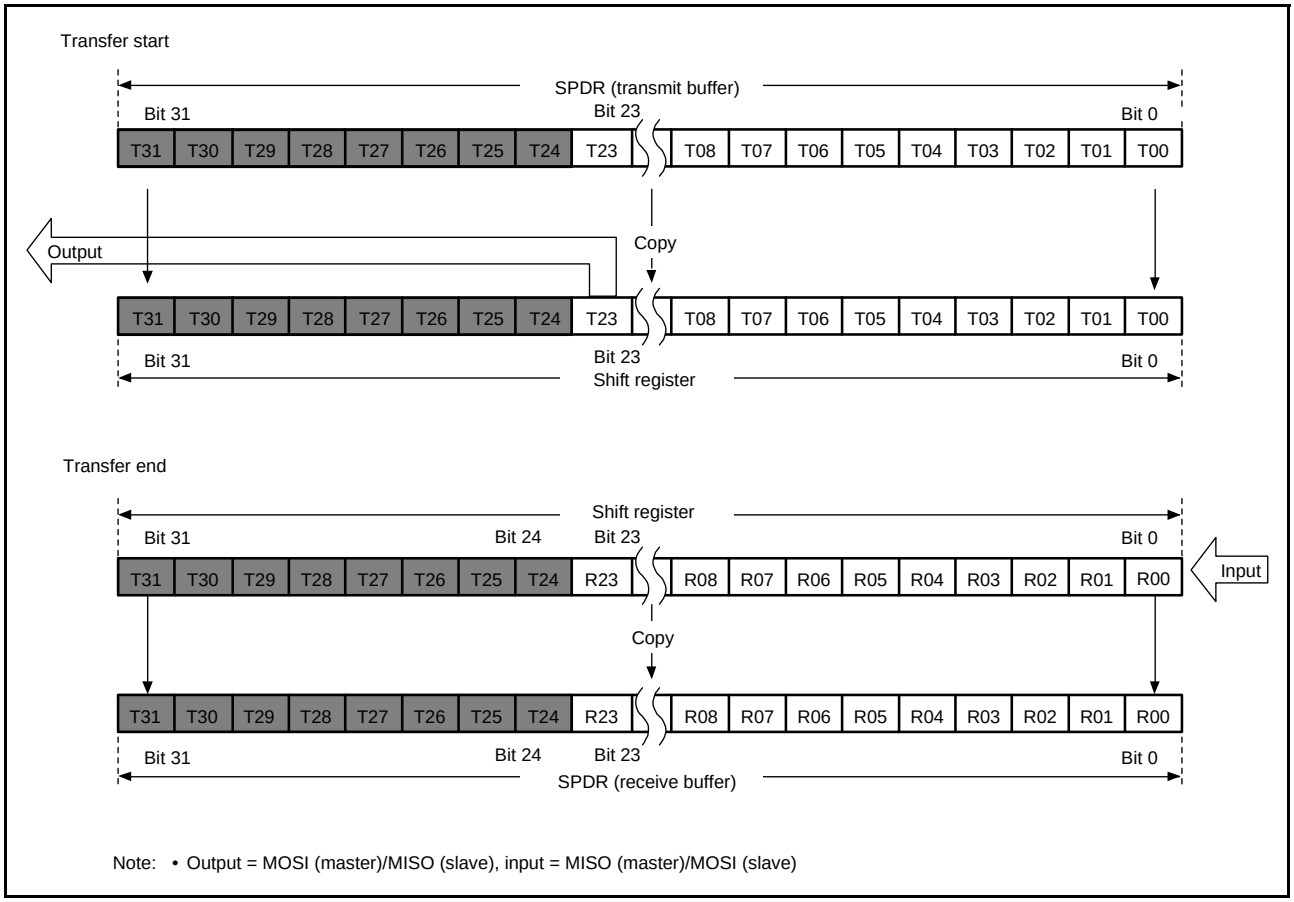


Figure 30.15 MSB-First Transfer (32-Bit Data, Parity Disabled)

(2) MSB-First Transfer (24-Bit Data)

Figure 30.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB-first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.



Note: • Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 30.16 MSB-First Transfer (24-Bit Data, Parity Disabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 30.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

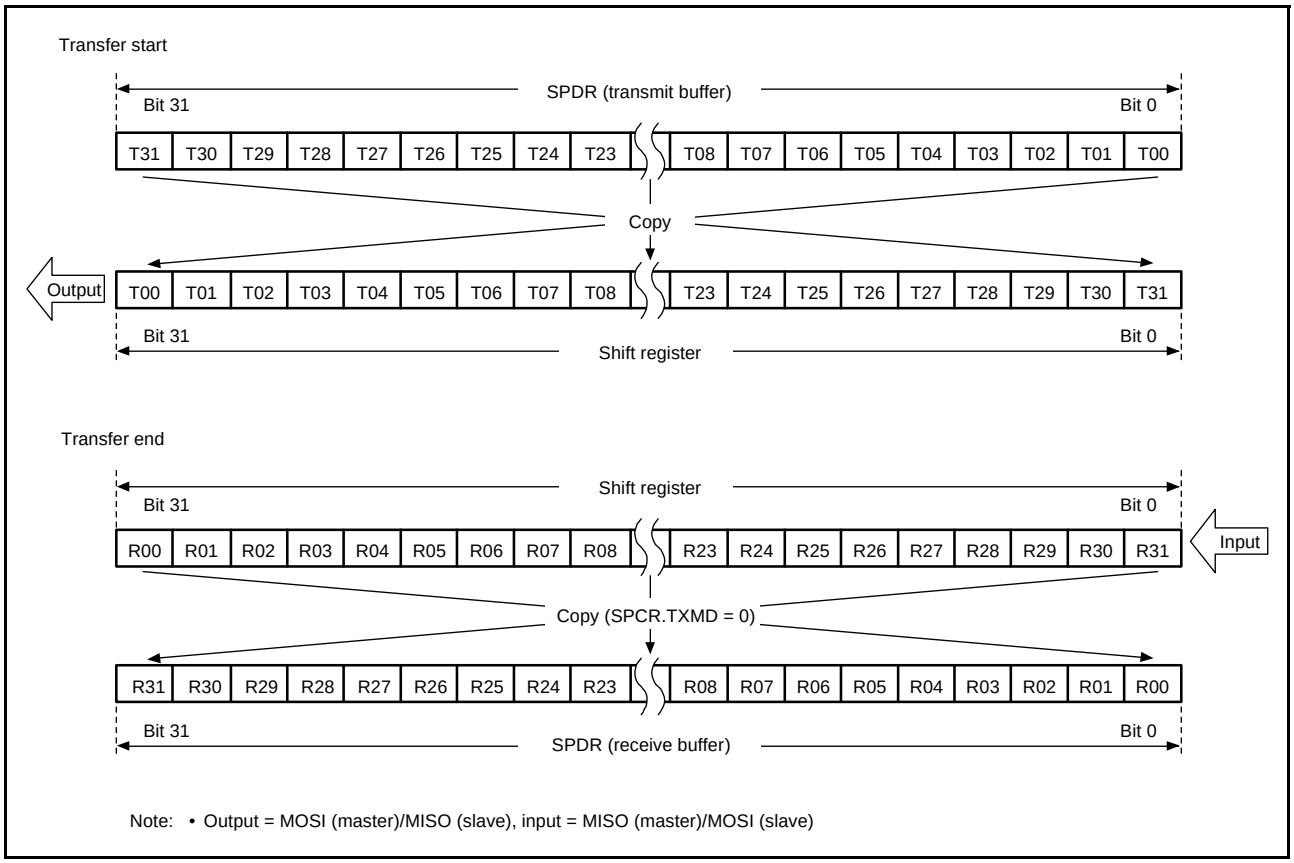


Figure 30.17 LSB-First Transfer (32-Bit Data, Parity Disabled)

(4) LSB-First Transfer (24-Bit Data)

Figure 30.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB-first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

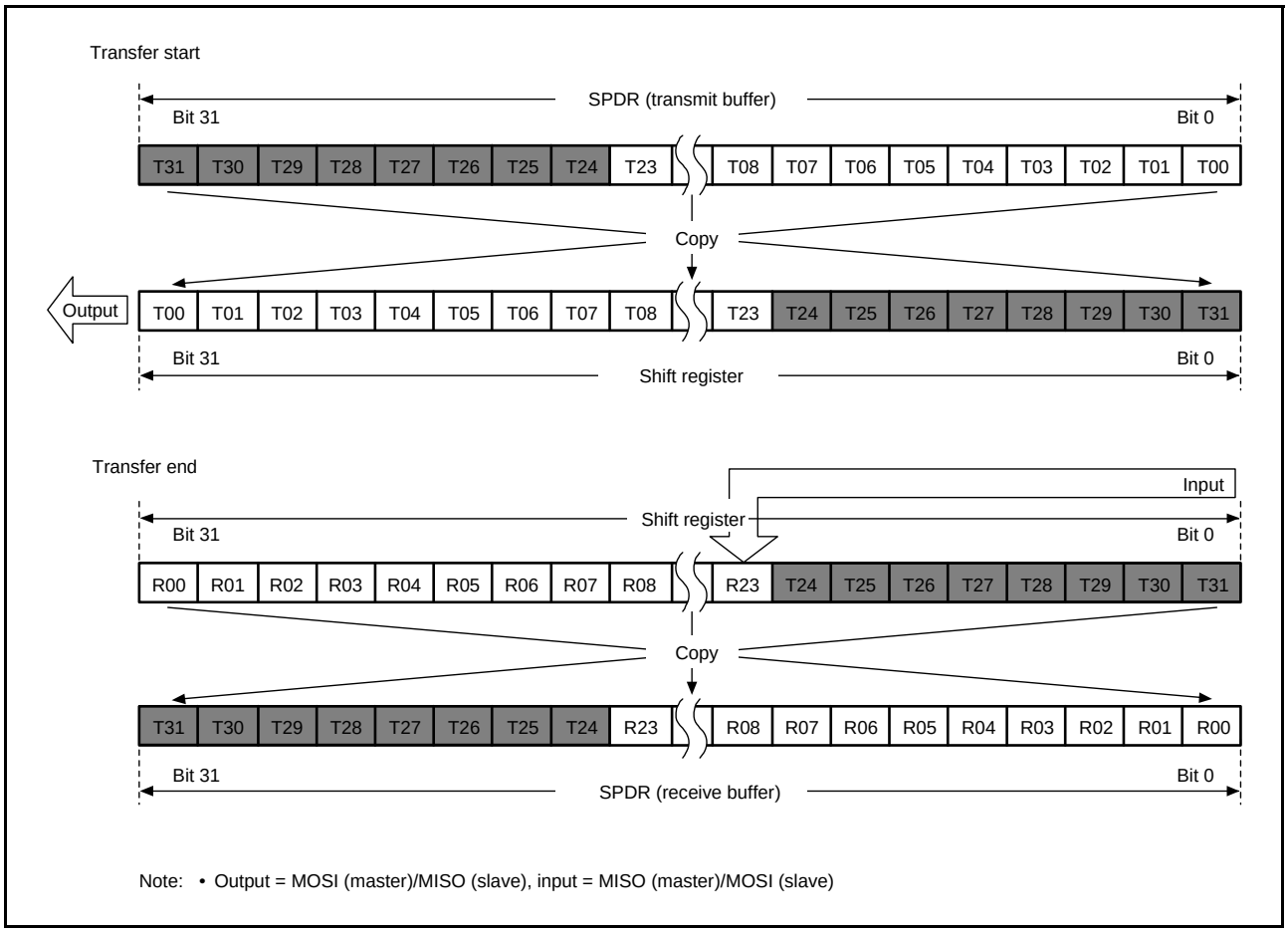


Figure 30.18 LSB-First Transfer (24-Bit Data, Parity Disabled)

30.3.4.2 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 30.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB-first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

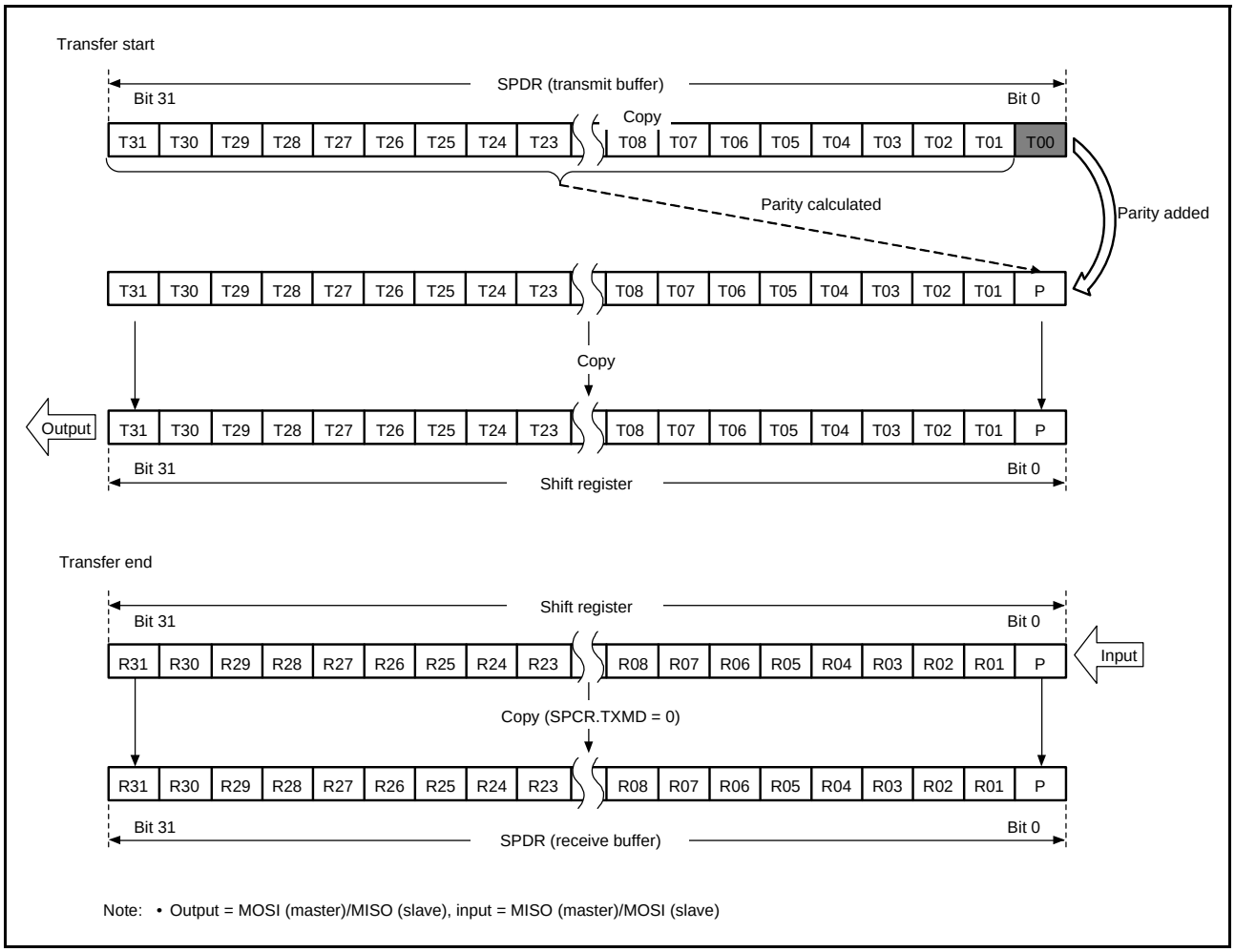


Figure 30.19 MSB-First Transfer (32-Bit Data, Parity Enabled)

(2) MSB-First Transfer (24-Bit Data)

Figure 30.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB-first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

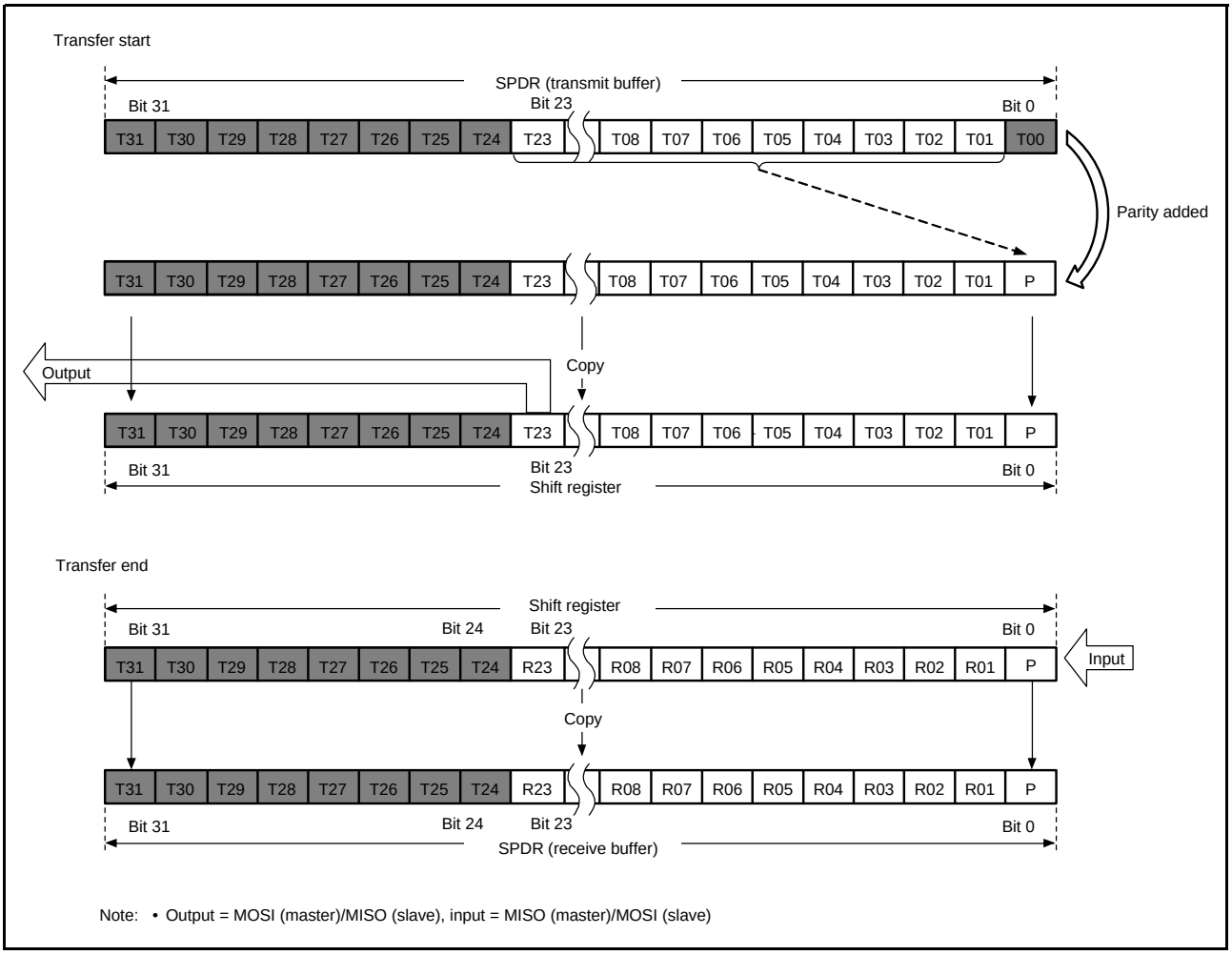


Figure 30.20 MSB-First Transfer (24-Bit Data, Parity Enabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 30.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the 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.

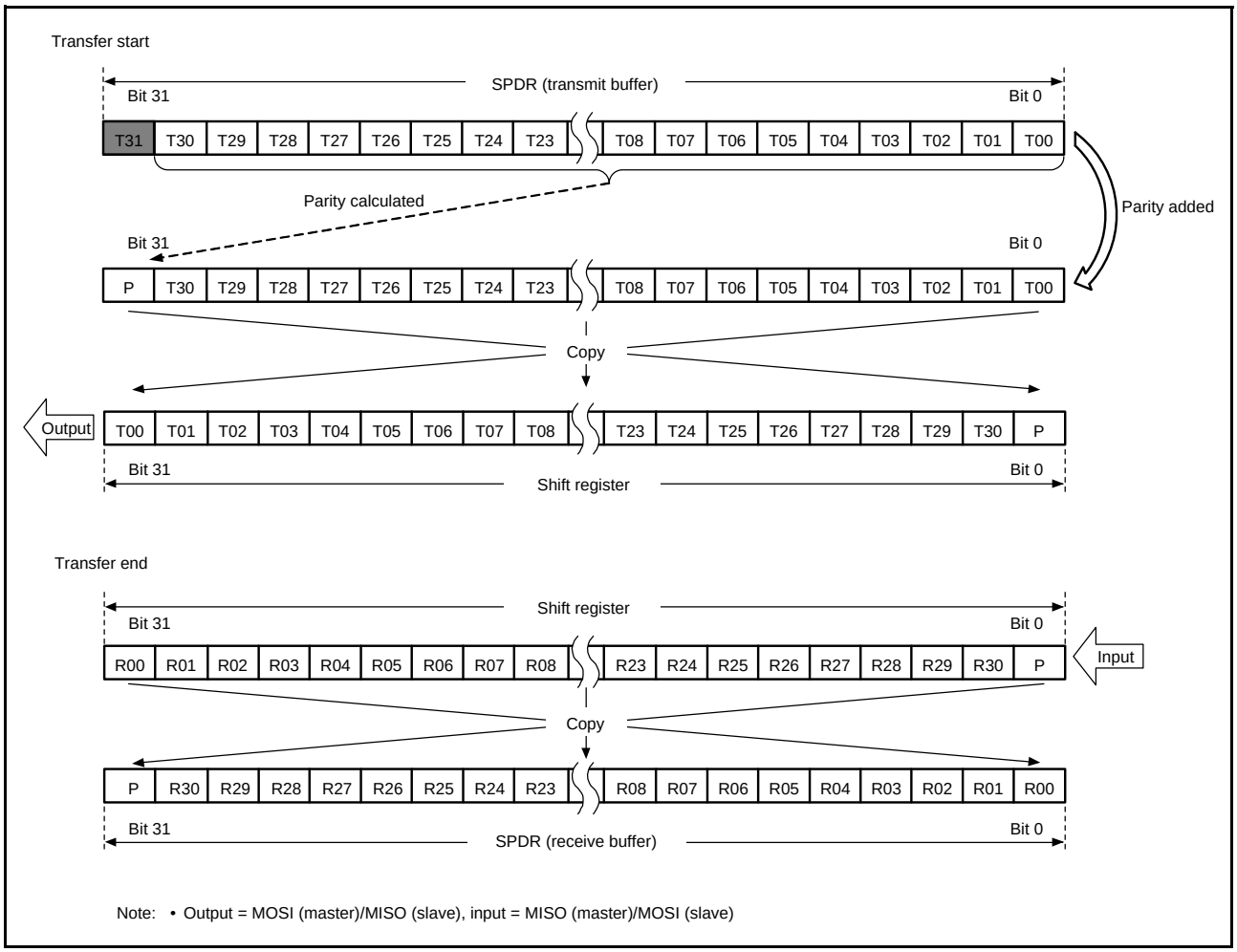


Figure 30.21 LSB-First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 30.22 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB-first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the 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 eight bits of the transmit buffer are stored in the higher-order eight 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 eight bits of the receive buffer.

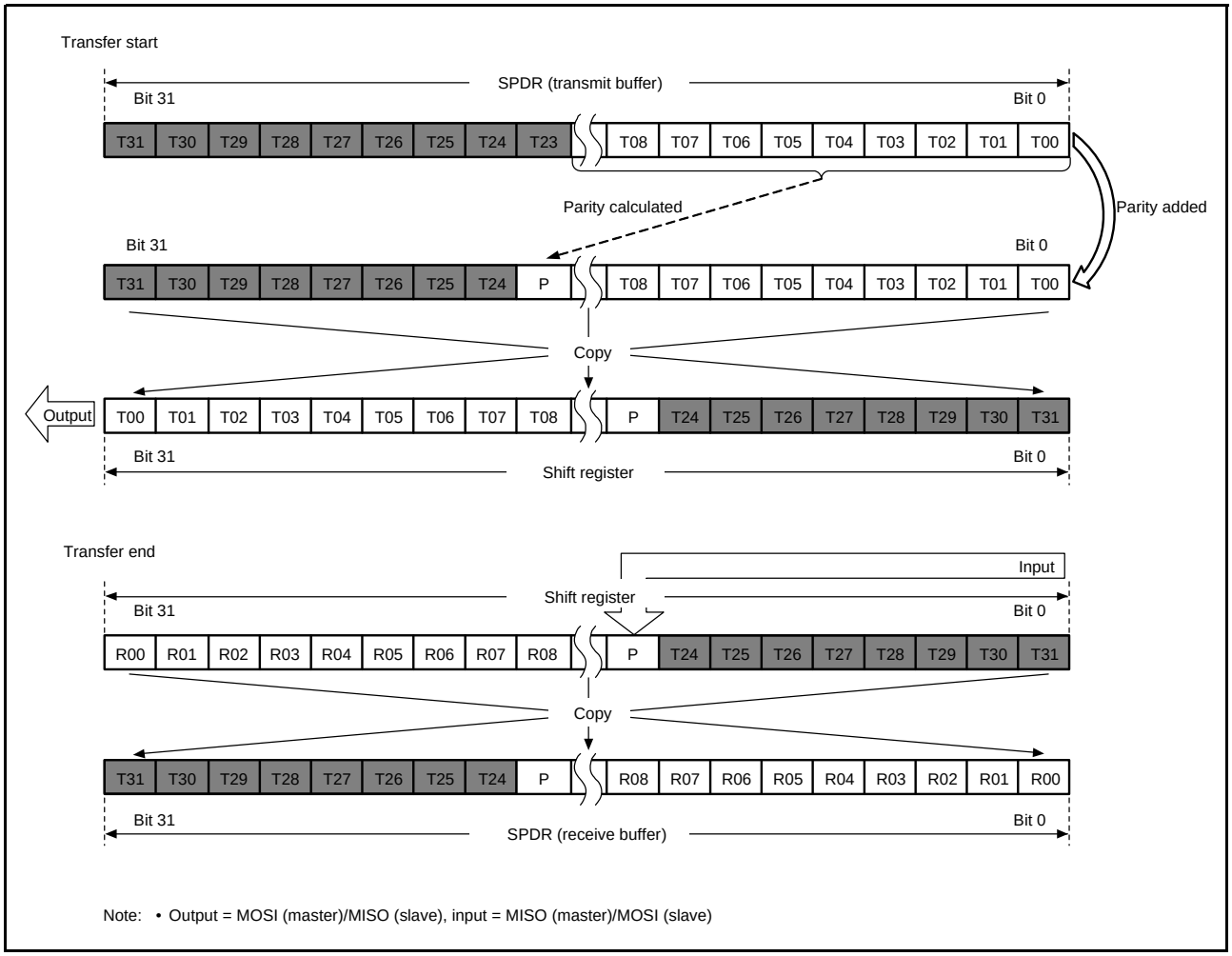


Figure 30.22 LSB-First Transfer (24-Bit Data, Parity Enabled)

30.3.5 Transfer Format

30.3.5.1 CPHA = 0

Figure 30.23 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) is not guaranteed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 30.23, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 30.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is always 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 SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 30.3.10.1, Master Mode Operation.

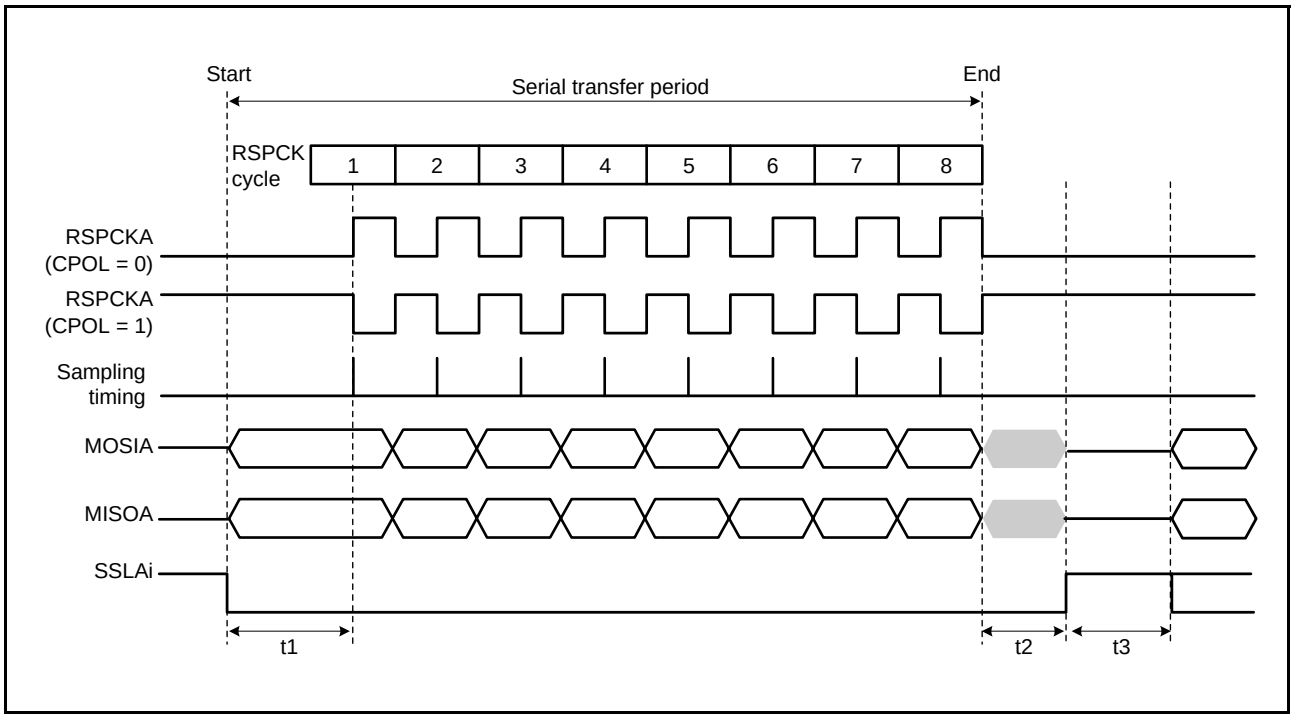


Figure 30.23 RSPI Transfer Format (CPHA = 0)

30.3.5.2 CPHA = 1

Figure 30.24 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 30.24, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI mode (master or slave). For details, see section 30.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 30.3.10.1, Master Mode Operation.

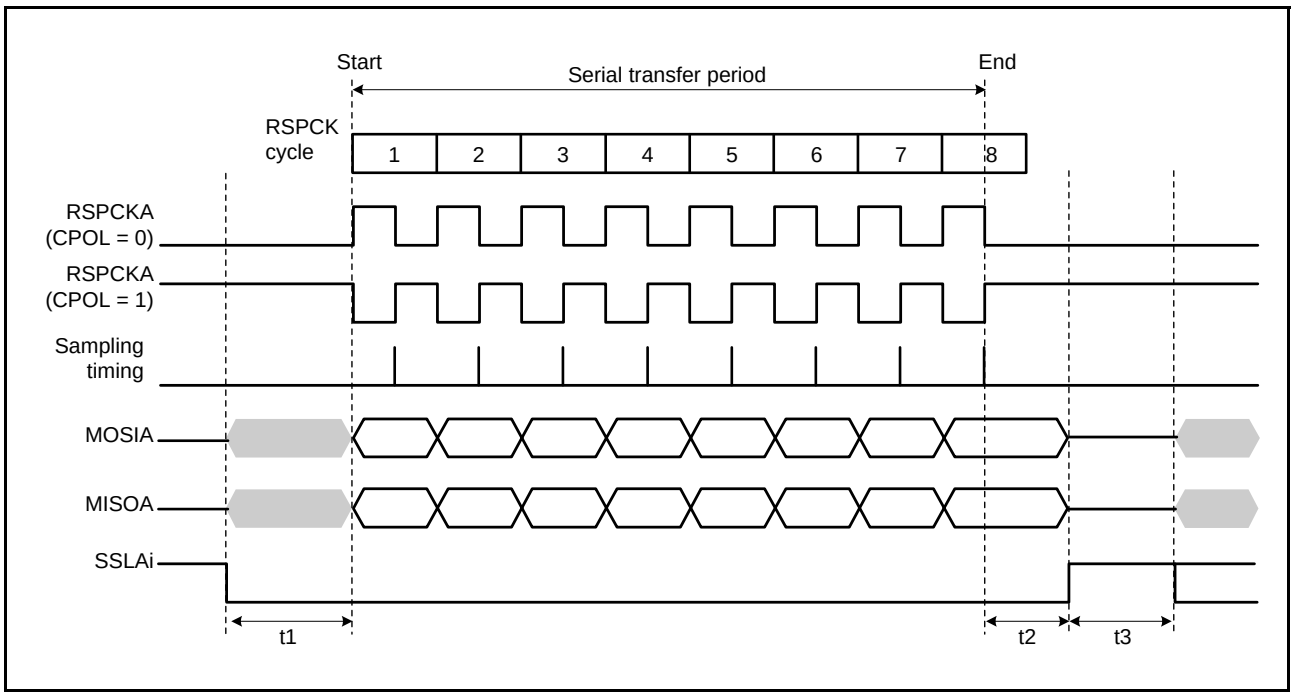


Figure 30.24 RSPI Transfer Format (CPHA = 1)

30.3.6 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 30.25 and Figure 30.26 indicates the condition of access to the SPDR register, where W denotes a write cycle.

30.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 30.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 30.25, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

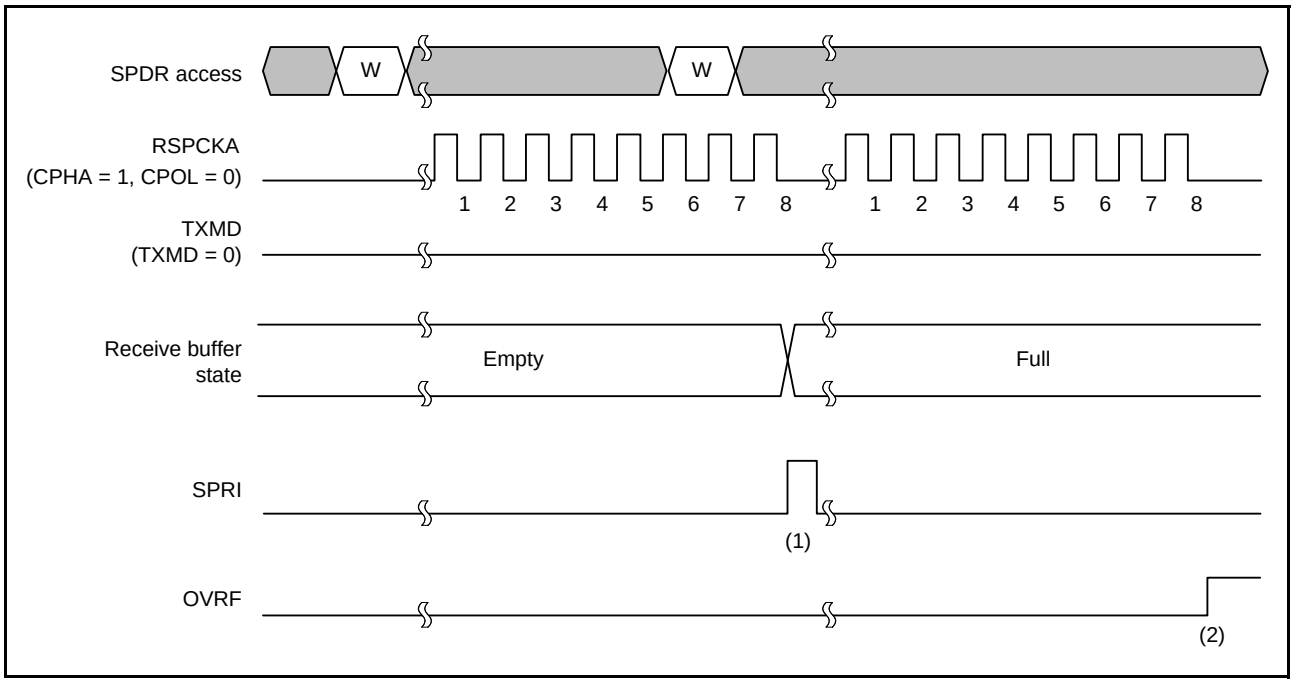


Figure 30.25 Operation Example of SPCR.TXMD = 0

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.

30.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 30.26 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 30.26, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

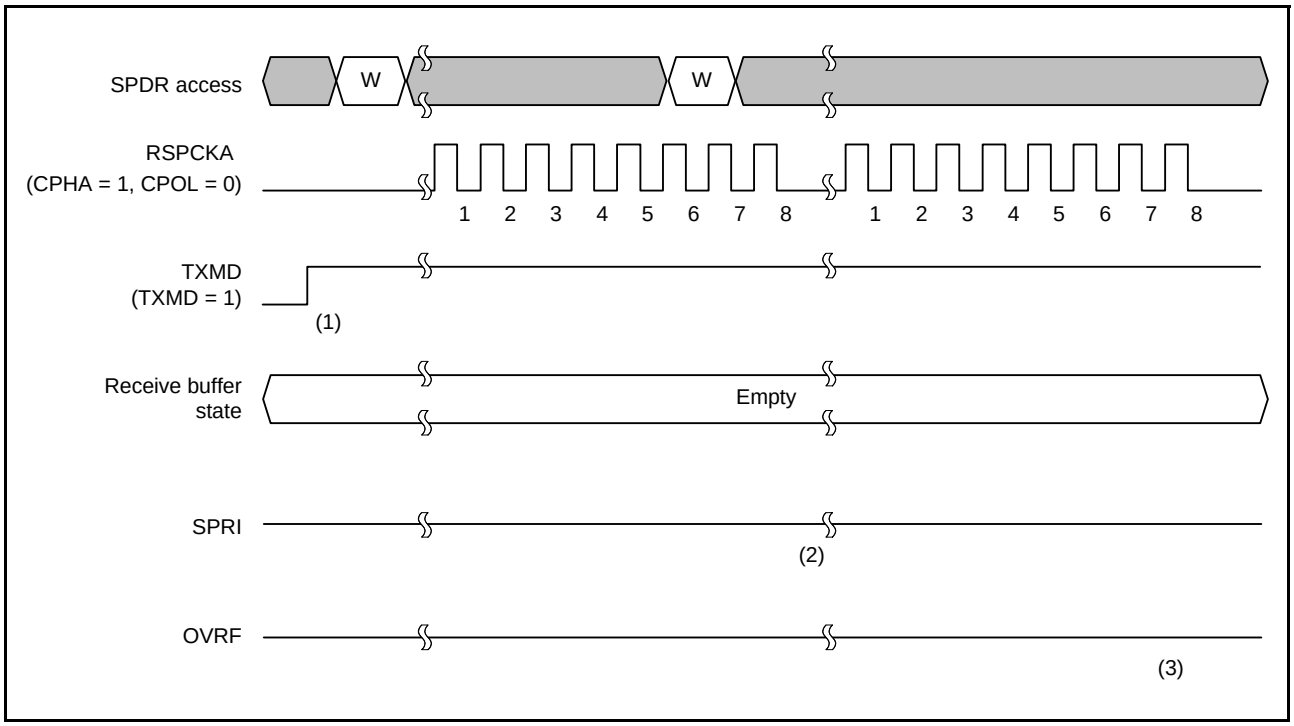


Figure 30.26 Operation Example of SPCR.TXMD = 1

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 cleared to 0 at the timings of (1) to (3).

30.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 30.27 shows an example of operation of the RSPI transmit buffer empty interrupt (SPTI) and the RSPI receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 30.27 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 30.27, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

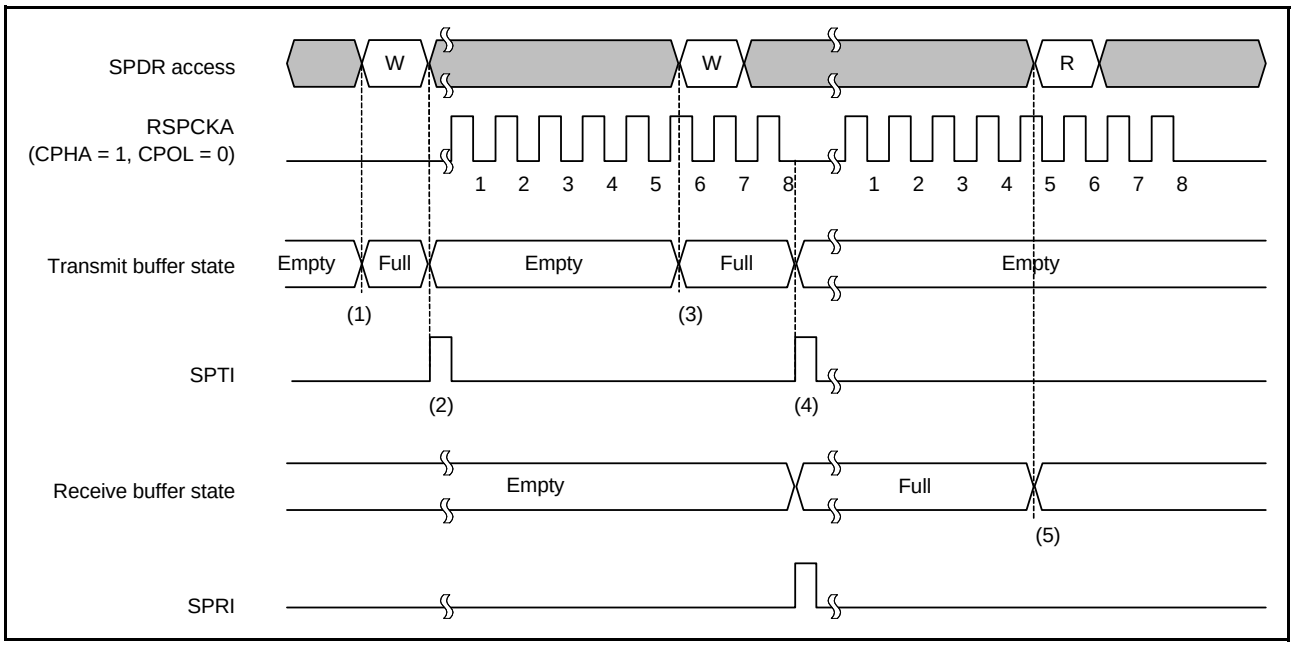


Figure 30.27 Operation Example of SPTI and SPRI Interrupts

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, see section 30.3.10, SPI Operation, and section 30.3.11, 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 an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.
If the RSPI function is disabled (the SPE bit in SPCR 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 (see section 30.3.8, 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 interrupt, set the SPCR.SPRIE bit to 1.
Transmission and reception interrupts or the corresponding IRn.IR flags (where n is the vector number) in ICU can be used to confirm the states of the transmission and reception buffers. See section 14, Interrupt Controller (ICUb), for the vector numbers.

30.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is serially transmitted, and the serially 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 30.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 30.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	Previously received serial data is output.	None
D	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing serial receive data.	Overrun error
E	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
F	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
G	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
H	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation A described in Table 30.8, 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 interrupt request.
Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. 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 C. To prevent extraneous data from being read, SPDR read operation should be executed using a receive interrupt request.
An overrun error shown in D is described in section 30.3.8.1, Overrun Error. A parity error shown in E is described in section 30.3.8.2, Parity Error. A mode fault error shown in F to H is described in section 30.3.8.3, Mode Fault Error.
For the transmit and receive interrupts, refer to section 30.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

30.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the OVRF flag in SPSR 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 30.28 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 30.28 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 30.28, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

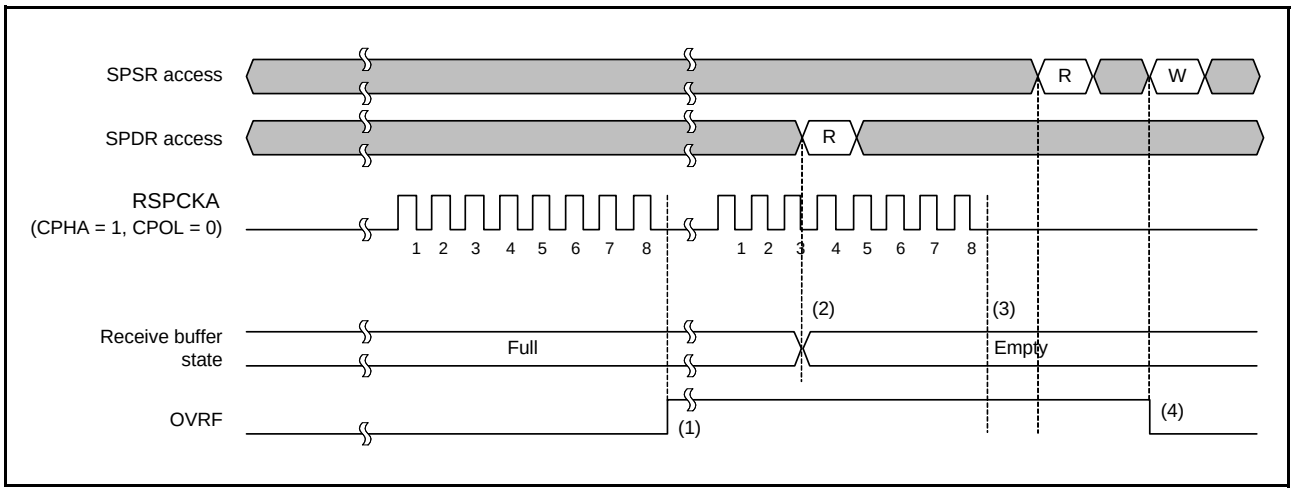


Figure 30.28 Operation Example of OVRF Flag

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 clear the OVRF flag.
3. If the serial transfer ends with the OVRF flag being 1 (an overrun error), the RSPI does not copy the data in the shift register to the receive buffer. A reception-buffer 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 the value 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, OVRF flag is cleared 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 cleared.

30.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit cleared 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 30.29 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 30.29 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 30.29, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

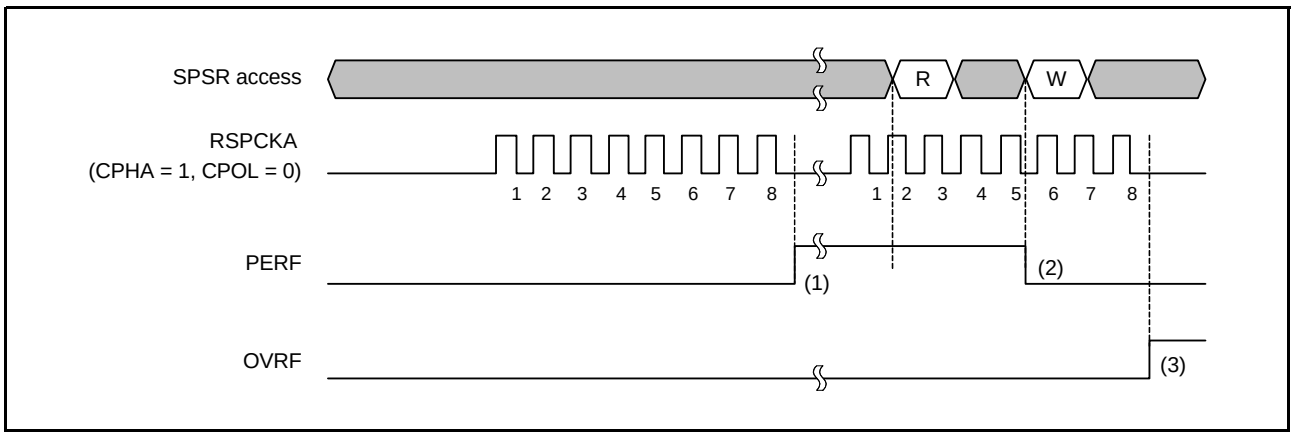


Figure 30.29 Operation Example of PERF Flag

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 the value 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is cleared 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 SPSR register or by using an RSPI error interrupt and reading 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.

30.3.8.3 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 SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (see section 30.3.9, 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 bit 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 bit must be set to 0.

30.3.9 Initializing RSPI

If the value 0 is written to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes 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 SPE bit in SPCR and initialization by a system reset.

30.3.9.1 Initialization by Clearing the SPE Bit

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (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 OVRF and MODF flags in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized, in order to disable any RSPI transmit interrupt, the value 0 should be written to the SPTIE bit simultaneously with the writing of the value 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write the value 0 to the SPTIE bit.

30.3.9.2 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 30.3.9.1, Initialization by Clearing the SPE Bit.

30.3.10 SPI Operation

30.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 30.3.8, 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, see section 30.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMDm), the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX), 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. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, see section 30.3.5, 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: SSLAi 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.

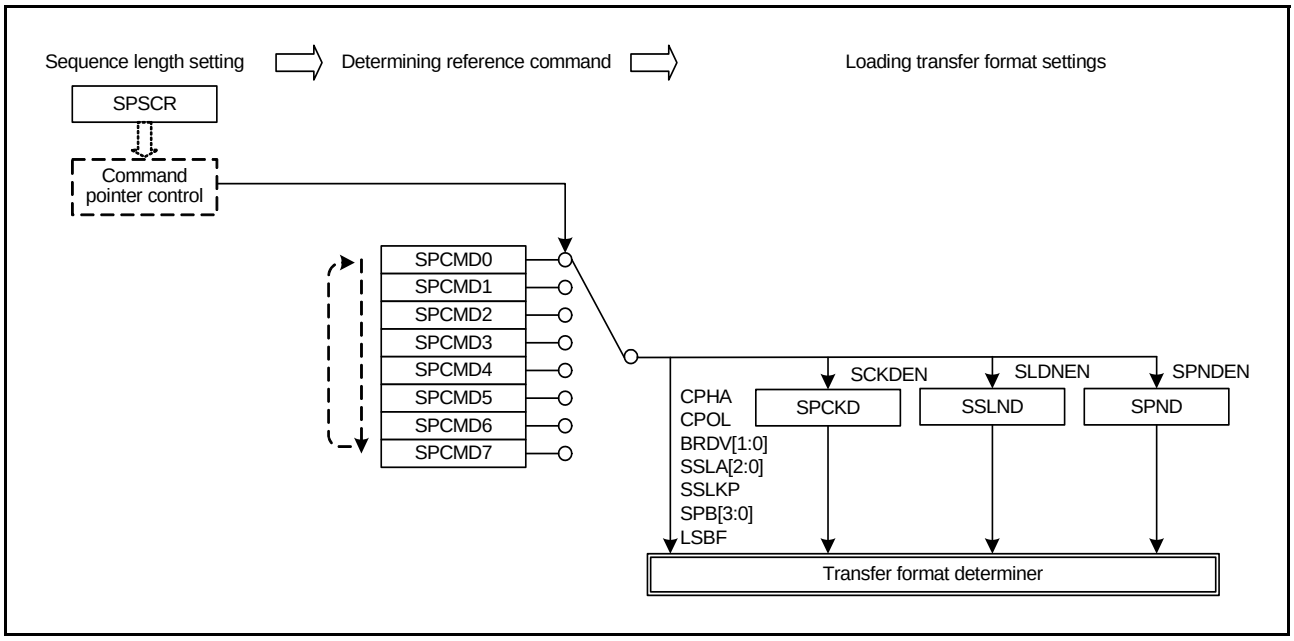


Figure 30.30 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

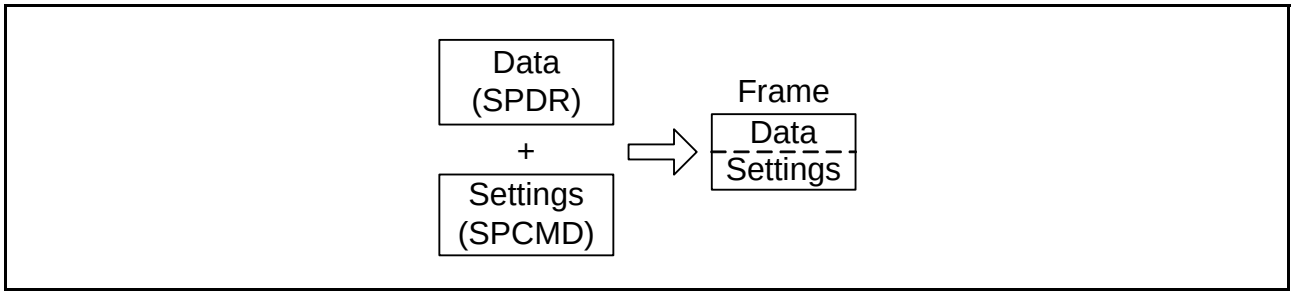


Figure 30.31 Concept of a Frame

Figure 30.32 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 30.4.

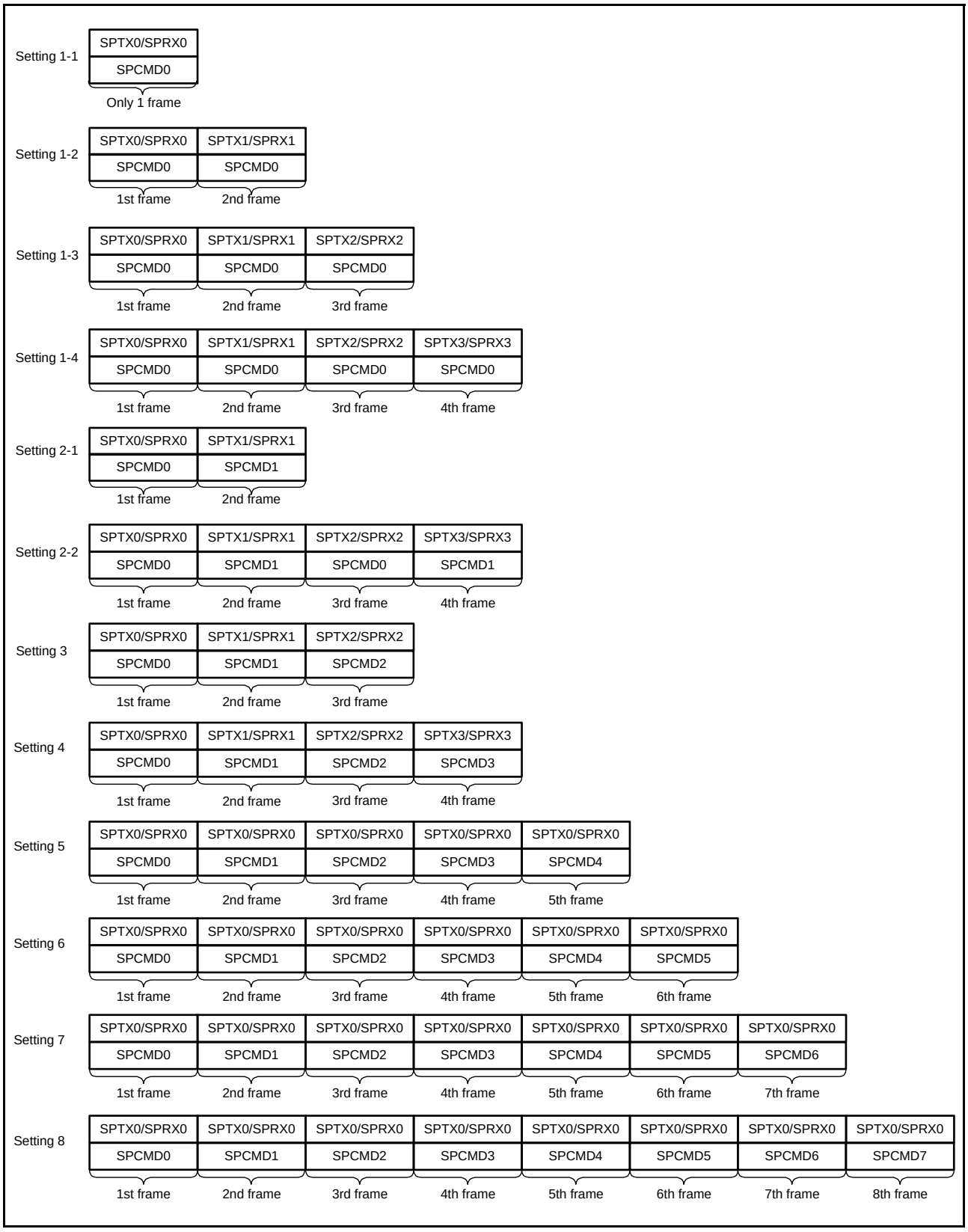


Figure 30.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMDm) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).

Figure 30.33 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 30.33. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

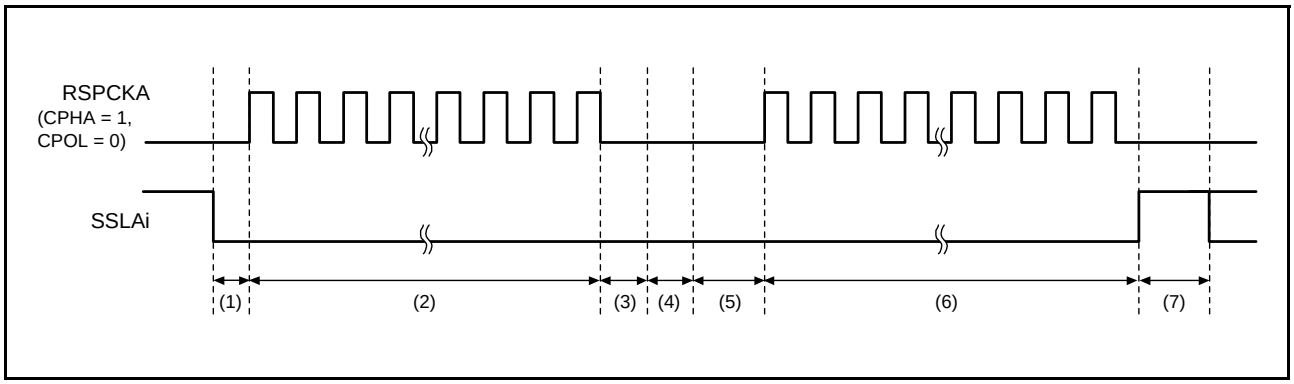


Figure 30.33 Example of Burst Transfer Operation using SSLKP Bit

(1) Based on SPCMD0, the RSPI asserts the SSLAi 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 SSLAi 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 SSLAi 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 SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 30.33) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.

The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 30.3.10, SPI Operation).

(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 30.9. For a definition of RSPCK delay, see section 30.3.5, Transfer Format.

Table 30.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on 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 30.10. For a definition of SSL negation delay, see section 30.3.5, Transfer Format.

Table 30.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on 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 30.11. For a definition of next-access delay, see section 30.3.5, Transfer Format.

Table 30.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 PCLK
1	000	1 RSPCK + 2 PCLK
	001	2 RSPCK + 2 PCLK
	010	3 RSPCK + 2 PCLK
	011	4 RSPCK + 2 PCLK
	100	5 RSPCK + 2 PCLK
	101	6 RSPCK + 2 PCLK
	110	7 RSPCK + 2 PCLK
	111	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 30.34 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, see the descriptions given in the individual blocks.

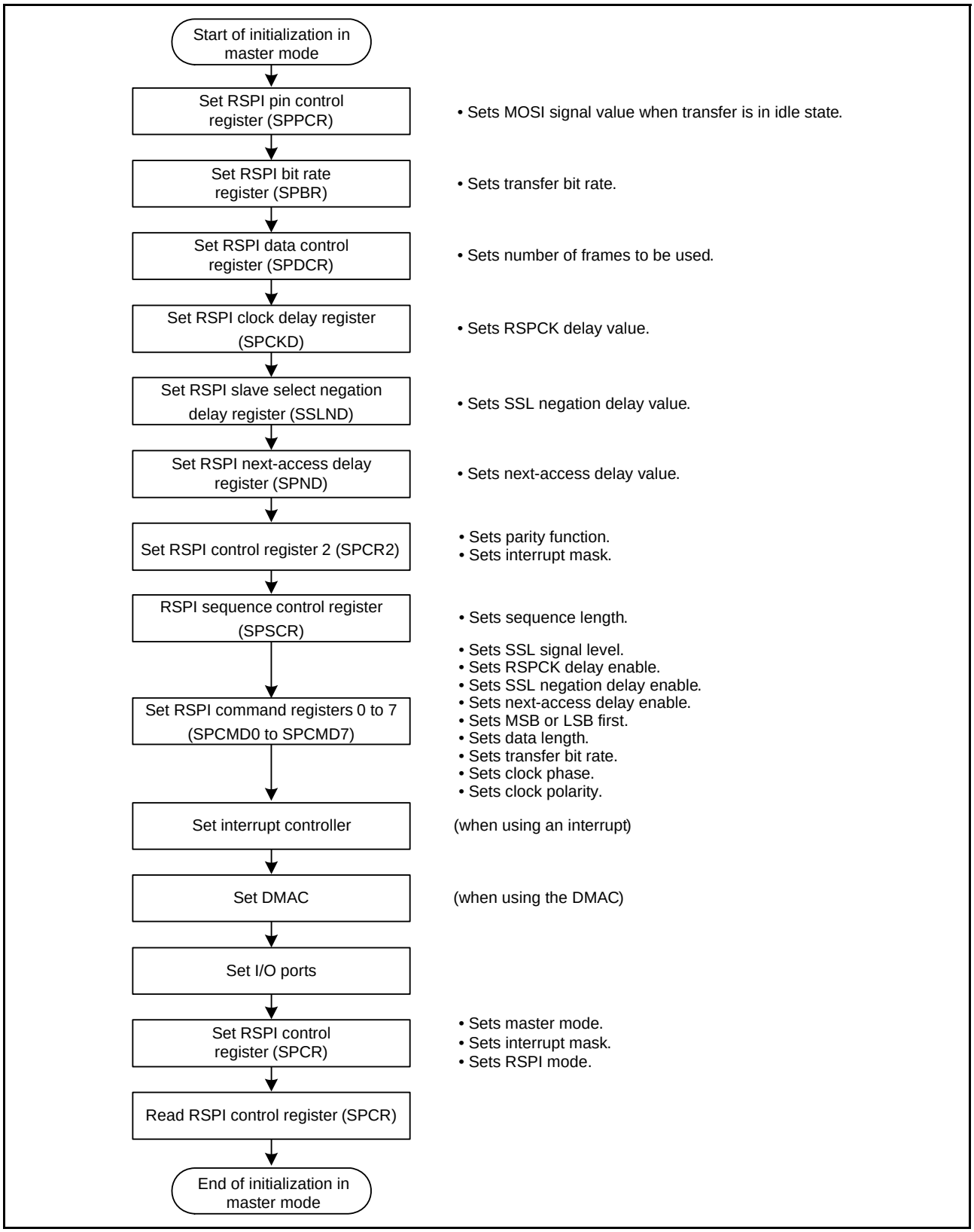


Figure 30.34 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 30.35 to Figure 30.37 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.

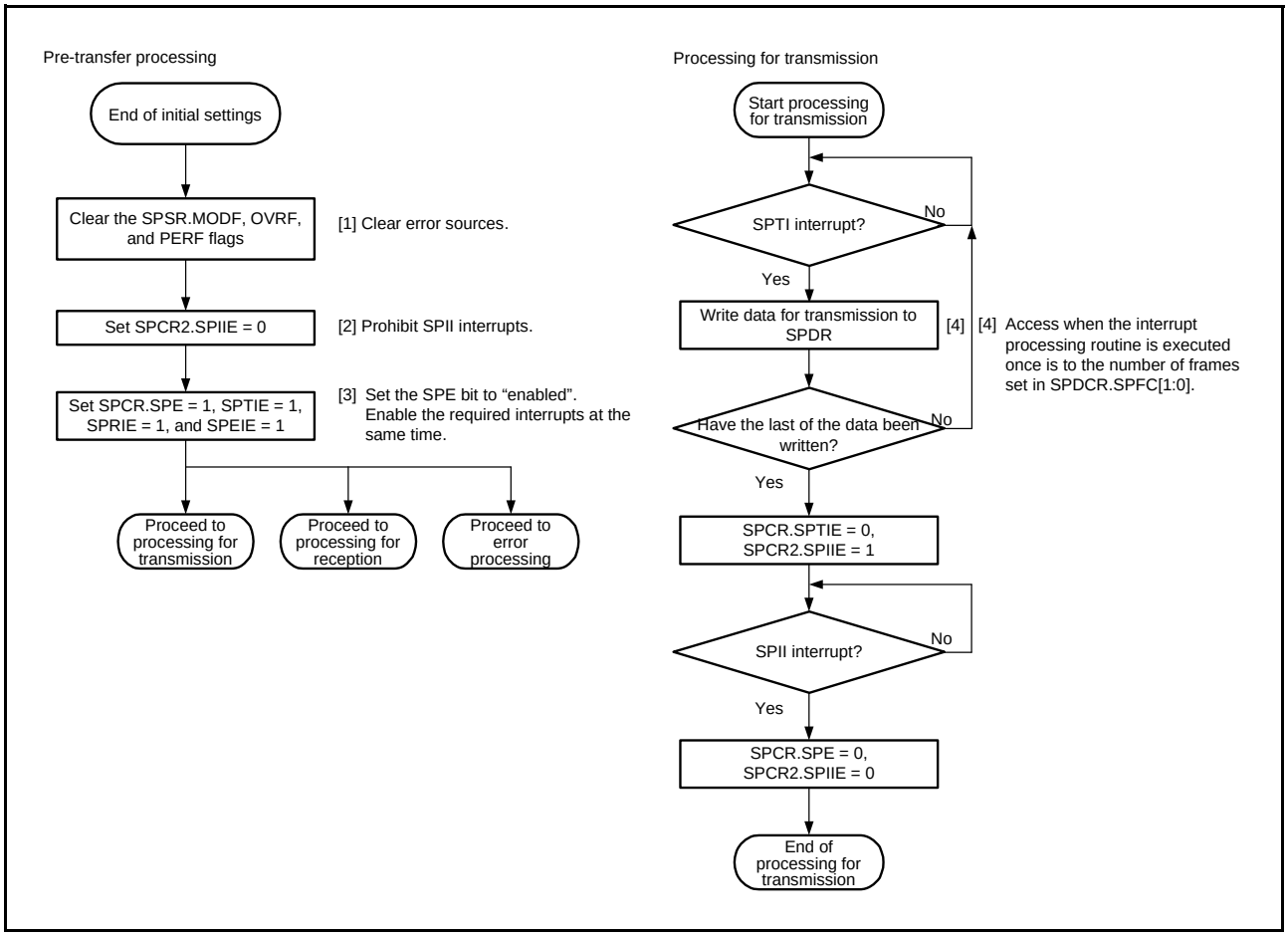


Figure 30.35 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required if reception is to proceed.

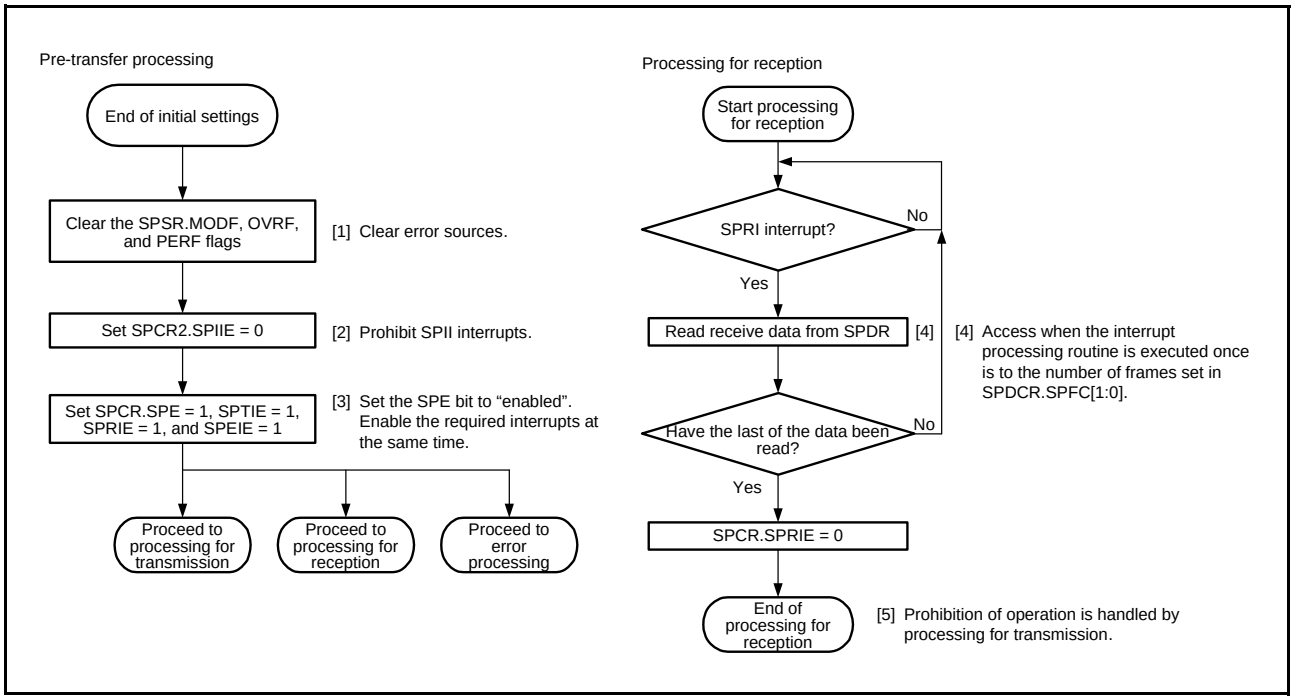


Figure 30.36 Flowchart in Master Mode (Reception)

(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 from within 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.

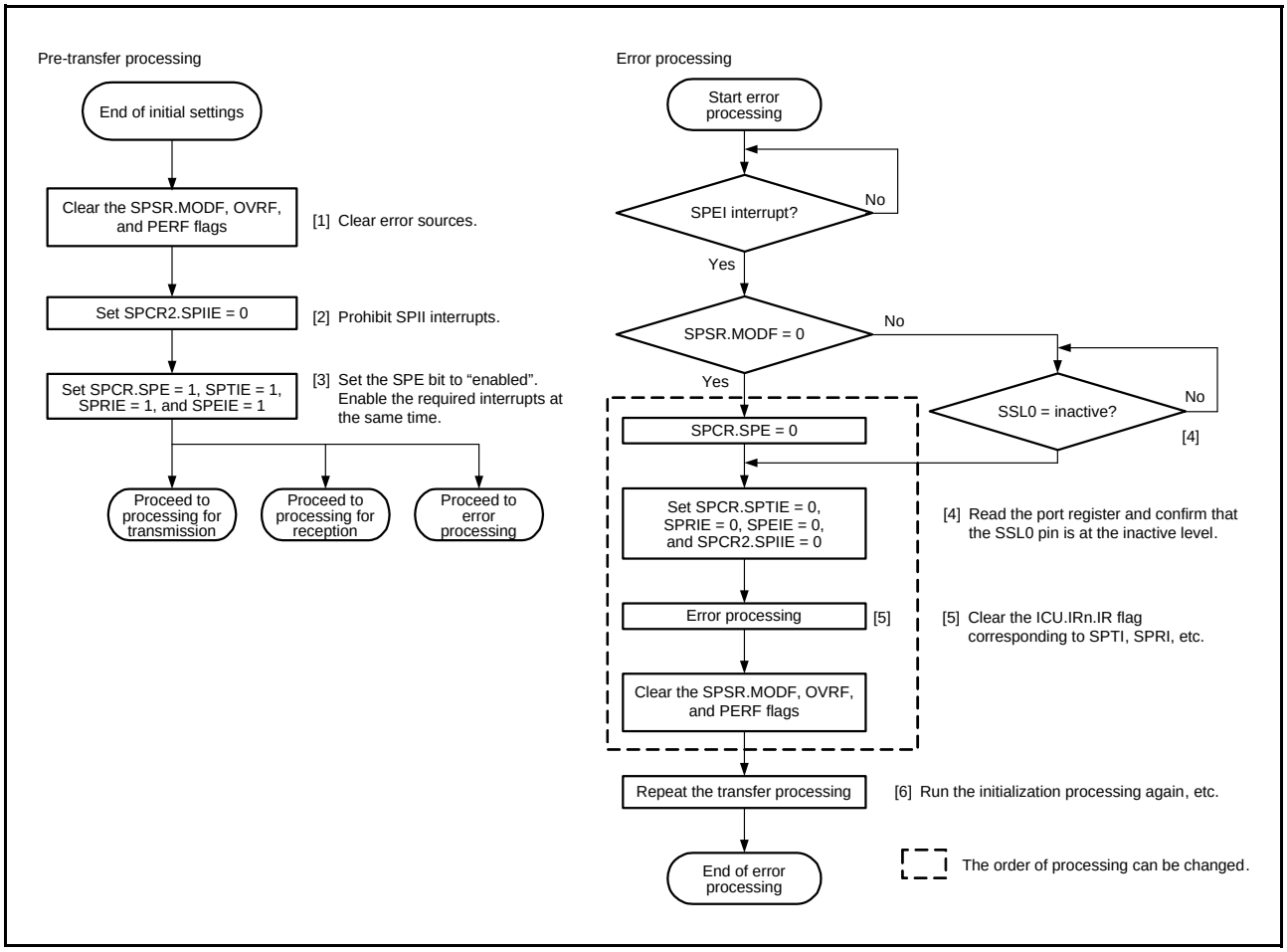


Figure 30.37 Flowchart for Master Mode (Error Processing)

30.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.
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 CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, see section 30.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR).
Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 30.3.8, 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 SSLA0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, see section 30.3.5, Transfer Format.

(3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 30.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is always fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 30.38 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, see the descriptions given in the individual blocks.

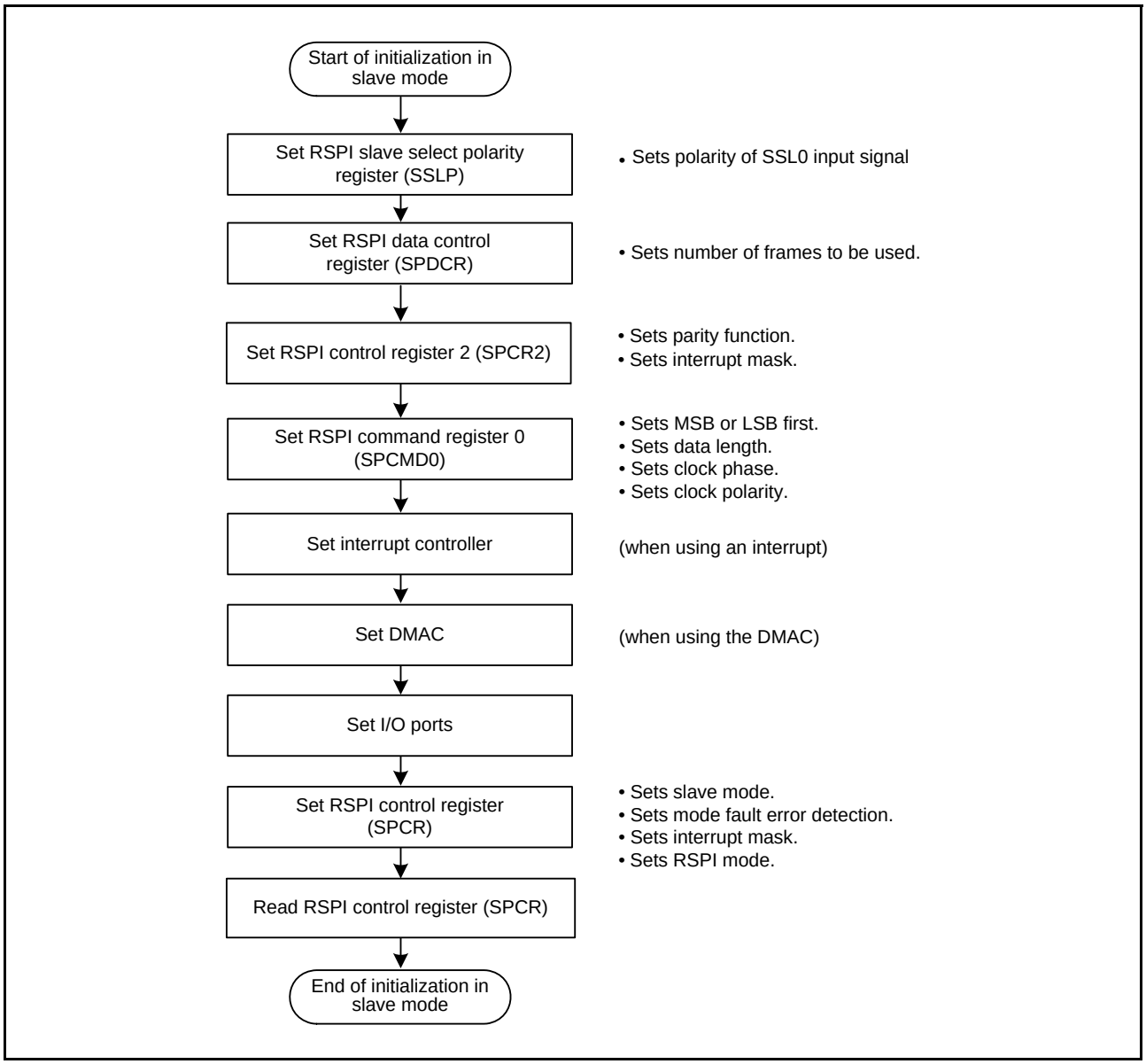


Figure 30.38 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 30.39 to Figure 30.41 show examples of the flow of software processing.

(a) Transmit Processing Flow

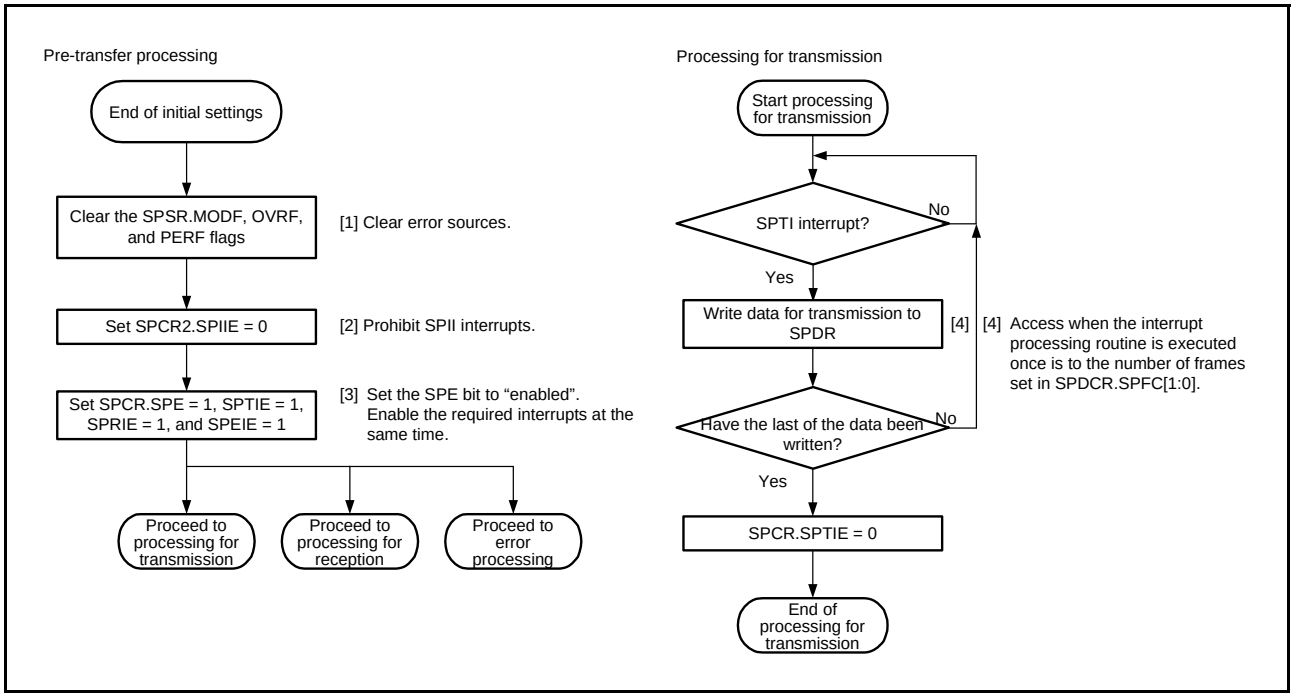


Figure 30.39 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

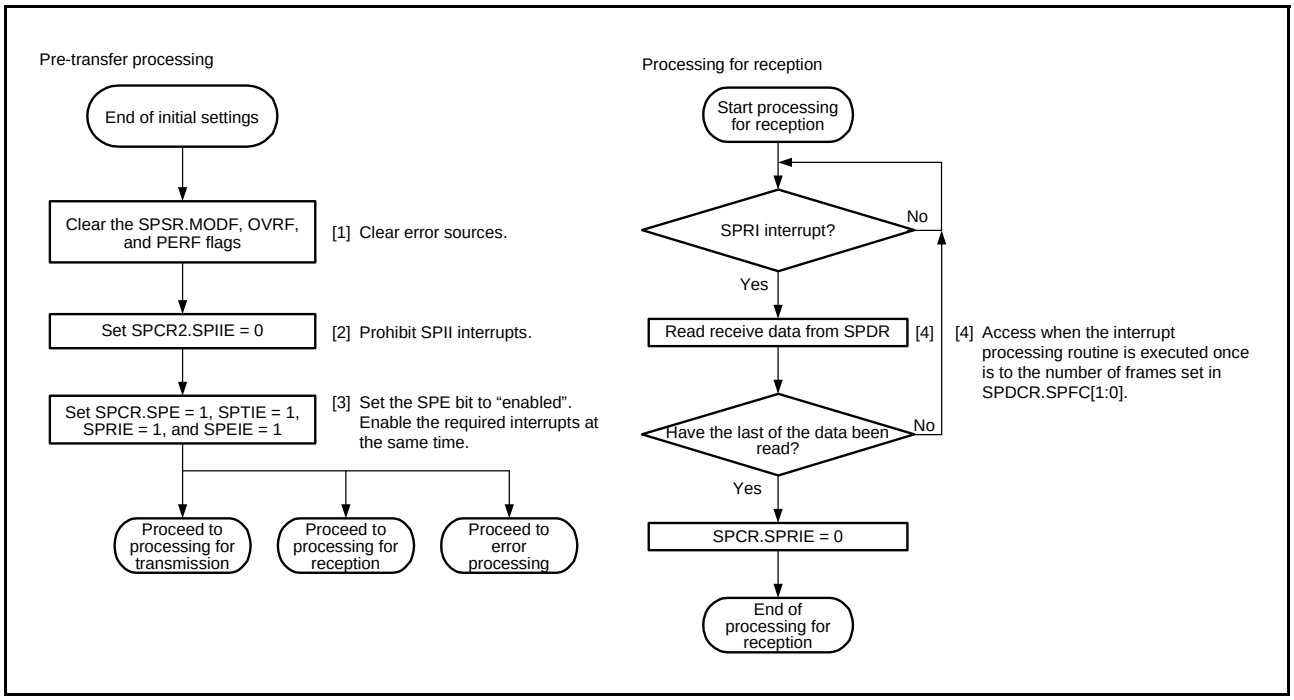


Figure 30.40 Flowchart in Slave Mode (Reception)

(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.

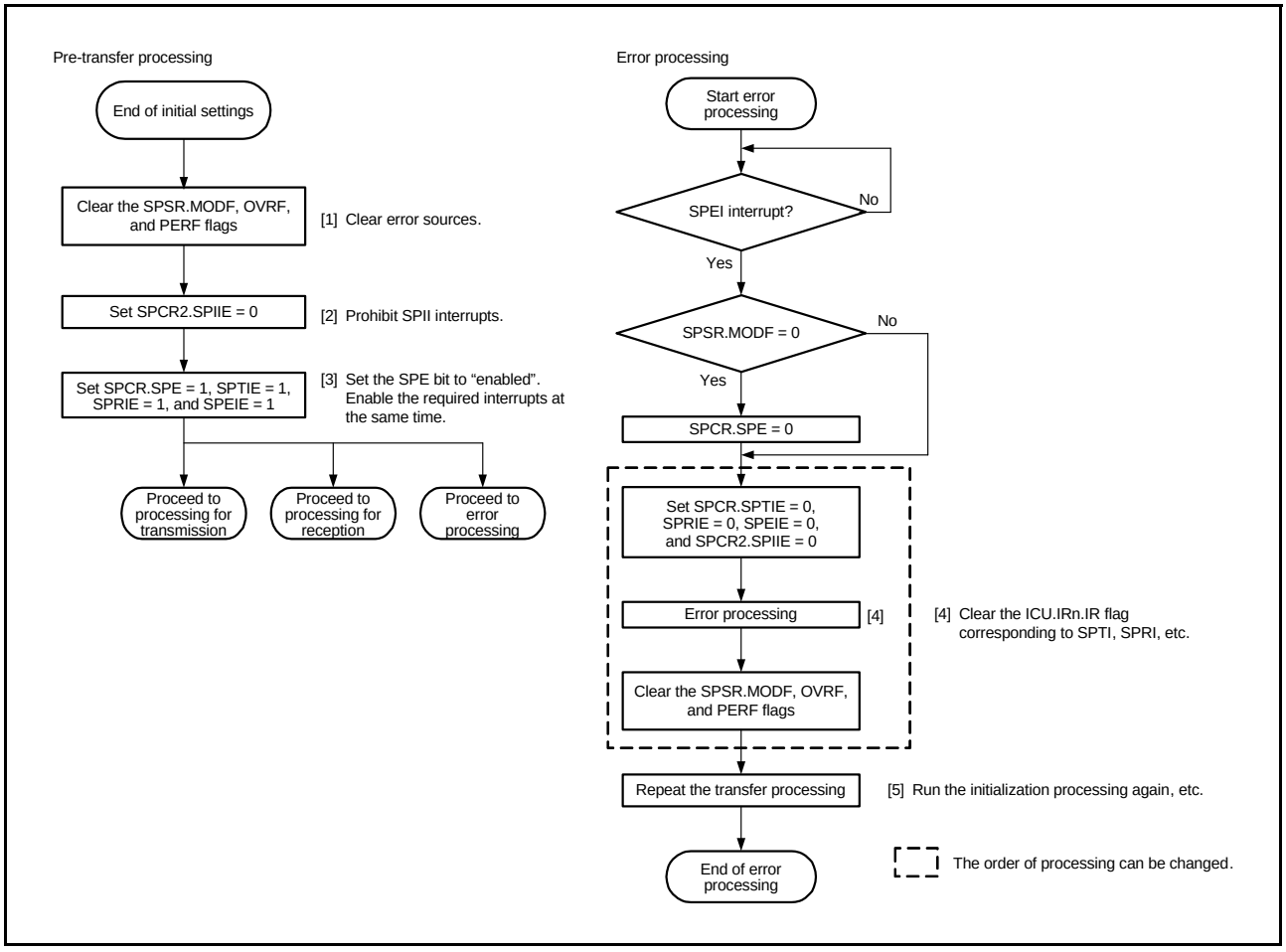


Figure 30.41 Flowchart for Slave Mode (Error Processing)

30.3.11 Clock Synchronous Operation

Setting the SPMS bit in the RSPI control register (SPCR) to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, operation is not guaranteed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

30.3.12 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 RSPI data register (SPDR) 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 transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, see section 30.3.5, Transfer Format.
However, transfer in clock-synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer, 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, see section 30.3.5, Transfer Format.
However, transfer in clock-synchronous operation is conducted without the SSLA0 output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
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: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA 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 SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 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.

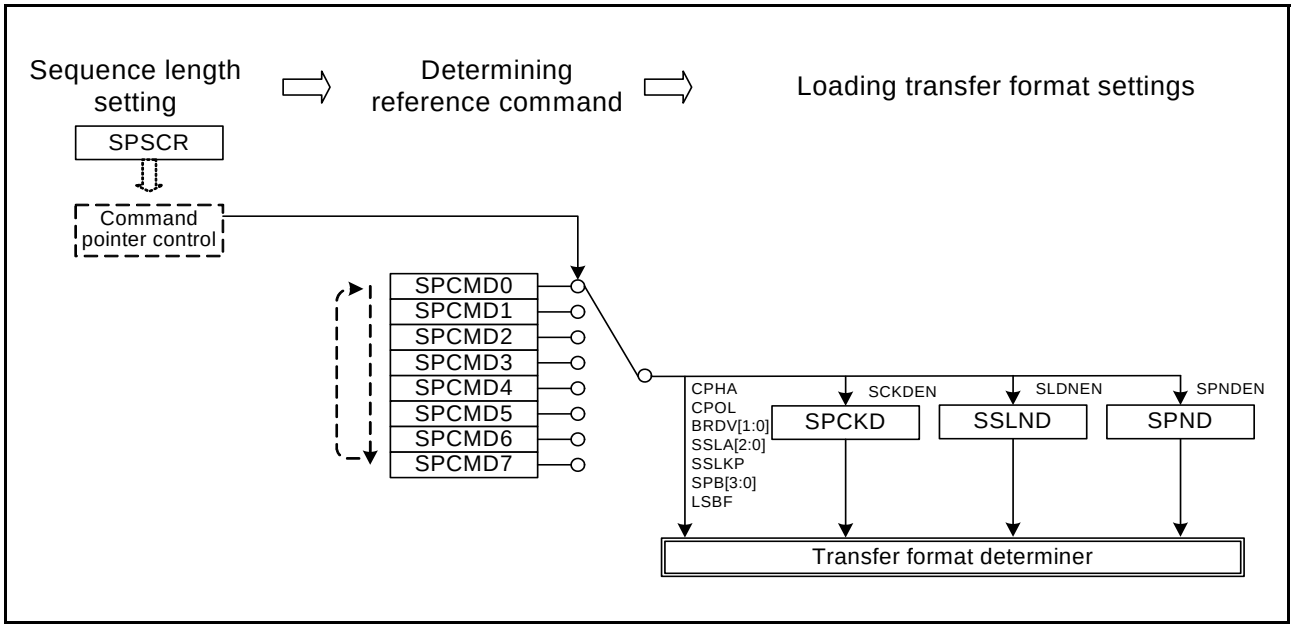


Figure 30.42 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

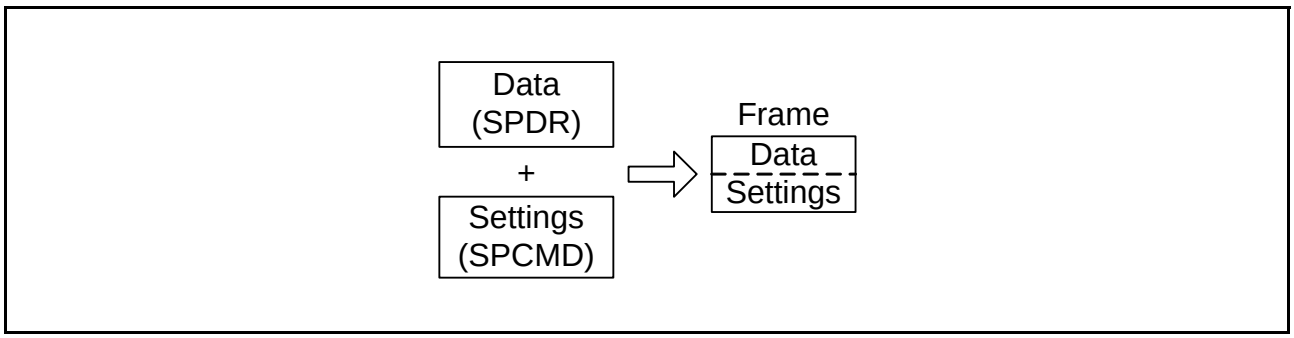


Figure 30.43 Concept of a Frame

Figure 30.44 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 30.4.

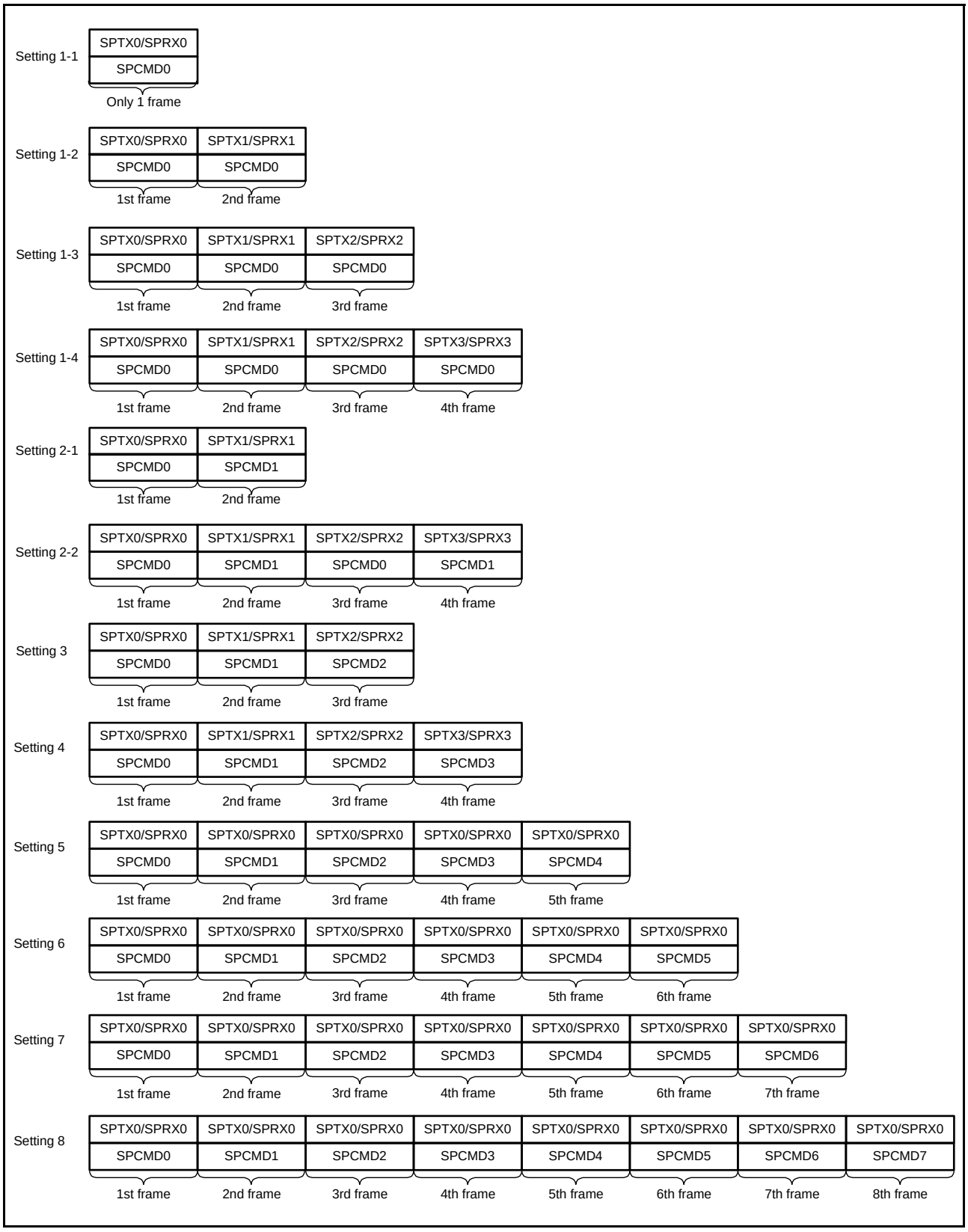


Figure 30.44 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 30.45 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, see the descriptions given in the individual blocks.

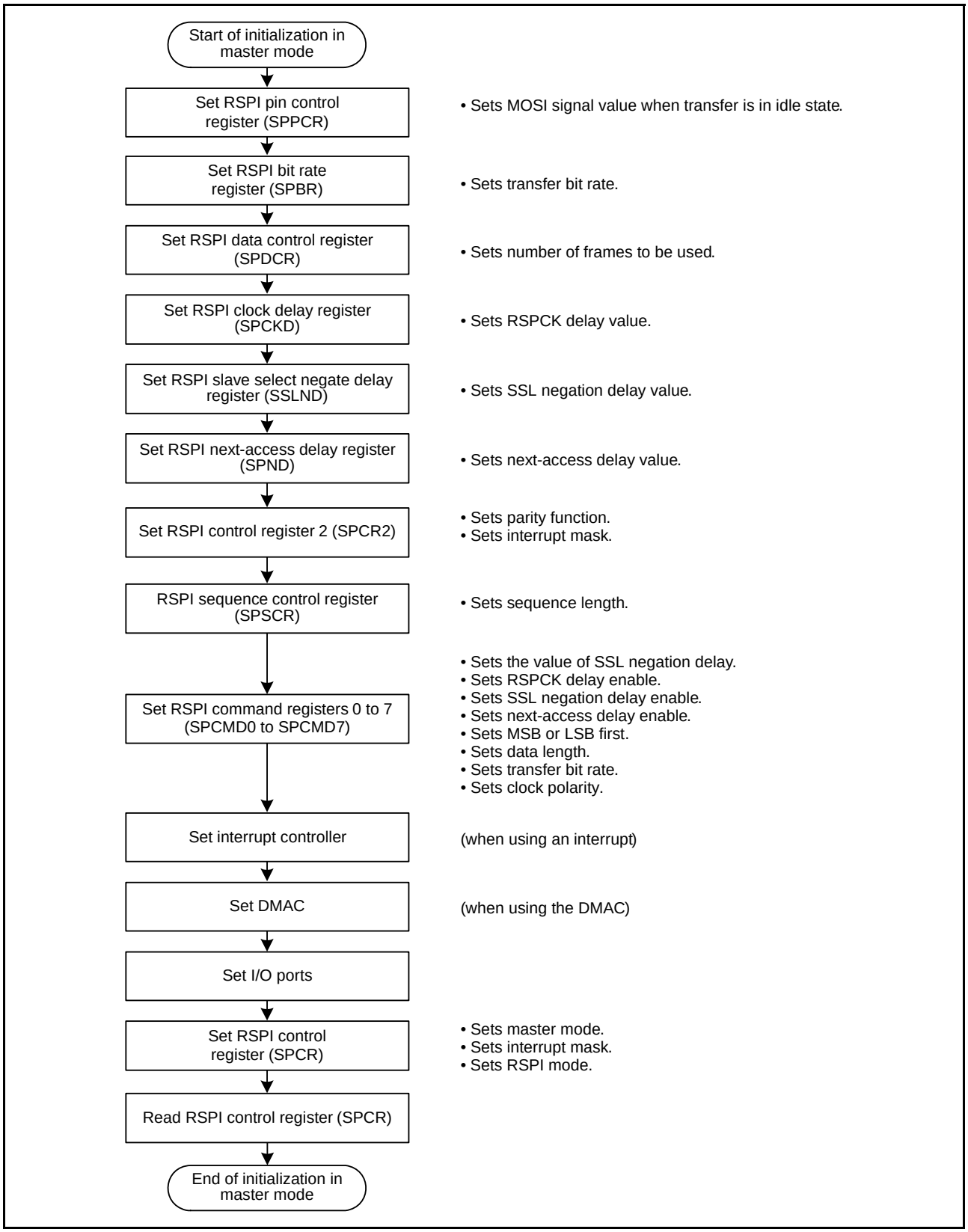


Figure 30.45 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, see section 30.3.10.1, (9) Software Processing Flow. Note that mode-fault errors will not occur.

30.3.13 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When 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 always drives the MISOA output signal.
For details on the RSPI transfer format, see section 30.3.5, 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 RSPCKA 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 RSPI data register (SPDR). 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, see section 30.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 30.46 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, see the descriptions given in the individual blocks.

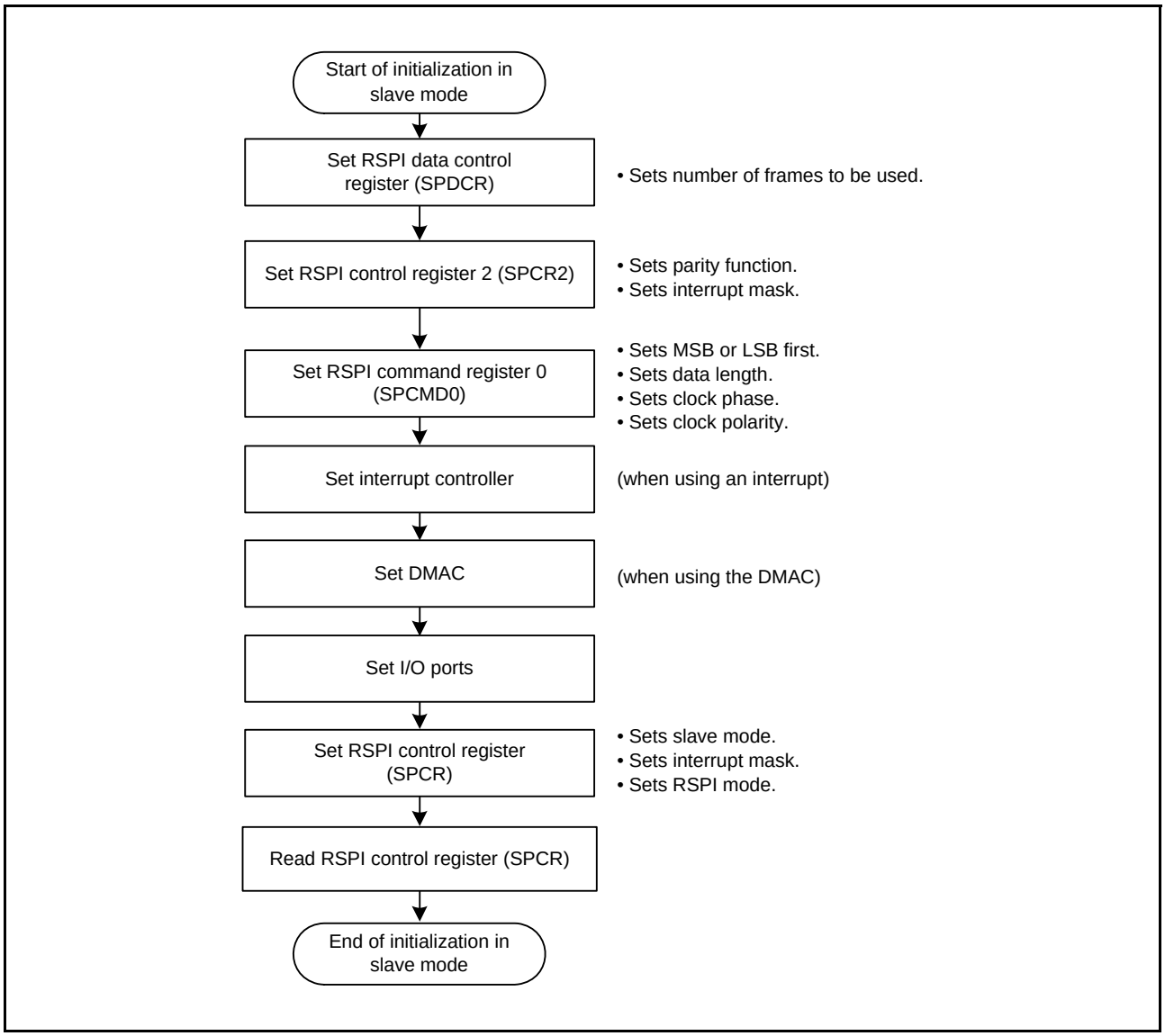


Figure 30.46 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, see section 30.3.10.2, (6) Software Processing Flow. Note that mode-fault errors will not occur.

30.3.14 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 30.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 30.47 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 30.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Reversed transmit data
1	0	Transmit data
1	1	Transmit data

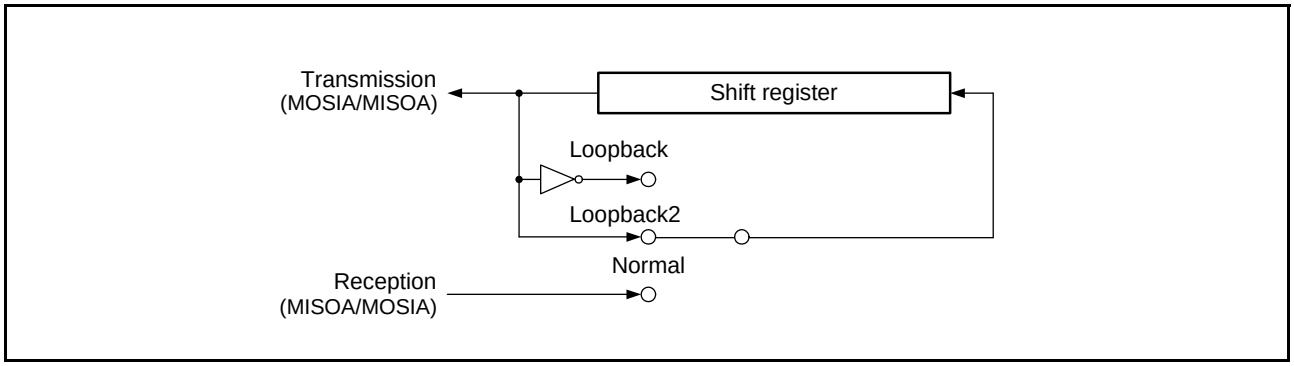


Figure 30.47 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

30.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 30.48.

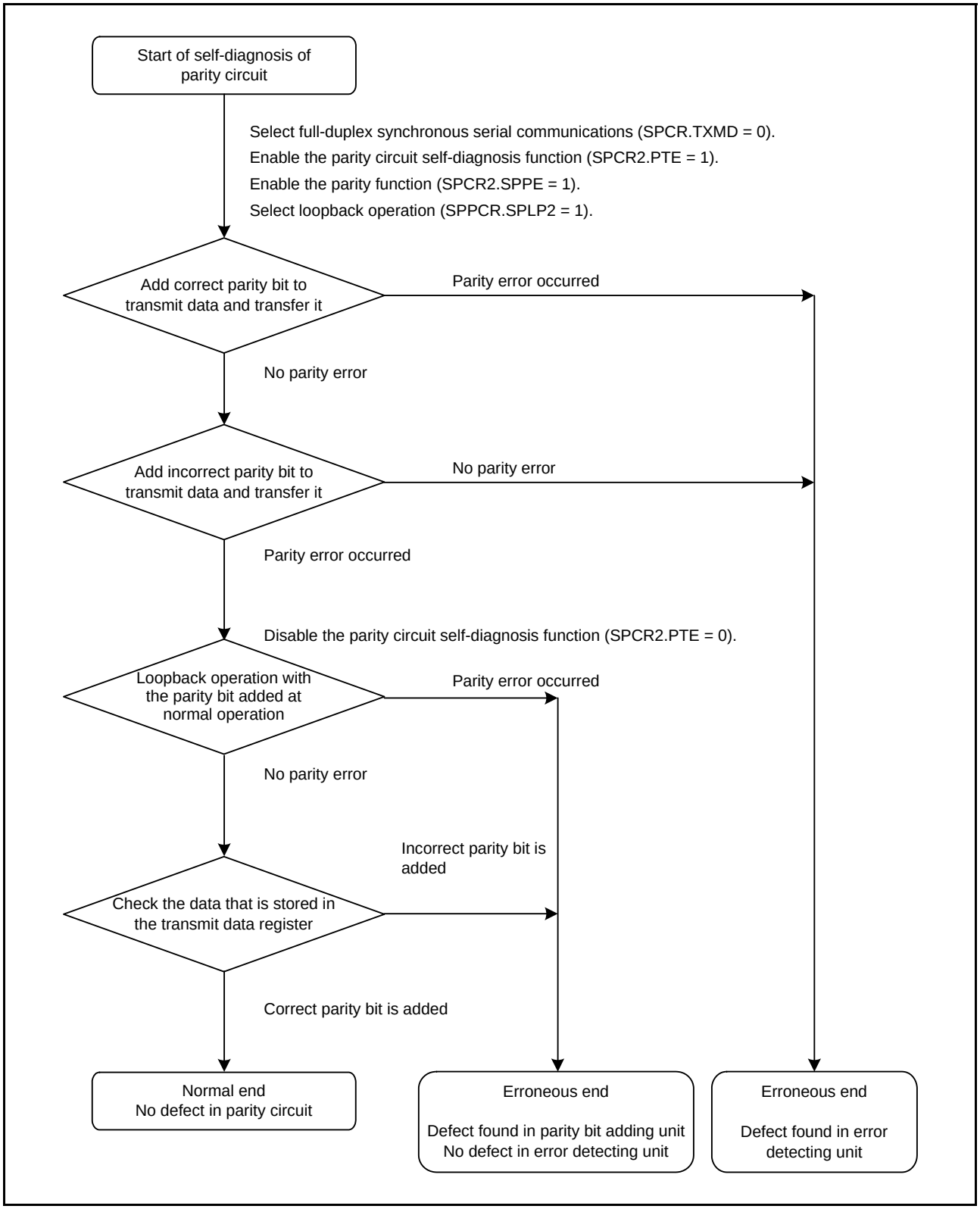


Figure 30.48 Flowchart for Self-Diagnosis of Parity Circuit

30.3.16 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 30.13. An interrupt is generated on satisfaction of an interrupt condition in Table 30.13. 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 16, DMA Controller (DMACA), or section 17, 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 30.13 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

30.4 Event Link Output

The event link controller (ELC) is capable of producing the following event-output signals. The event link output signal is output regardless of the interrupt enable bit setting.

30.4.1 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.

30.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

30.4.3 Mode Fault, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 30.14 lists the occurrence conditions of a mode fault event.

Table 30.14 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLA0 Pin	Remarks
Master (MSTR bit = 1)	1	Active	Under this condition (when the MSTR bit is 1 and the SPCR.MODFEN bit is 1), mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (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.

30.4.4 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 SPE bit in the SPCR is set to 0 (RSPI is initialized).

30.4.5 Transmit End Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output when the IDLNF bit changes from 1 to 0.

Table 30.15 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.

30.5 Usage Note

30.5.1 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. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

30.5.2 Cautionary Note on the Low Power Consumption Functions

When a low power consumption function is to be used to lower power consumption by the RSPI, use the low power consumption function after the SPCR.SPE bit is set to 0 and transfer ends.

30.5.3 Points to Note 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. Clear 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. Clear the ICU.IRn.IR flag to 0.

31. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes of data blocks.

31.1 Overview

Table 31.1 lists the specifications of the CRC calculator, and Figure 31.1 shows a block diagram of the CRC calculator.

Table 31.1 Specifications of CRC

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
Data block size	8 bits
CRC processor unit	Operation executed on eight 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	CRC code generation for LSB-first or MSB-first communication selectable
Low power consumption function	Module stop state can be set

Note 1. The circuit does not have functionality to divide data for calculation into a data-block size. Write data in 8-bit units.

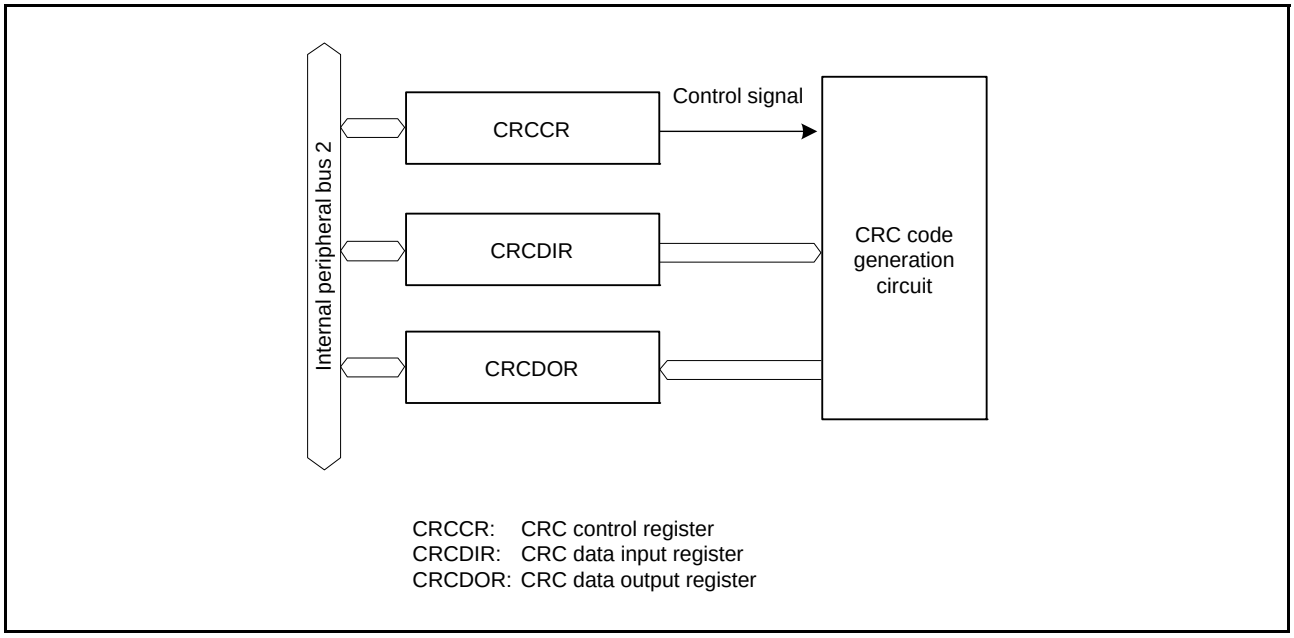


Figure 31.1 Block Diagram of CRC Calculator

31.2 Register Descriptions

31.2.1 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: $X^8 + X^2 + X + 1$ 1 0: $X^{16} + X^{15} + X^2 + 1$ 1 1: $X^{16} + X^{12} + X^5 + 1$	R/W
b2	LMS	CRC Calculation Switching	0: Performs CRC operation for LSB-first communication. The lower-order byte (bits 7 to 0) is the first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes. 1: Performs CRC operation for MSB-first communication. The higher-order byte (bits 15 to 8) is first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes.	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: Clear the CRCDOR register*1 This bit is read as 0.	W

Note 1. Only 1 can be written.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is cleared to 0000h.

This bit is read as 0. Only 1 can be written.

31.2.2 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 an 8-bit readable/writable register, to which the bytes to be CRC-operated are written.

31.2.3 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 16-bit readable/writable register that contains the result of CRC calculation.
In general, the value will be 0 if there is no CRC error when the calculated CRC code matches the CRC code that continues on, for verification, from the transferred data.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the lower-order byte (b7 to b0). The higher-order byte (b15 to b8) is not updated.

31.3 Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.
The following figures show examples in which the CRCCR.GPS[1:0] bits are set to 11b so the CRC code is calculated by using a 16-bit CRC (with the polynomial $X^{16} + X^{12} + X^5 + 1$), and the CRC code is calculated for the value “F0h”.
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.

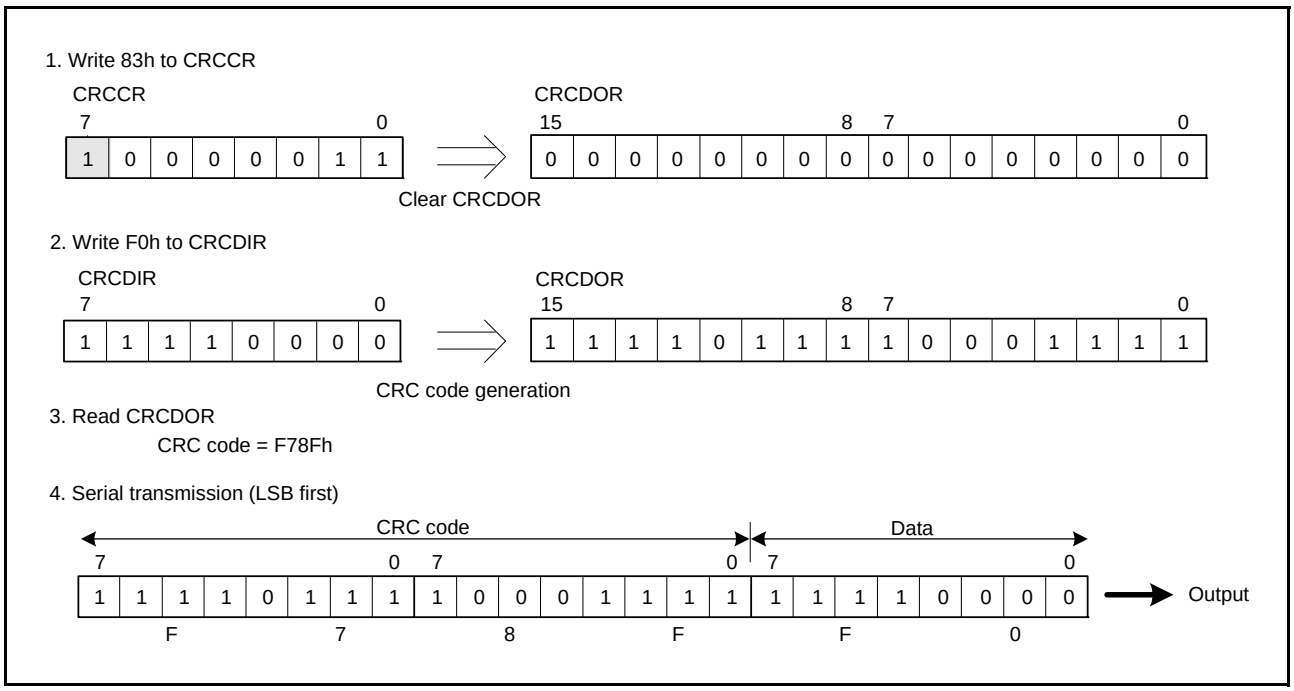


Figure 31.2 LSB-First Data Transmission

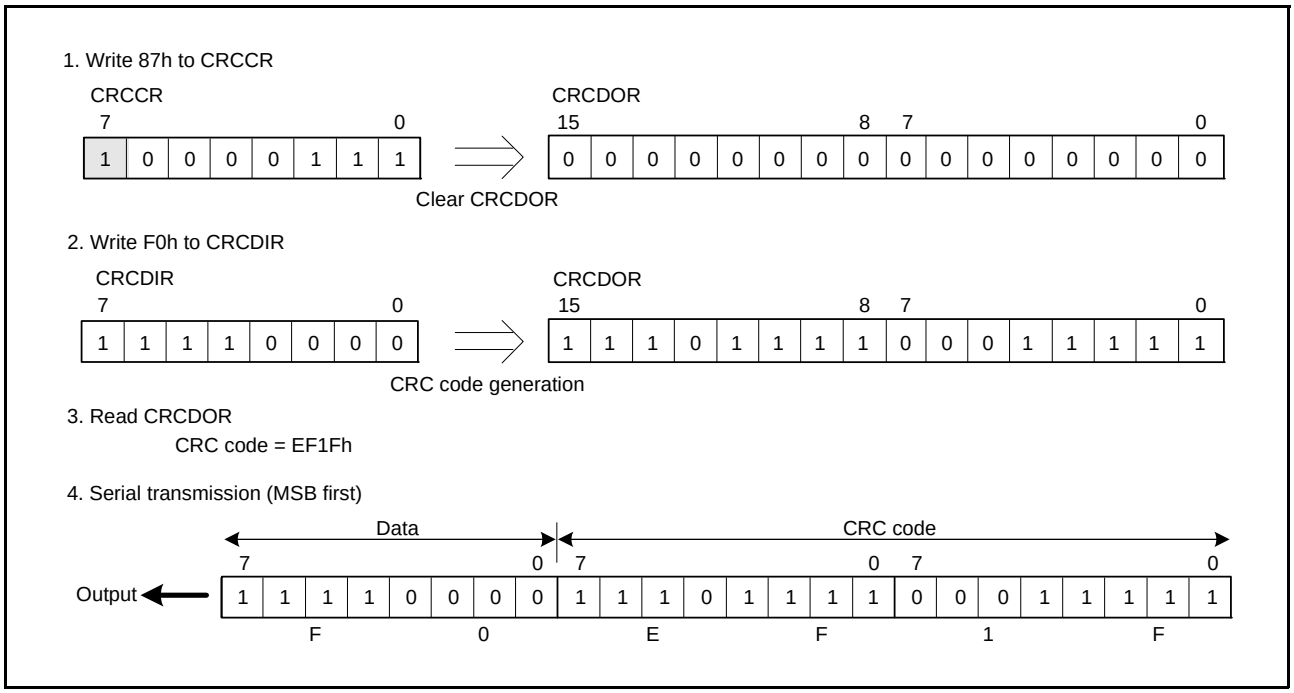


Figure 31.3 MSB-First Data Transmission

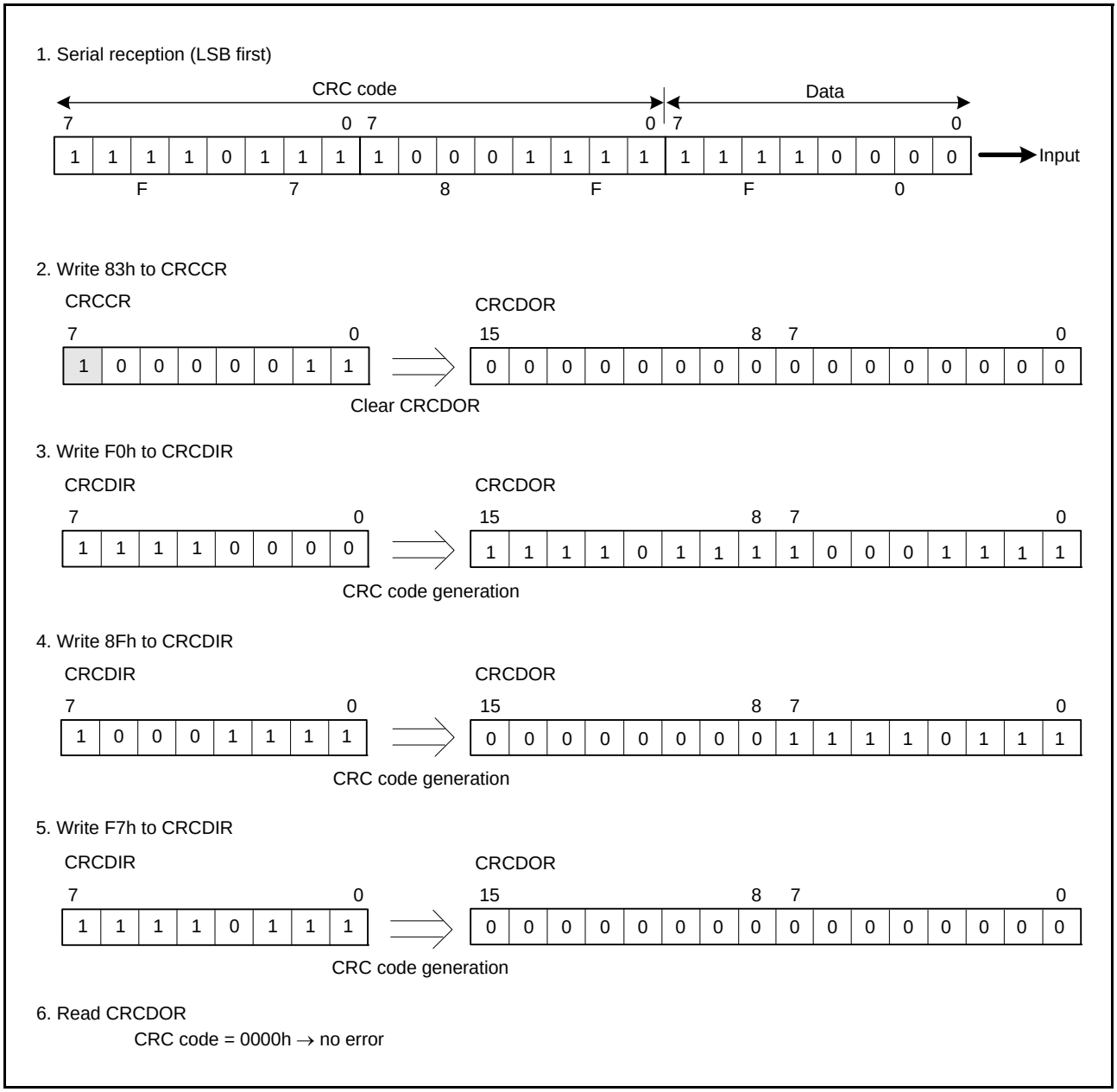


Figure 31.4 LSB-First Data Reception

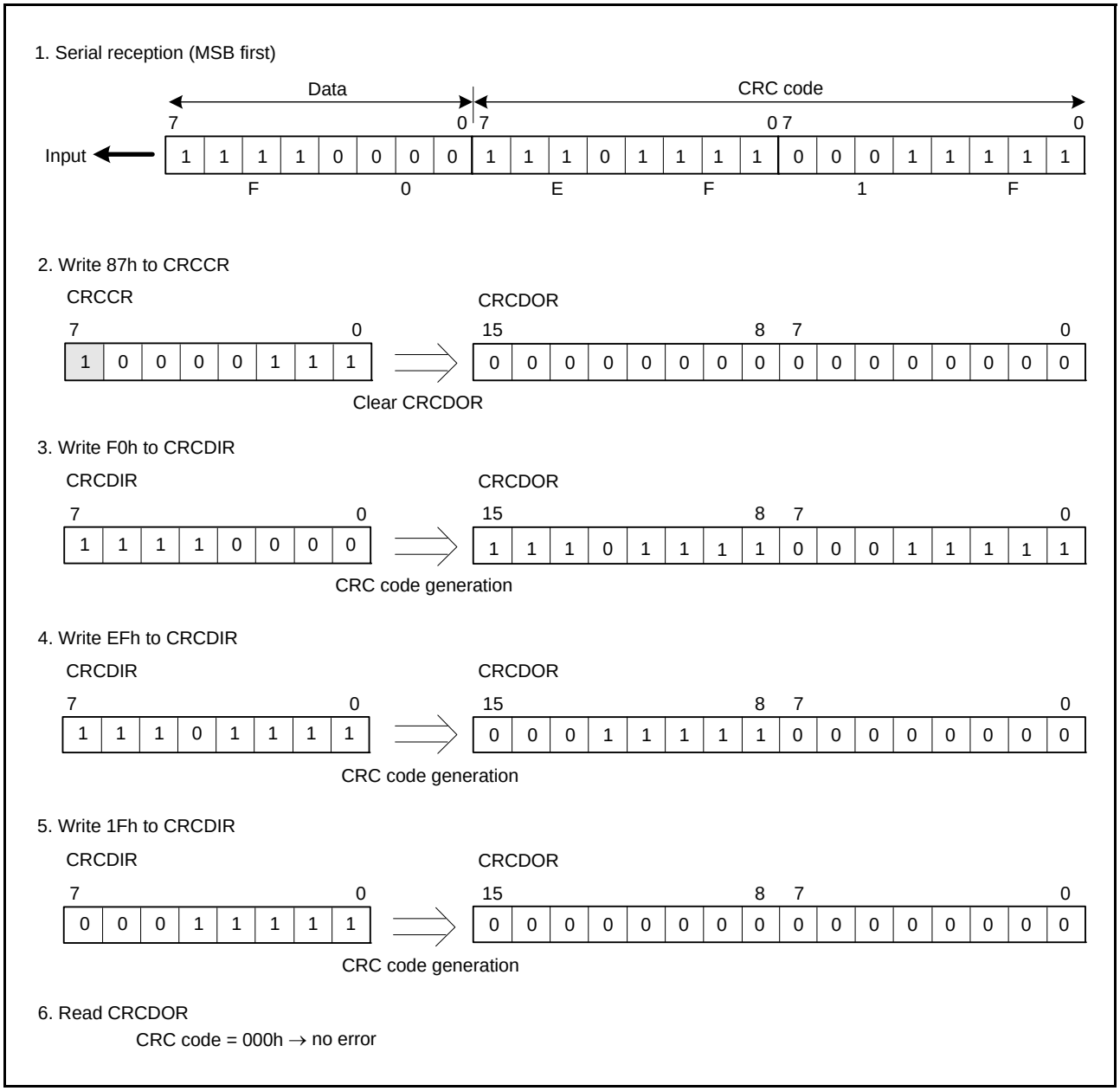


Figure 31.5 MSB-First Data Reception

31.4 Usage Notes

31.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for operation of the CRC calculator to be stopped. Register access is enabled by clearing the module stop state. For details, see section 11, Low Power Consumption.

31.5 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB-first or MSB-first.

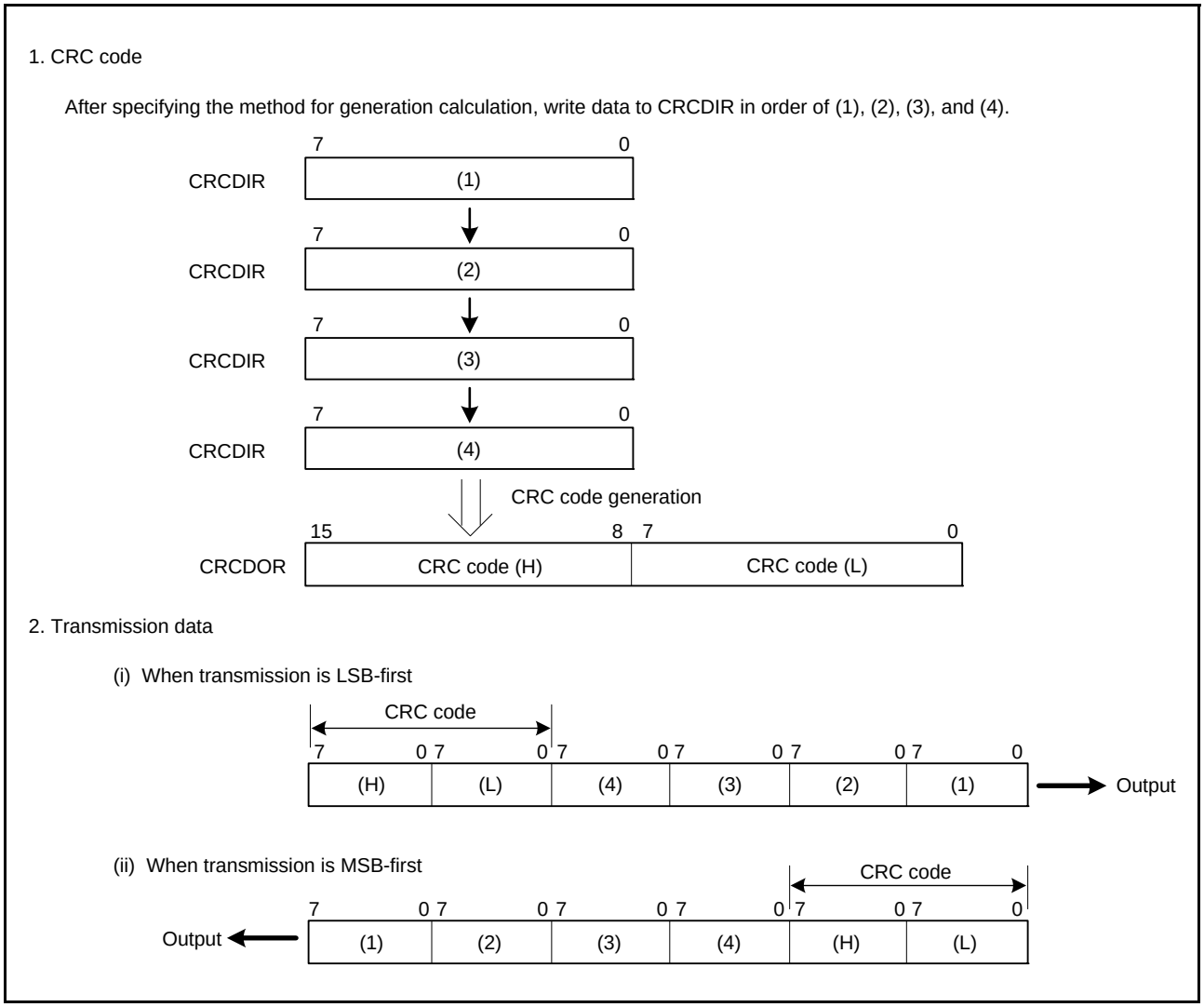


Figure 31.6 LSB-First and MSB-First Data Transmission

32. 12-Bit A/D Converter (S12ADb)

32.1 Overview

The RX220 Group 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 32.1 lists the specifications of the 12-bit A/D converter and Table 32.2 indicates the functions of the 12-bit A/D converter. Figure 32.1 shows a block diagram of the 12-bit A/D converter.

Table 32.1 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	1.56 µs per channel (when A/D conversion clock ADCLK = 32 MHz)
A/D conversion clock	Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency division ratio should be one of the following. PCLK to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8, 2:1, 4:1 ADCLK is set using the clock generation circuit.
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. 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.
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.

Table 32.1 Specifications of 12-Bit A/D Converter (2 / 2)

Item	Description
Conditions of A/D conversion start	• Software trigger • Synchronous trigger Trigger by MTU or ELC • Asynchronous trigger A/D conversion can be triggered from the ADTRG0# pin.
Function	• 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 PCLK 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. When the module stop state is canceled, A/D conversion can be started after 1 μs has elapsed.

Table 32.2 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
		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.

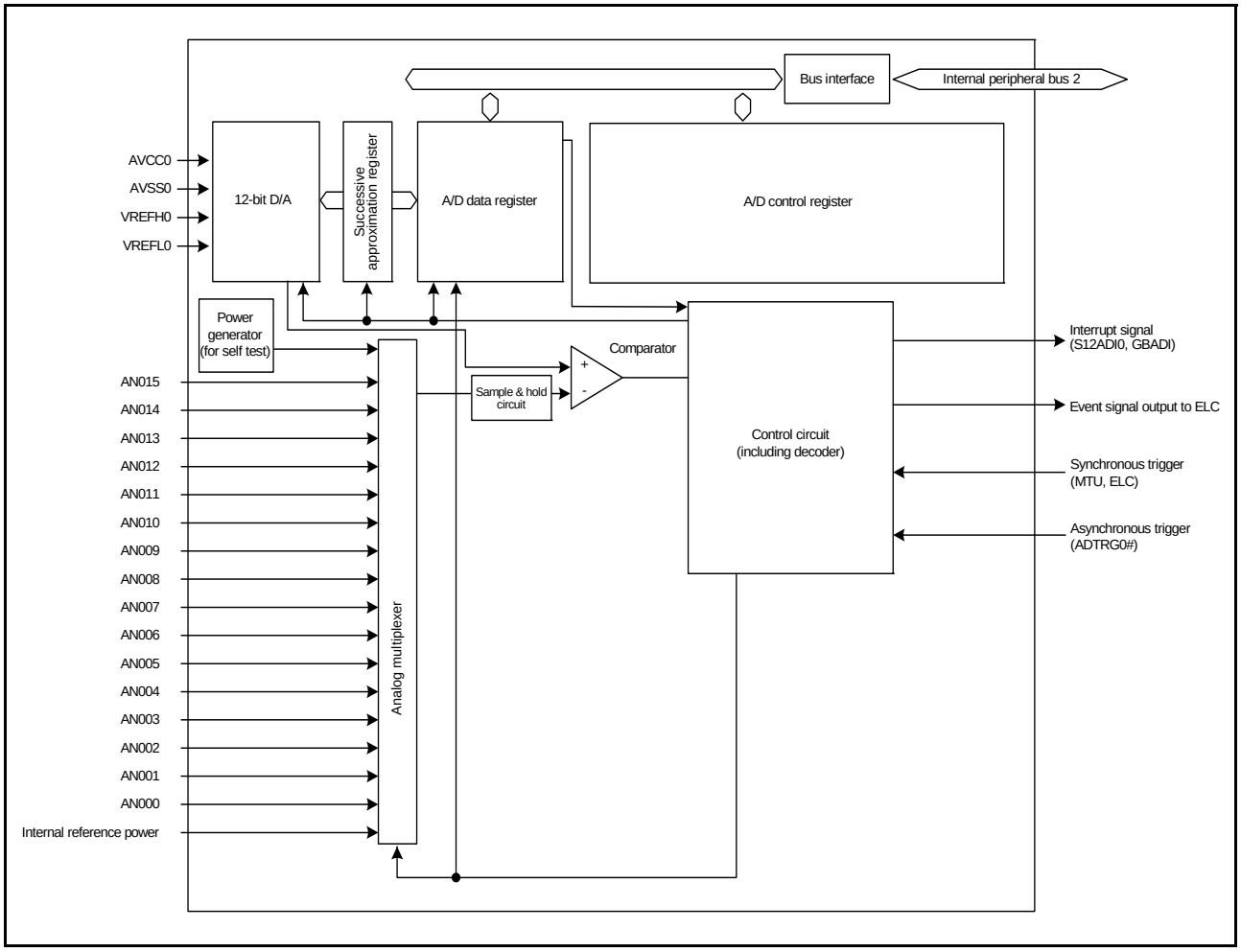


Figure 32.1 Block Diagram of 12-Bit A/D Converter

Table 32.3 lists the input pins of the 12-bit A/D converter.

Table 32.3 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 ground pin
AN000 to AN015	Input	Analog input pins
ADTRG0#	Input	External trigger input pin for starting A/D conversion

32.2 Register Descriptions

32.2.1 A/D Data Registers y (ADDRy) (y = 0 to 15)

ADDRy are 16-bit read-only registers which store the A/D conversion results of channels AN000 to AN015.
The A/D data registers use the following different formats 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: ADDR0:0008 9020h to ADDR15:0008 903Eh

	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: ADDR0:0008 9020h to ADDR15:0008 903Eh

	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: ADDR0:0008 9020h to ADDR15:0008 903Eh

	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 ADDRy show the value added by the A/D-converted value of the respective channels. 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.

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)

32.2.2 A/D Data Duplication Register (ADDBLDR)

ADDBLDR is a 16-bit read-only register used in double trigger mode. ADDBLDR holds the results of A/D conversion of the analog input of the channel selected for data duplication when the conversion is started by the second trigger. ADDBLDR uses the following different formats 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 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]	—	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 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
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 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 in ADDBLDR show the value added by the A/D-converted value of the respective channels. 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.

32.2.3 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.

32.2.4 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 32.2.10, 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 32.2.10, 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

32.2.5 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 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 PCLK 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 32.4 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 32.4 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 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 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 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 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 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 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.

32.2.6 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 Channel 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.

32.2.7 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.

32.2.8 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 32.2 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).

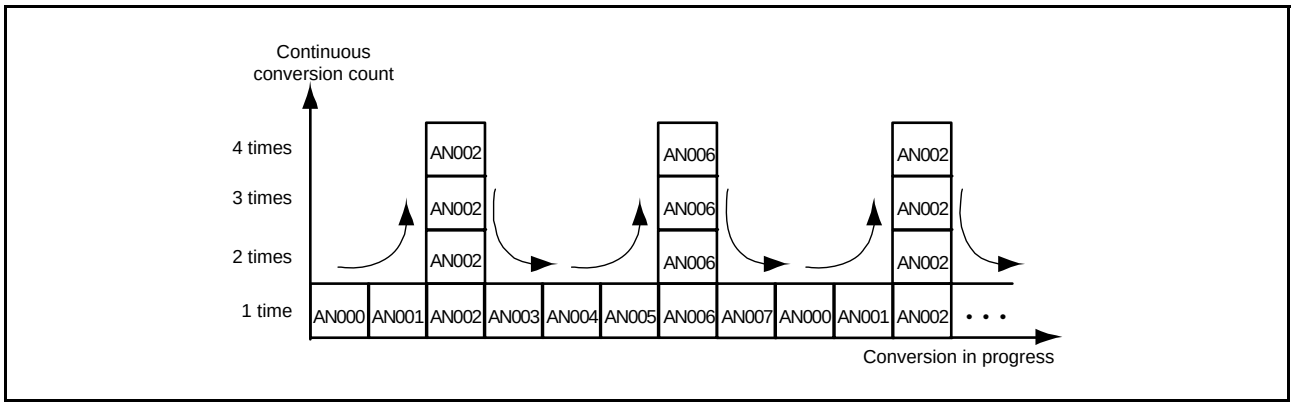


Figure 32.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS[2] = 1, and ADS[6] = 1

32.2.9 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.

32.2.10 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	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	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

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 32.2.1, A/D Data Registers y (ADDRy) (y = 0 to 15), section 32.2.2, A/D Data Duplication Register (ADDBLDR), section 32.2.3, A/D Internal Reference Voltage Data Register (ADOCDR), and section 32.2.4, A/D Self-Diagnosis Data Register (ADRD).

32.2.11 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 32.5 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 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 32.6 shows the A/D conversion startup sources selected by TRSA[3:0] bits

Table 32.5 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

Table 32.6 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

32.2.12 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 should be set to 0 in single scan mode. The OCS bit should be set while the ADST bit is 0.

32.2.13 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

	b7	b6	b5	b4	b3	b2	b1	b0
	SST[7:0]							
Value after reset:	0	0	0	1	0	1	0	0

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 32 MHz, one state is 31.25 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 32.7 shows the A/D sampling state registers and corresponding channels.

Table 32.7 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

32.2.14 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.
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.

32.3 Operation

32.3.1 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 or ELC triggers selected by TRSA[3:0] bits in ADSTRGR.

32.3.2 Single Scan Mode

32.3.2.1 Basic Operation

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 both 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 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.

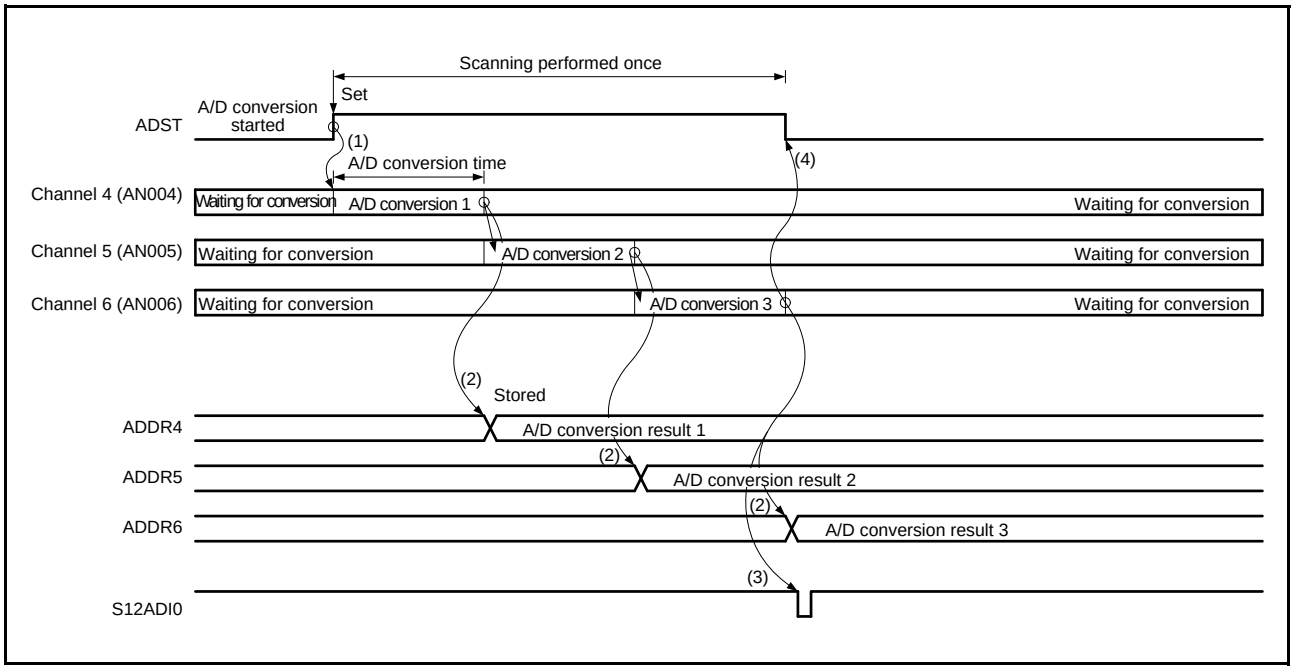


Figure 32.3 Example of Operation in Single Scan Mode (Basic Operation: AN004 to AN006 Selected)

32.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed for the self-diagnosis voltage (VREFH0 × 0, VREFH0 × 1/2, or VREFH0 × 1) generated from the reference power supply voltage (VREFH0) supplied to the 12-bit 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 both 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 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.

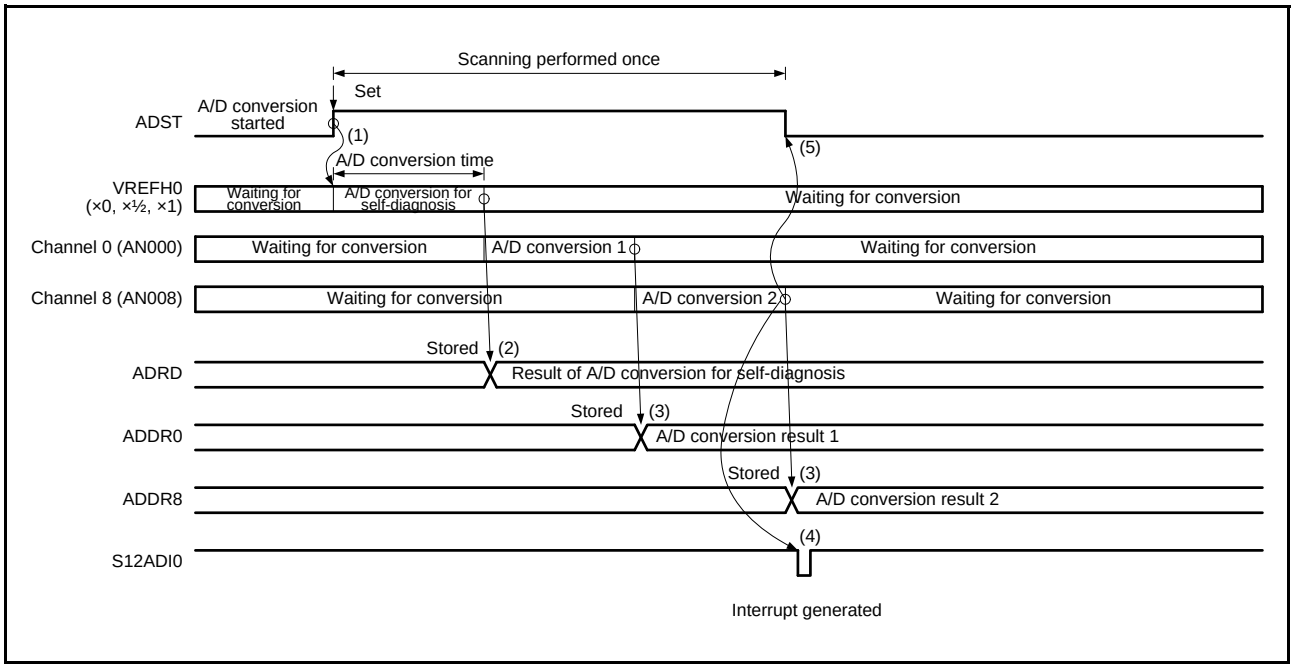


Figure 32.4 Example of Operation in Single Scan Mode (Basic Operation + Self-Diagnosis)

32.3.2.3 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 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.

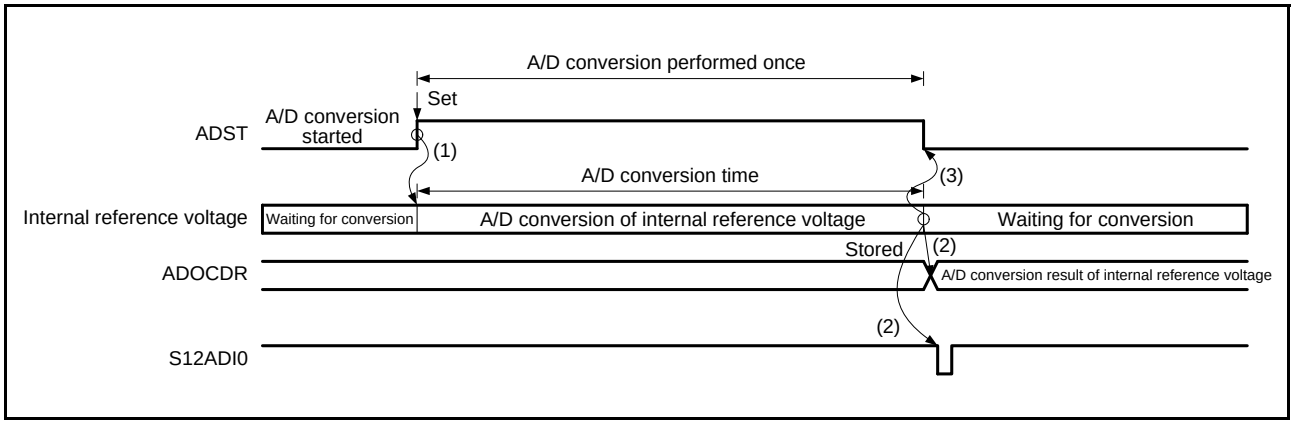


Figure 32.5 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)

32.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by the MTU 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 both 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 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 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.

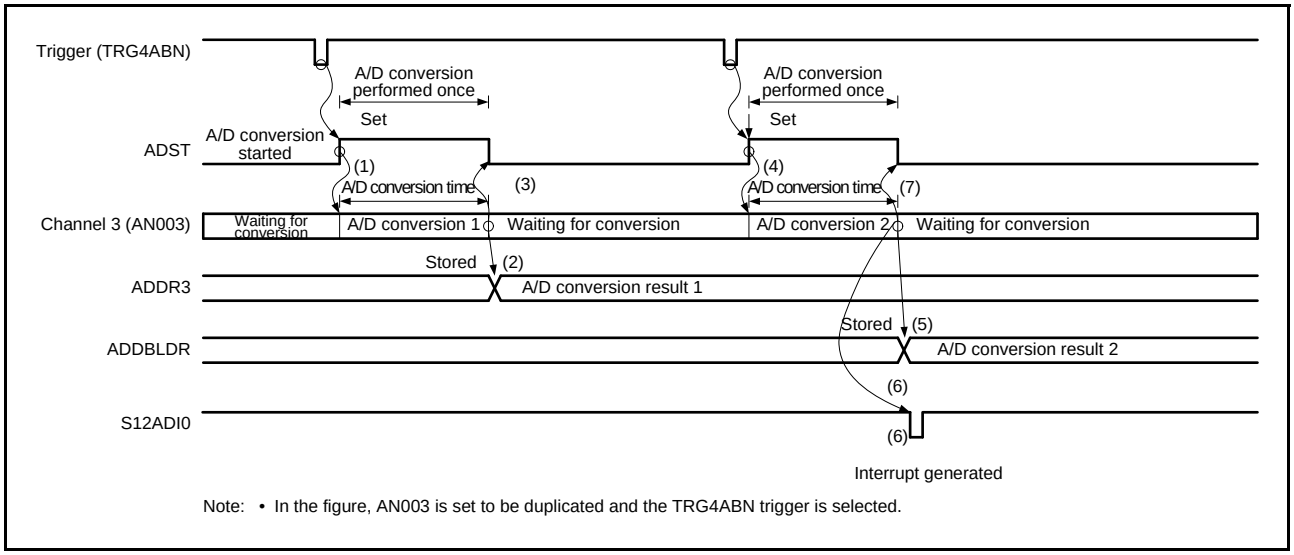


Figure 32.6 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

32.3.3 Continuous Scan Mode

32.3.3.1 Basic Operation

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 both 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 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.

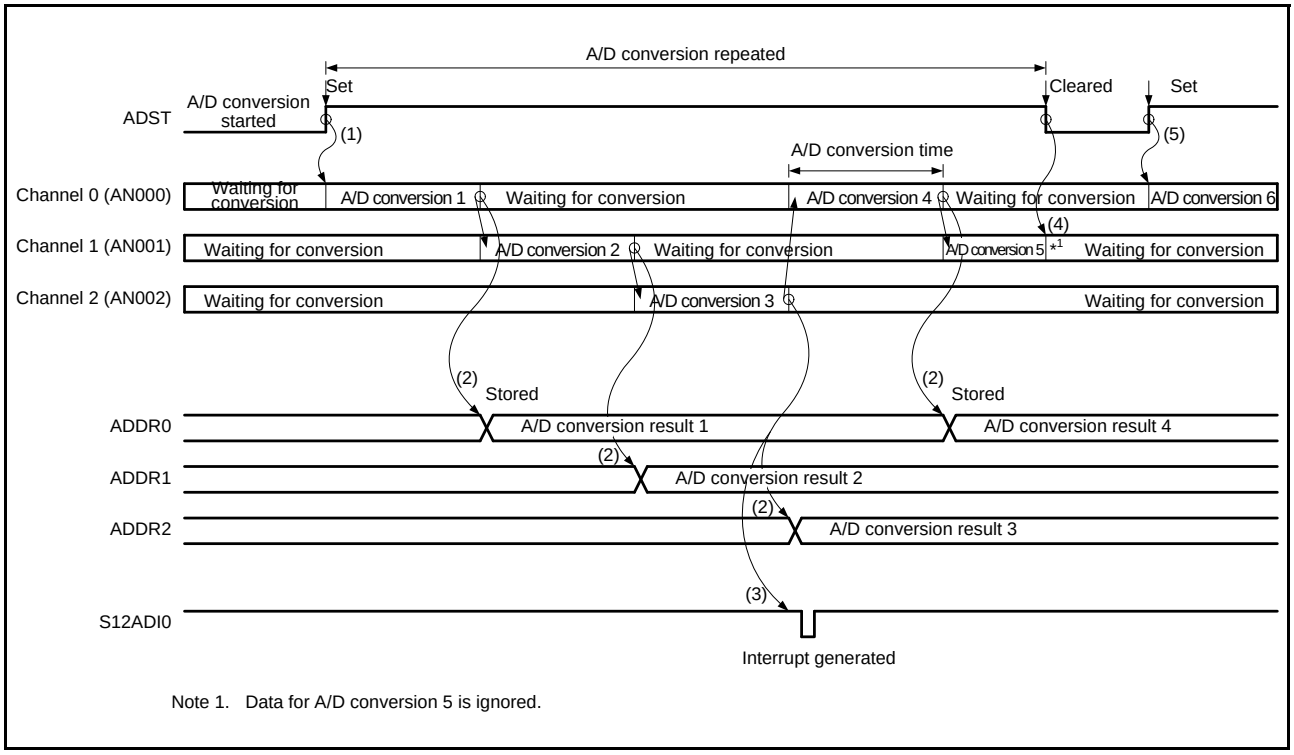


Figure 32.7 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

32.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed for the self-diagnosis voltage (VREFH0 × 0, VREFH0 × 1/2, or VREFH0 × 1) generated from the reference power supply voltage (VREFH0) supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, for which the sequence is repeated as below.

In continuous scan mode, the internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both 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 or ELC), or asynchronous trigger input.

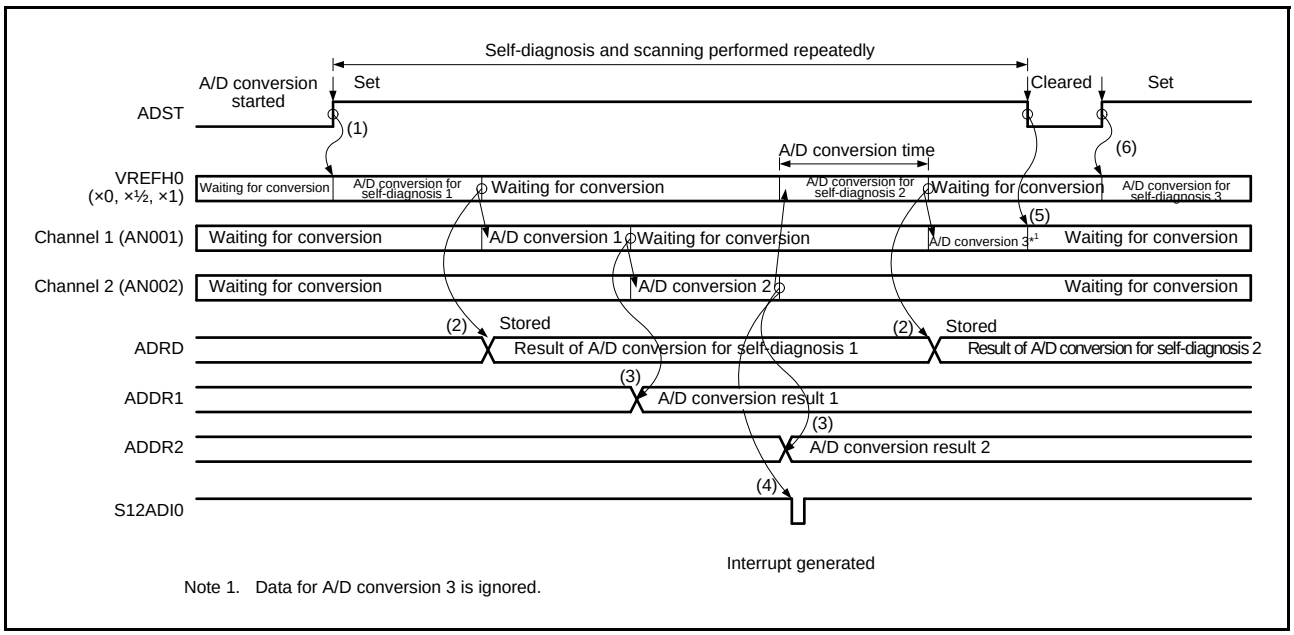
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(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 32.8 Example of Operation in Continuous Scan Mode (Basic Operation + Self-Diagnosis)

32.3.4 Group Scan Mode

32.3.4.1 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 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 both 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).

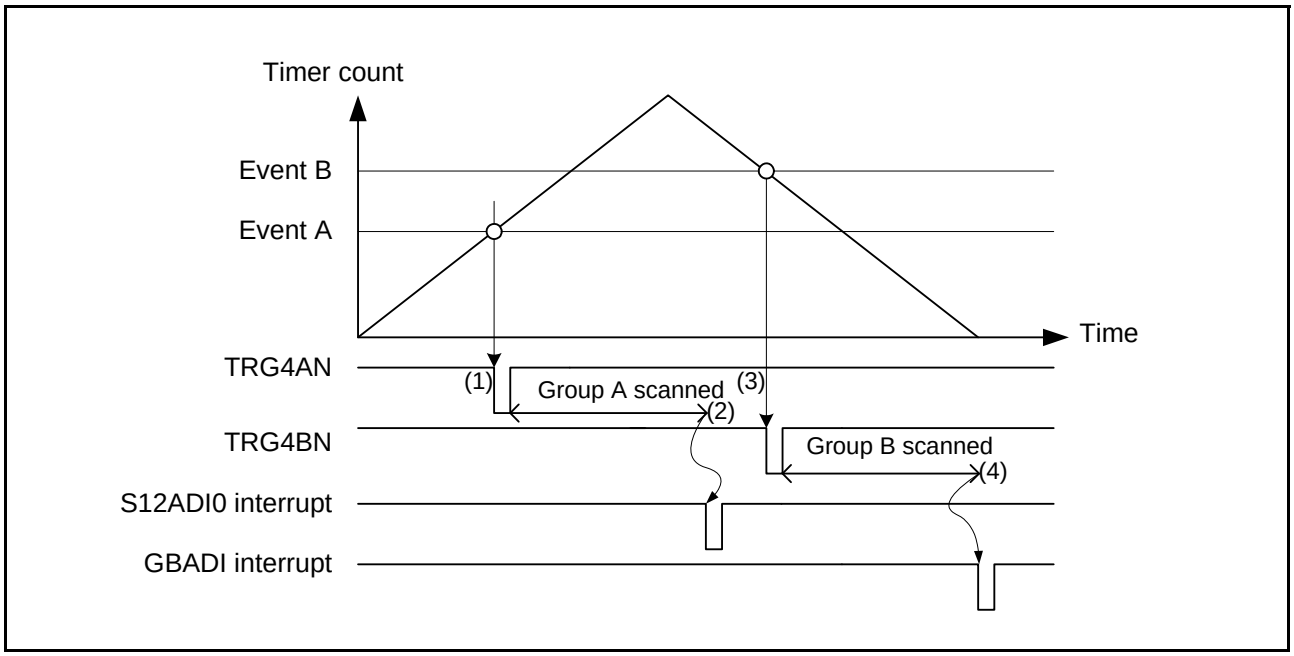


Figure 32.9 Example of Operation in Group Scan Mode (Basic Operation: MTU Triggers Used)

32.3.4.2 A/D Conversion in Double Trigger Mode

In group scan mode with double trigger mode, single scan operation started by the MTU or ELC trigger is performed twice for group A. For group B, single scan operation started by the MTU 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 both 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).

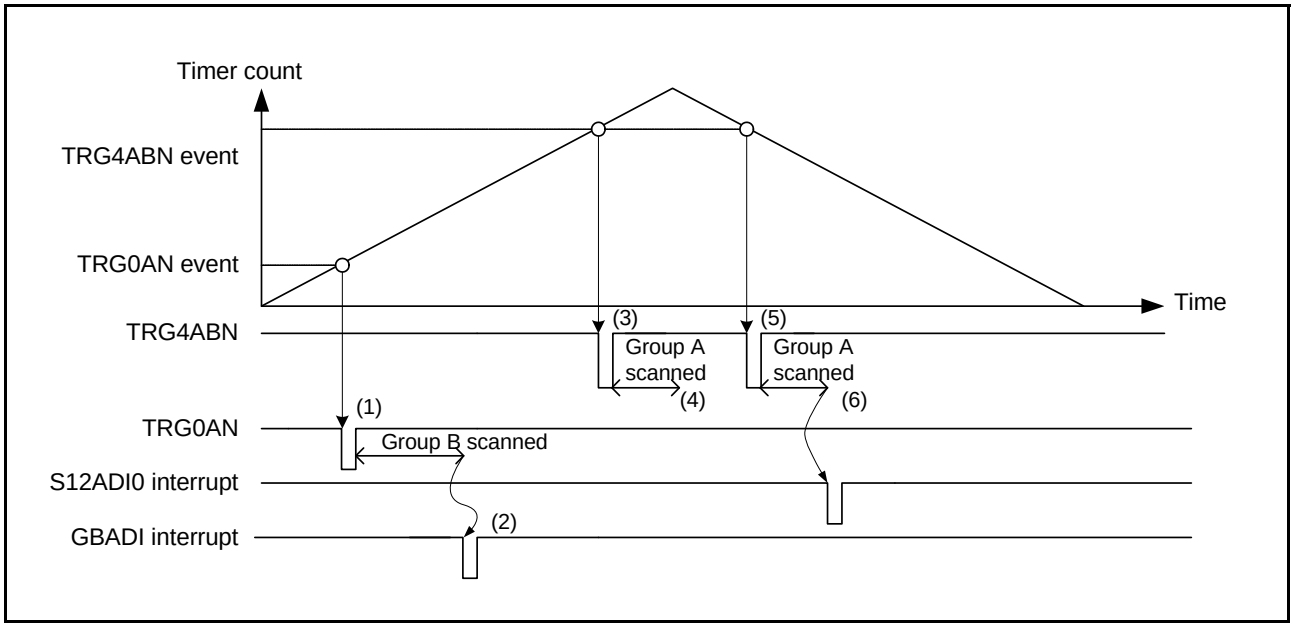


Figure 32.10 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: MTU Triggers Used)

32.3.5 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by software trigger; the triggers from the MTU, ELC, or ADTRG0# (external trigger). After start-of-scanning-delay time (t_D) has passed, the A/D converter samples, executes the disconnection detection assist process, the conversion process for self-diagnosis, and then starts the A/D conversion process.
Figure 32.11 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software trigger or triggers from the MTU or ELC. Figure 32.12 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), disconnection detection assist processing time (t_{DIS})*1, self-diagnosis A/D conversion processing time (t_{DIAG})*2, A/D conversion processing time (t_{CONV}), and end-of-scanning-delay time (t_{ED}). Table 32.8 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:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $(t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)$.
The disconnection detection assist processing time (t_{DIS}) is the value set in the ADNDIS[3:0] 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 ADSSTRn.SST[7:0] bits*3.

Note 1. When the disconnection detection assist function is not used, $t_{DIS} = 0$.
Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$.
Note 3. Registers in Table 32.7.

Table 32.8 Scan Conversion Time (in Terms of PCLK and ADCLK Cycles)

Item	Symbol	Conditions	Scan Conversion Time (Cycles)
Start-of-scanning-delay time*1	t_D	MTU, ELC, or software trigger	2 PCLK + 4 ADCLK
		External trigger	4 PCLK + 4 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
End-of-scanning-delay time*1	t_{ED}	—	1 PCLK + 3 ADCLK
Scan conversion time*2	t_{SCAN}	—	5 PCLK + (50n + 87) ADCLK

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, refer to Figure 32.11 and Figure 32.12.
Note 2. It is assumed that scan conversion is activated by the external trigger, 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.

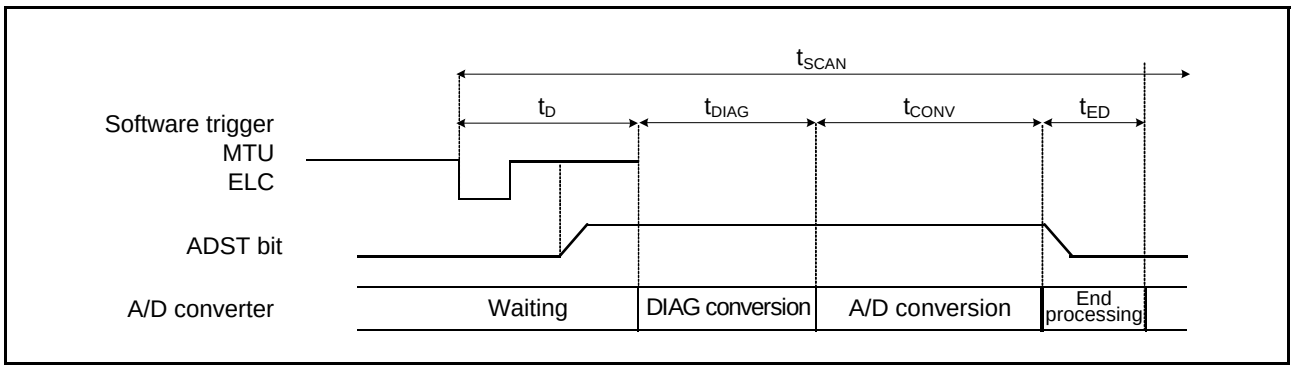


Figure 32.11 Scan Conversion Timing (Activated by Software, or Triggers from the MTU or ELC)

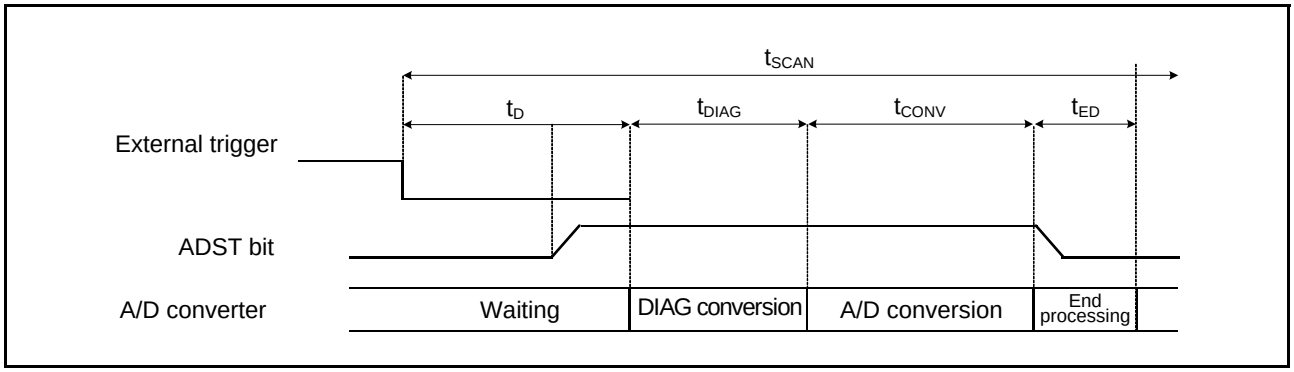


Figure 32.12 Scan Conversion Timing (Activated by ADTRG0#)

32.3.6 Usage Example of Automatic Register Clearing Function

Setting the ACE bit in ADCER to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADOCDR, and ADDBLDR) 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.

32.3.7 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.

32.3.8 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 32.13 shows the A/D conversion when the disconnection detection assist function is used. Figure 32.14 shows the example of disconnection detection at the VREFH0 (precharge is selected) and Figure 32.15 shows the example of disconnection detection at the VREFL0 side (discharge is selected).

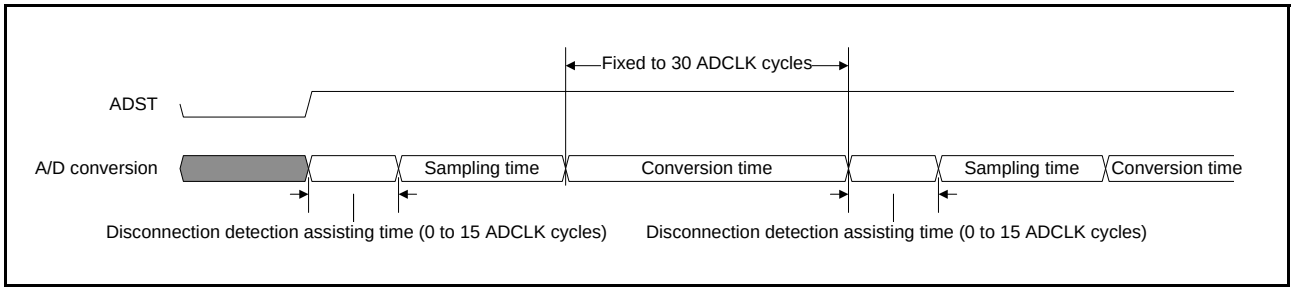
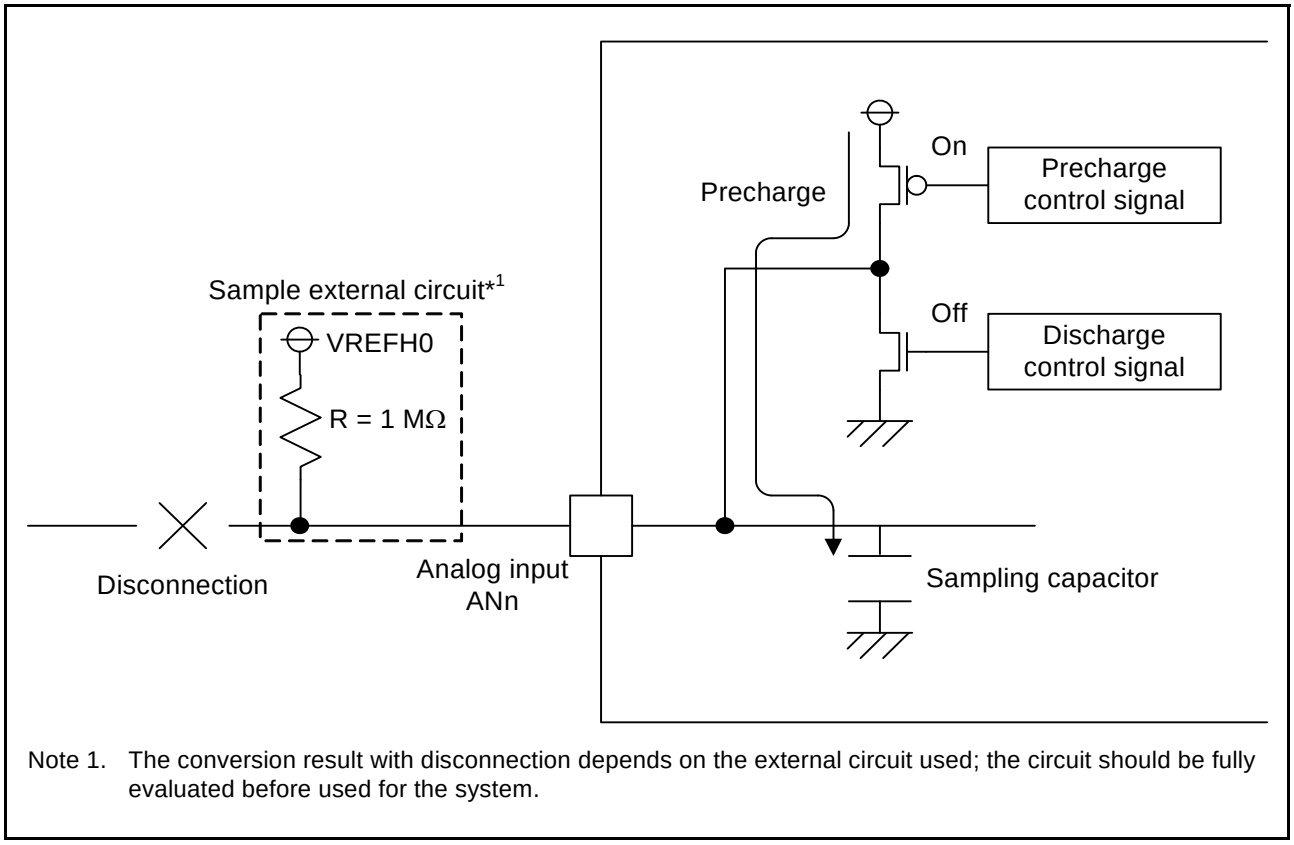
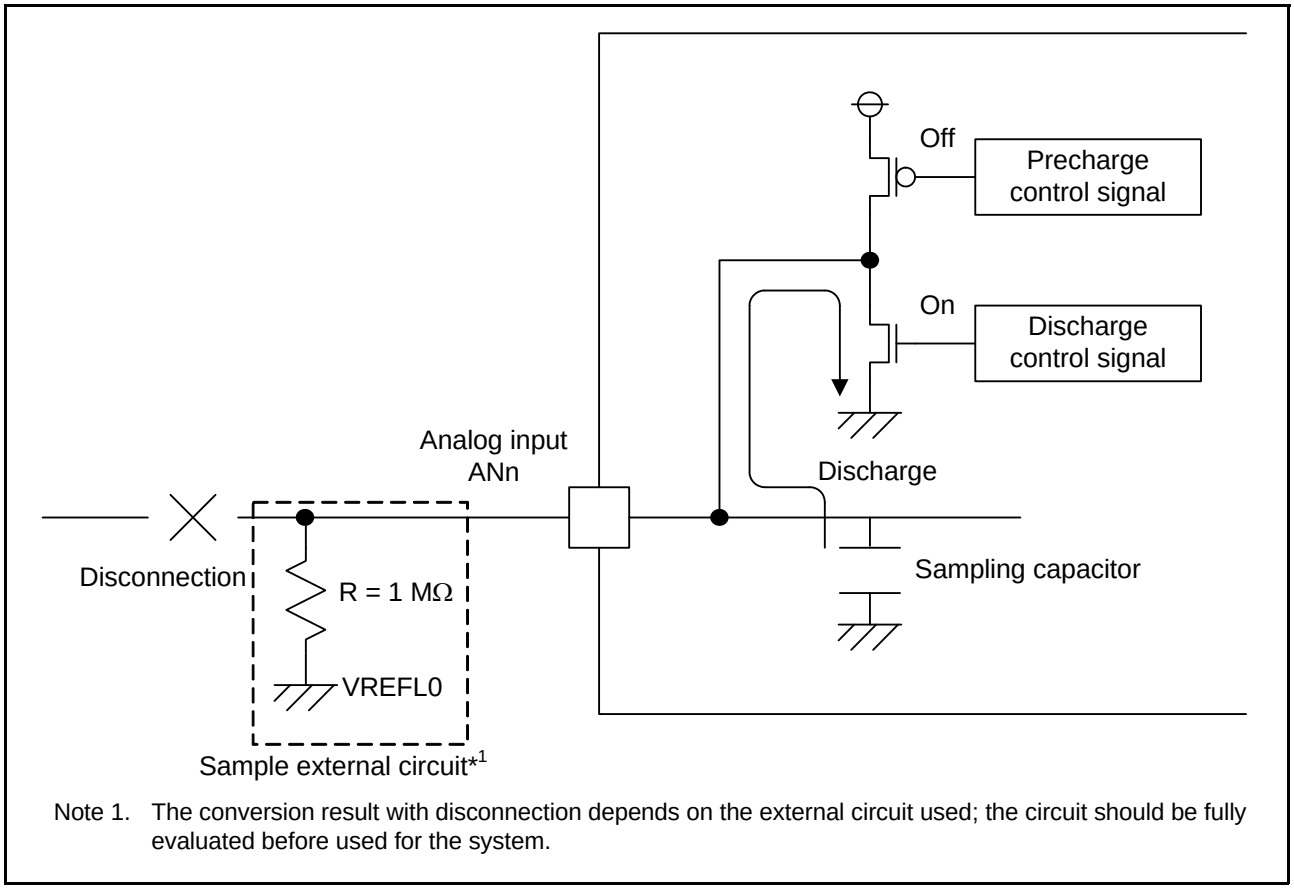


Figure 32.13 A/D Conversion with Disconnection Detection Assist Function Used



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 32.14 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 32.15 VREFL0 Disconnection Detection Example (Discharge Selected)

32.3.9 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 32.16 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 32.7.3, A/D Conversion Restarting Timing and Termination Timing.

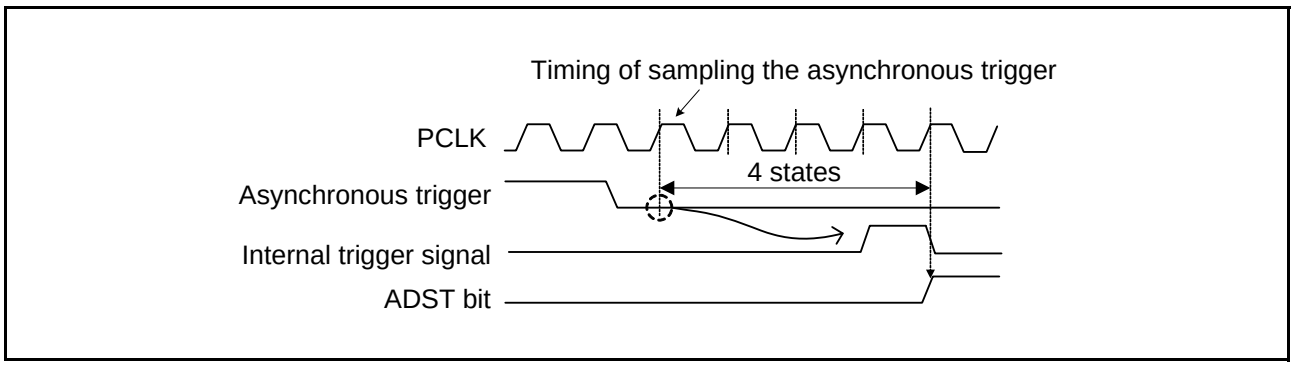


Figure 32.16 Asynchronous Trigger Input Timing

32.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

The A/D conversion can be started by a synchronous trigger of the MTU 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.

32.4 Interrupt Sources and DMA Transfer Requests

32.4.1 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 17, Data Transfer Controller (DTCa), and for details on DMAC settings, see section 16, DMA Controller (DMACA).

32.5 Event Linkage

32.5.1 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.

32.5.2 12-bit A/D Converter Operation by Event from ELC

The 12-bit A/D converter can be started by the predetermined event by so setting ELSRn of the ELC.

32.5.3 Notes on Event Reception from ELC during 12-bit A/D Conversion

When an event occurs during A/D conversion, it is invalid.

32.6 A/D Conversion Accuracy Definitions

The RX220 Group's A/D conversion accuracy is defined as below:

- Resolution
  The number of 12-bit A/D converter digital output codes
- Offset error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 000000000000 to 000000000001, excluding quantization error.
- Full-scale error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111111110 to 111111111111, excluding quantization error.
- Quantization error
  The deviation inherent in the 12-bit A/D converter, given by 1/2 LSB
- Nonlinearity error
  The error with respect to the ideal A/D conversion characteristic between zero voltage and the full-scale error, excluding offset error, full-scale error, and quantization error.
- Absolute accuracy
  The deviation between the digital value and analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

32.7 Usage Notes

32.7.1 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.

32.7.2 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).

32.7.3 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.

32.7.4 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.

32.7.5 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.
After the module stop state is canceled, wait for 1 μs to start A/D conversion. For details, see section 11, Low Power Consumption.

32.7.6 Notes on Entering Low Power Consumption States

Before entering module stop mode or software standby mode, make sure to stop A/D conversion. Here, set the ADST bit in ADCSR 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 the module stop state mode or software standby mode.

32.7.7 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.56 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 1.0 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 3.0 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 32.17. 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.

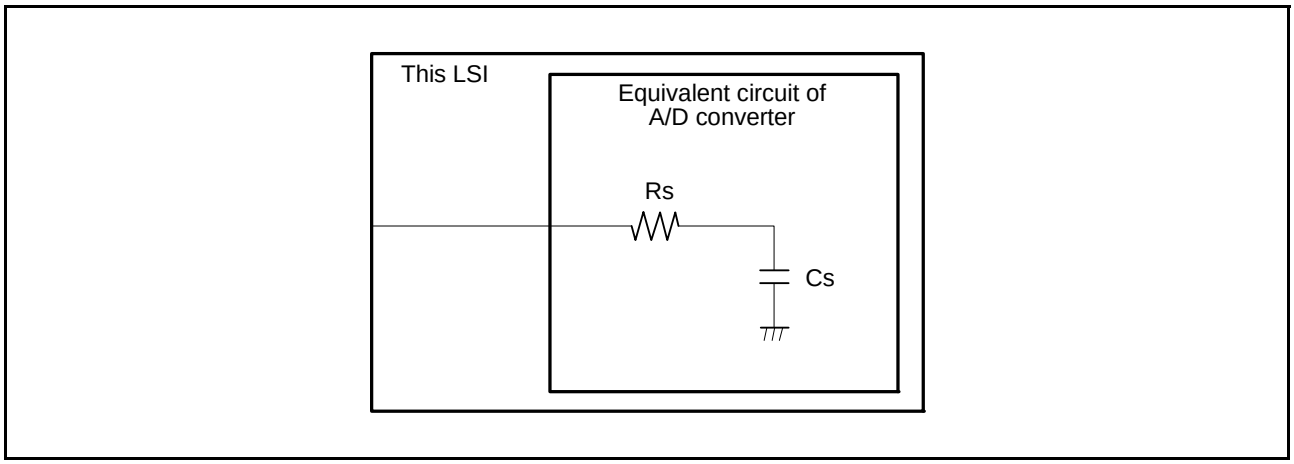


Figure 32.17 Internal Equivalent Circuit of Analog Input Pin

Table 32.9 Specifications of Analog Input Pins

Item		Min.	Max.	Unit
Allowable signal source impedance*1		—	1.0	kΩ
Internal equivalent circuit of a pin*2	Rs	—	6.0	kΩ
	Cs	—	16	pF

Note 1. Value to achieve high-speed conversion of 1.56 µs. The value differs depending on the analog power supply voltage and analog input pins to be used. For details, see section 38, Electrical Characteristics.
Note 2. Values when the voltage condition is AVCC0 ≥ 2.7 V.

32.7.8 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.

32.7.9 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 32.18, and connection should be made so that the following conditions are satisfied at the supply side.
  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
- VREFH0 range
  The reference voltage range applied to the VREFH0 pin should be VREFH0 ≤ AVCC0.

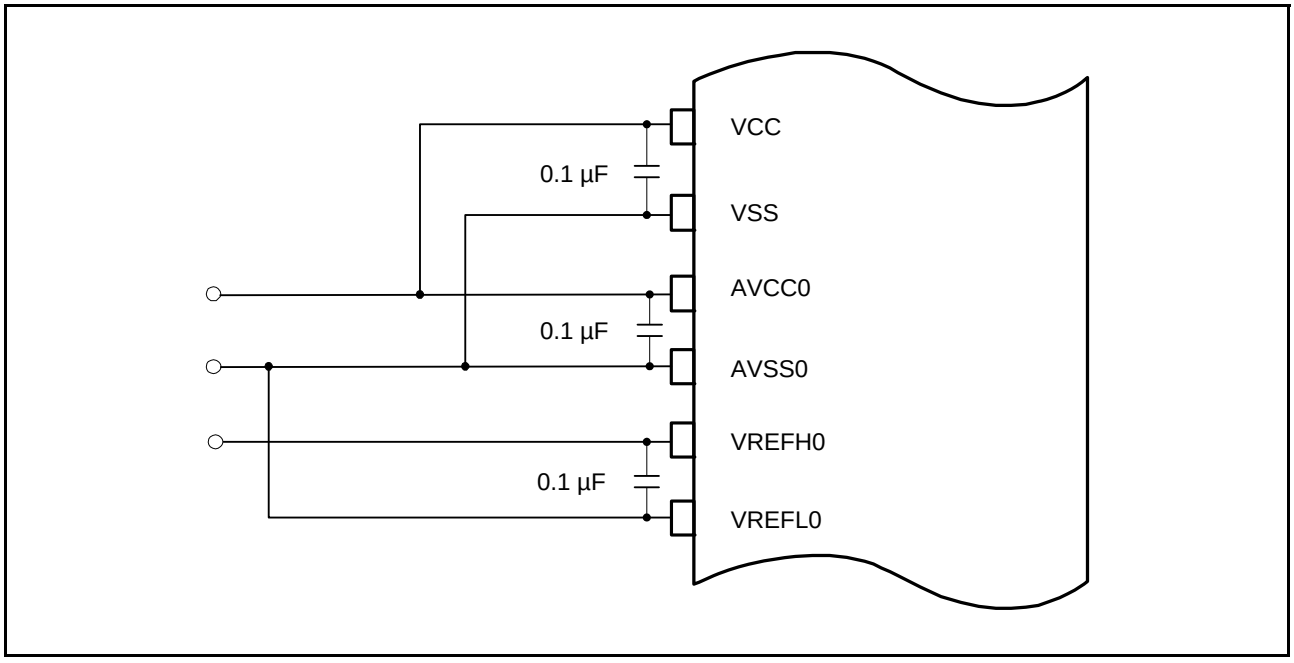


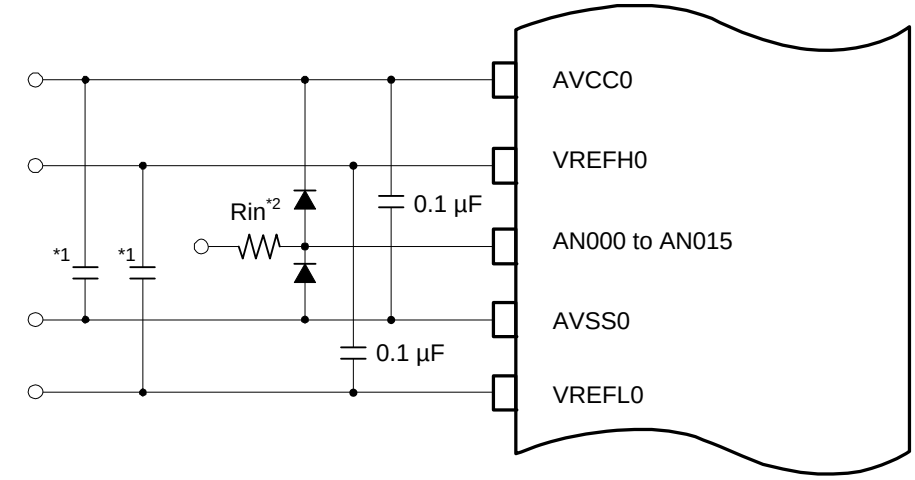
Figure 32.18 Power Supply Pin Connection Example

32.7.10 Notes on Board Design

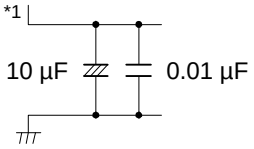
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN015), reference power supply pin (VREFH0), reference ground pin (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).

32.7.11 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 32.19.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 32.19 Sample Protection Circuit for Analog Inputs

32.7.12 Port Setting when 12-bit A/D Converter Inputs are Used

If any one of port 4 or port E pins is used as an analog input pin for the 12-bit A/D converter, none of port 0 and port 4 output pins should be used. This is because an analog power supply is used for parts of the ports 0 and 4 circuits.

32.7.13 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

33. Comparator A

Comparator A compares a reference input voltage and an analog input voltage. Comparator A1 and comparator A2 are independent of each other. Note that these comparators A1 and A2 share the voltage detection circuit with voltage monitor 1 and voltage monitor 2. Either “comparator A1 and comparator A2” or “voltage monitor 1 and voltage monitor 2” can be selected to use the voltage detection circuit.

33.1 Overview

The comparison result of the reference input voltage and analog input voltage can be read by software. An input voltage to the CVREFA pin can be selected as the reference input voltage. Also, the comparator A1 interrupt and comparator A2 interrupt can be used.
Table 33.1 lists the comparator A specifications, Figure 33.1 shows a block diagram of comparator A, and Table 33.2 shows the pin configuration of comparator A.

Table 33.1 Comparator A Specifications

Item		Comparator A1	Comparator A2
Analog input voltage		Input voltage to the CMPA1 pin	Input voltage to the CMPA2 pin
Reference input voltage		Input voltage to the CVREFA pin	
Comparison target		Compares whether the analog input voltage has passed through the reference input voltage by rising or falling.	
Comparison result monitor		LVD1MON bit in the LVD1SR register	LVD2MON bit in the LVD2SR register
		Indicates whether the analog input voltage is higher or lower than the reference input voltage	
Interrupt		Comparator A1 interrupt (non-maskable or maskable can be selected)	Comparator A2 interrupt (non-maskable or maskable can be selected)
Interrupt request generation timing		Input voltage to the CMPA1 pin has risen above or fallen below the CVREFA pin reference input voltage, or either of the two.	Input voltage to the CMPA2 pin has risen above or fallen below the CVREFA pin reference input voltage, or either of the two.
Event generation timing to ELC		Input voltage to the CMPA1 pin has risen above or fallen below the CVREFA pin reference input voltage, or either of the two.	—
Digital filter	Switching enable/ disable	Supported	
	Sampling time	(LOCO divided by n) × 2 n: 1, 2, 4, 8	
Comparison result output		Comparison result can be output from a port by going through the event link controller (ELC).	—

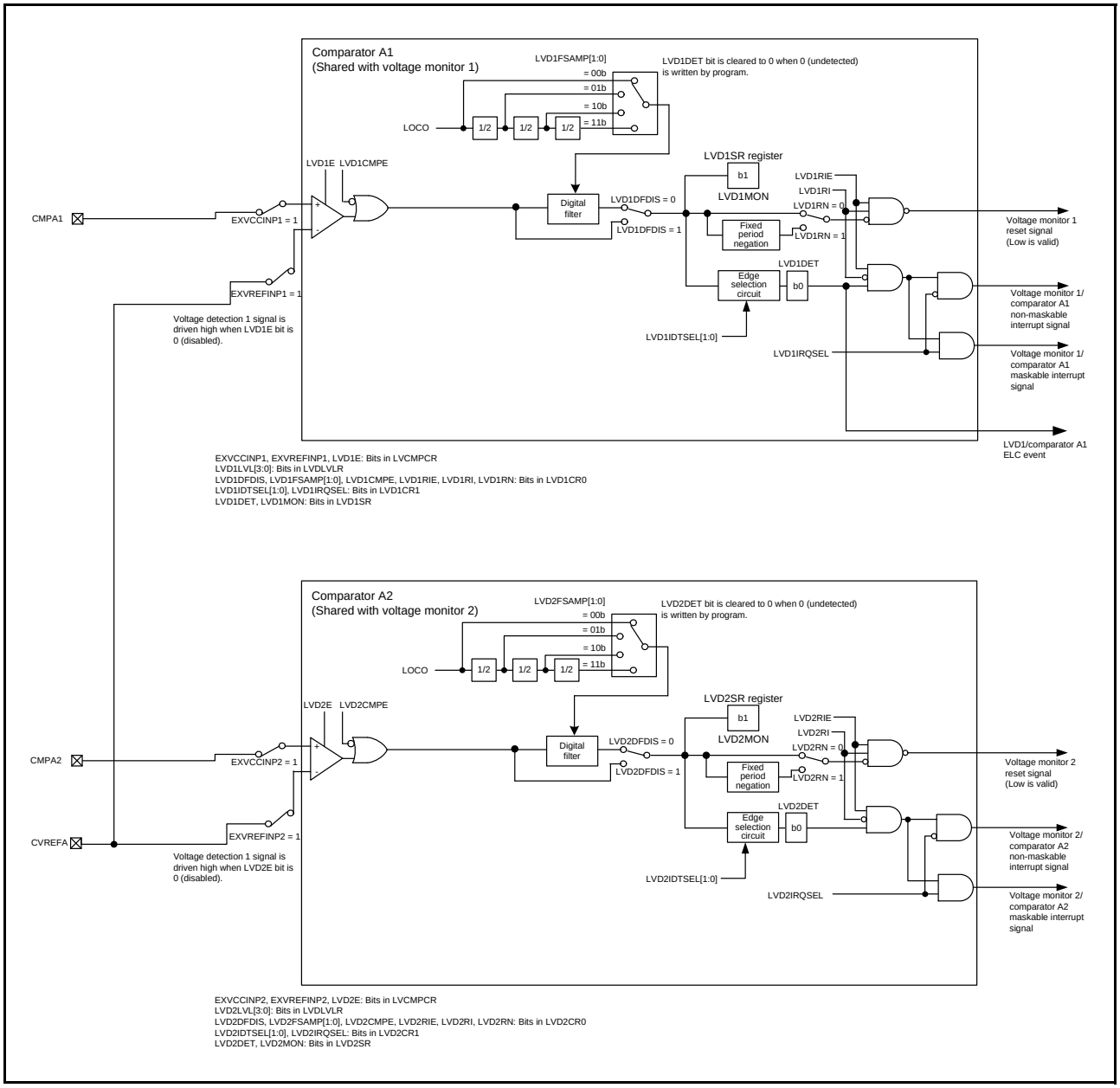


Figure 33.1 Block Diagram of Comparator A

Table 33.2 Pin Configuration of Comparator A

Pin Name	I/O	Function
CMPA1	Input	Comparator A1 analog pin
CMPA2	Input	Comparator A2 analog pin
CVREFA	Input	Comparator reference voltage pin

33.2 Register Descriptions

33.2.1 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 1 (LVD1CR1)

Address: 0008 00E0h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL[1:0]	Voltage Monitoring 1/Comparator A1 Interrupt/ELC Event Generation Condition Select	b1 b0 0 0: CMPA1 ≥ CVREFA 0 1: CMPA1 < CVREFA 1 0: Falling below or rising above is detected. 1 1: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1/Comparator A1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	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.

33.2.2 Voltage Monitoring 1 Circuit/Comparator A1 Status Register (LVD1SR)

Address: 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/Comparator A1 Voltage Change Detection Flag	0: Not detected 1: Comparator A1 comparison result change has been detected.	R/W *1
b1	LVD1MON	Voltage Monitoring 1/Comparator A1 Signal Monitor Flag	0: CMPA1 < CVREFA 1: CMPA1 ≥ CVREFA	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 two system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1/Comparator A1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again after a period of PCLKB2 cycle or more has elapsed.
A wait time of two or more PCLKB cycles can be secured by reading an I/O register of which access cycles is specified with PCLKB.

LVD1MON Flag (Voltage Monitoring 1/Comparator A1 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).

33.2.3 Voltage Monitoring 2 Circuit/Comparator A2 Control Register 1 (LVD2CR1)

Address: 0008 00E2h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2/Comparator A2 Interrupt Generation Condition Select	b1 b0 0 0: CMPA2 ≥ CVREFA 0 1: CMPA2 < CVREFA 1 0: Falling below or rising above is detected. 1 1: Setting prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2/ Comparator A2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	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.

33.2.4 Voltage Monitoring 2 Circuit/Comparator A2 Status Register (LVD2SR)

Address: 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/Comparator A2 Voltage Change Detection Flag	0: Not detected 1: Comparator A2 comparison result change has been detected.	R/W *1
b1	LVD2MON	Voltage Monitoring 2/Comparator A2 Signal Monitor Flag	0: CMPA2 < CVREFA 1: CMPA2 ≥ CVREFA	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 two system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2/Comparator A2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again after a period of PCLKB2 cycle or more has elapsed.
A wait time of two or more PCLKB cycles can be secured by reading an I/O register of which access cycles is specified with PCLKB.

LVD2MON Flag (Voltage Monitoring 2/Comparator A2 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).

33.2.5 Voltage Monitoring Circuit/Comparator A Control Register (LVCMPCR)

Address: 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	EXVCCINP2	EXVREFINP2	EXVCCINP1	EXVREFINP1
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXVREFINP1	Comparator A1 Reference Voltage External Input Select	1: CVREFA pin input voltage Set this bit to 1 when using comparator A1.	R/W
b1	EXVCCINP1	Comparator A1 Comparison Voltage External Input Select	1: CMPA1 pin input voltage Set this bit to 1 when using comparator A1.	R/W
b2	EXVREFINP2	Comparator A2 Reference Voltage External Input Select	1: CVREFA pin input voltage Set this bit to 1 when using comparator A2.	R/W
b3	EXVCCINP2	Comparator A2 Comparison Voltage External Input Select	1: CMPA2 pin input voltage Set this bit to 1 when using comparator A2.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1/ Comparator A1 Enable	0: Voltage detection 1 and comparator A1 circuit are disabled. 1: Voltage detection 1 and comparator A1 circuit are enabled.	R/W
b6	LVD2E	Voltage Detection 2/ Comparator A2 Enable	0: Voltage detection 2 and comparator A2 circuit are disabled. 1: Voltage detection 2 and comparator A2 circuit are 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.

The EXVREFINP1, EXVCCINP1, EXVREFINP2, and EXVCCINP2 bits can only be modified when the LVD1E and LVD2E bits are 0 (voltage detection circuit disabled)

LVD1E Bit (Voltage Detection 1/Comparator A1 Enable)

When using voltage detection 1/comparator A1 interrupt/reset or the LVD1SR.LVD1MON bit, set the LVD1E bit to 1. The voltage detection 1/comparator A1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

LVD2E Bit (Voltage Detection 2/Comparator A2 Enable)

When using voltage detection 2/comparator A2 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.

33.2.6 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 0 (LVD1CR0)

Address: 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/ Comparator A1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD1DFDIS	Voltage Monitoring 1/ Comparator A1 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit/ Comparator A1 Comparison Result Output Enable	0: Comparator A1 circuit comparison result output is disabled. 1: Comparator A1 circuit comparison result output is 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: LOCO divided by 1 0 1: LOCO divided by 2 1 0: LOCO divided by 4 1 1: LOCO divided by 8	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit/ Comparator A1 Mode Select	0: The comparator A1 interrupt is generated when CMPA1 has crossed the CVREFA 1: The comparator A1 reset enabled when the CMPA1 has crossed the CVREFA	R/W
b7	LVD1RN	Voltage Monitoring 1/ Comparator A1 Reset Negation Select	0: Negation follows a stabilization time (tLVD1) after CMPA1 > CVREFA 1: Negation follows a stabilization time (tLVD1) after assertion of the comparator A1 reset.	R/W

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

LVD1RIE Bit (Voltage Monitoring 1/Comparator A1 Interrupt/Reset Enable)

The voltage monitoring 1 reset or voltage monitoring 1 non-maskable interrupt should not be generated during flash memory programming/erasure.

LVD1DFDIS Bit (Voltage Monitoring 1/Comparator A1 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is to be set to 0 (digital filter circuit enabled).

Set the LVD1DFDIS bit to 1 (digital filter circuit disabled) when using a voltage monitoring 1 circuit in software standby mode.

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

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

LVD1RN Bit (Voltage Monitoring 1/Comparator A1 Reset Negation Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby is to be made, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after CMPA1 > CVREFA is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset signal).

33.2.7 Voltage Monitoring 2 Circuit/Comparator A2 Control Register 0 (LVD2CR0)

Address: 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/ Comparator A2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD2DFDIS	Voltage Monitoring 2/ Comparator A2 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit/ Comparator A2 Comparison Result Output Enable	0: Comparator A2 circuit comparison result output is disabled. 1: Comparator A2 circuit comparison result output is 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: LOCO divided by 1 0 1: LOCO divided by 2 1 0: LOCO divided by 4 1 1: LOCO divided by 8	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit/ Comparator A2 Mode Select	0: The comparator A2 interrupt is generated when CMPA2 has crossed the CVREFA 1: The comparator A2 reset enabled when the CMPA2 has crossed the CVREFA	R/W
b7	LVD2RN	Voltage Monitoring 2/Comparator A2 Reset Negation Select	0: Negation follows a stabilization time (tLVD2) after CMPA2 > CVREFA 1: Negation follows a stabilization time (tLVD2) after assertion of the comparator A2 reset.	R/W

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

LVD2RIE Bit (Voltage Monitoring 2/Comparator A2 Interrupt/Reset Enable)

The voltage monitoring 2 reset or voltage monitoring 2 non-maskable interrupt should not be generated during flash memory programming/erasure.

LVD2DFDIS Bit (Voltage Monitoring 2/Comparator A2 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD2DFDIS bit is to be set to 0 (digital filter circuit enabled).

Set the LVD2DFDIS bit to 1 (digital filter circuit disabled) when using a voltage monitoring 2 circuit in software standby mode.

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

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

LVD2RN Bit (Voltage Monitoring 2/Comparator A2 Reset Negation Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby is to be made, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after CMPA2 > CVREFA is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset signal).

33.3 Monitoring Comparison Results

33.3.1 Monitoring Comparator A1

Table 33.3 lists the procedures for setting up monitoring against comparator A1. After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by comparator A1.

Table 33.3 Procedures for Setting up Monitoring against comparator A1

Step	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Set the LVCMPCR.EXVREFINP1 bit to 1 (CVREFA pin input voltage). Set the LVCMPCR.EXVCCINP1 bit to 1 (CMPA1 pin input voltage).	
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 LVCMPCR.LVD1E bit to 1 (enabling the circuit for comparator A1).	
4	Wait for at least td(E-A) or longer.	
5	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by the comparator A1 circuit).	
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).	— (No waiting is required.)

33.3.2 Monitoring Comparator A2

Table 33.4 lists the procedures for setting up monitoring against comparator A2. After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by comparator A2.

Table 33.4 Procedures for Setting up Monitoring against comparator A2

Step	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Set the LVCMPCR.EXVREFINP2 bit to 1 (CVREFA pin input voltage). Set the LVCMPCR.EXVCCINP2 bit to 1 (CMPA2 pin input voltage).	
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 LVCMPCR.LVD2E bit to 1 (enabling the circuit for comparator A2).	
4	Wait for at least td(E-A) or longer.	
5	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by the comparator A2 circuit).	
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).	— (No waiting is required.)

33.4 Operation

Comparator A1 and comparator A2 operate independently.
The comparison result of the reference input voltage and analog input voltage can be read by software. An input voltage to the CVREFA pin can be used as the reference input voltage. The comparator A1 interrupt and the comparator A2 interrupt can be used, and non-maskable or maskable can be selected for each interrupt type.

33.4.1 Comparator A1

Table 33.5 shows the procedure for setting bits related to the comparator A1 interrupt/ELC, Table 33.6 shows the procedures for stopping bits related to the comparator A1 interrupt/ELC, and Figure 33.2 shows an operating example of comparator A1.

Table 33.5 Procedures for Setting Bits Related to the Comparator A1 Interrupt/ELC

Step No.	When Using Digital Filter	When Using No Digital Filter
1 *2	Set the LVCMPCR.EXVREFINP1 bit to 1 (CVREFA pin input voltage). Set the LVCMPCR.EXVCCINP1 bit to 1 (CMPA1 pin input voltage).	
2 *1	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.	—
3 *1, *2	Set the LVD1CR0.LVD1RI bit to 0 (comparator A1 interrupt).	
4	Set the LVD1CR1.LVD1IDTSEL[1:0] bits to select the interrupt request timing. Set the LVD1CR1.LVD1IRQSEL bit to select the interrupt type.	
5 *2	Set the LVCMPCR.LVD1E bit to 1 (enabling the circuit for comparator A1).	
6 *2	Wait for at least td(E-A) or longer.	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by the comparator A1 circuit).	
8	Wait for at least one cycle of the LOCO.	—
9	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).	—
10	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).	—
11	Clear the LVD1SR.LVD1DET bit to 0.	
12	Set the LVD1CR0.LVD1RIE bit to 1 (enable comparator A1 interrupts). Events to the ELC are always output regardless of the setting of this bit.	

Note 1. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.
Note 2. Steps 1, 3, 5, and 6 are not required if operation is with the setting to select the comparator A1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR0.LVD1DFDIS and LVD1FSAMP bits or LVD1CR1.LVD1IRQSEL and LVD1IDTSEL bits after operation is stopped or if restarting is in a case where the settings related to the comparator A1 circuit were not changed after operation was stopped.

Table 33.6 Procedures for Stopping Bits Related to the Comparator A1 Interrupt/ELC

Step No.	
1	Set the LVD1CR0.LVD1RIE bit to 0 (disable comparator A1 interrupts).
2	Clear the LVD1CR0.LVD1CMPE bit to 0 (comparator A1 circuit comparison result output is disabled).
3 *1	Clear the LVCMPCR.LVD1E bit to 0 (comparator A1 circuit disabled).
4	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE.

Note 1. Step 3 is not required if operation is with the setting to select the comparator A1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR0.LVD1DFDIS and LVD1FSAMP bits or LVD1CR1.LVD1IRQSEL and LVD1IDTSEL bits after operation is stopped or if restarting is in a case where the settings related to the comparator A1 circuit were not changed after operation was stopped.

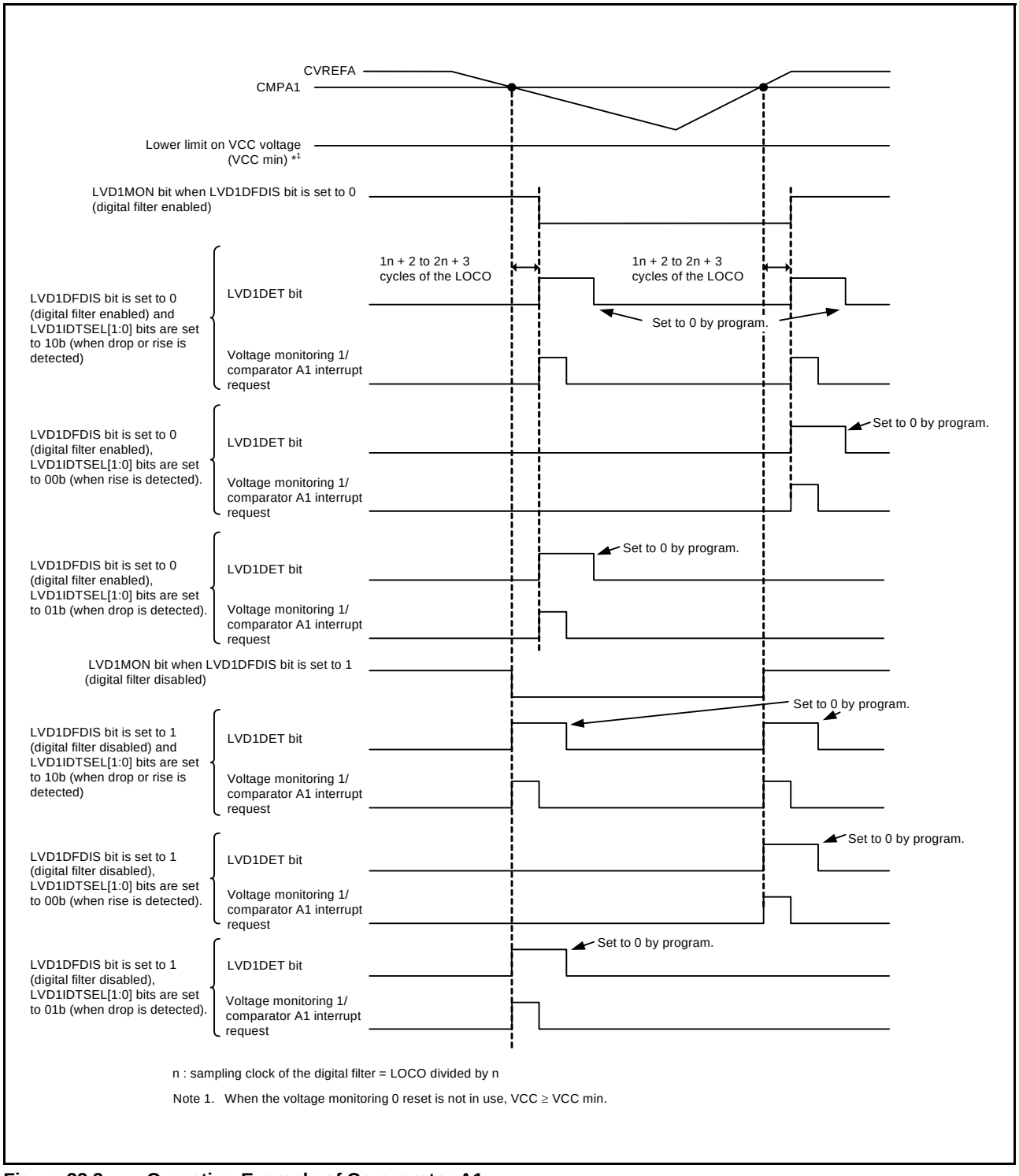


Figure 33.2 Operating Example of Comparator A1

33.4.2 Comparator A2

Table 33.7 shows the procedure for setting bits related to the comparator A2 interrupt, Table 33.8 shows the procedures for stopping bits related to the comparator A2 interrupt, and Figure 33.3 shows an operating example of comparator A2.

Table 33.7 Procedures for Setting Bits Related to the Comparator A2 Interrupt

Step No.	When Using Digital Filter	When Using No Digital Filter
1 *2	Set the LVCMPCR.EXVREFINP2 bit to 1 (CVREFA pin input voltage). Set the LVCMPCR.EXVCCINP2 bit to 1 (CMPA2 pin input voltage).	
2 *1	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.	—
3 *1, *2	Set the LVD2CR0.LVD2RI bit to 0 (comparator A2 interrupt).	
4	Set the LVD2CR1.LVD2IDTSEL[1:0] bits to select the interrupt request timing. Set the LVD2CR1.LVD2IRQSEL bit to select the interrupt type.	
5 *2	Set the LVCMPCR.LVD2E bit to 1 (enabling the circuit for comparator A2).	
6 *2	Wait for at least td(E-A) or longer.	
7	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by the comparator A2 circuit).	
8	Wait for at least one cycle of the LOCO.	—
9	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).	—
10	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).	—
11	Clear the LVD2SR.LVD2DET bit to 0.	
12	Set the LVD1CR0.LVD1RIE bit to 1 (enable comparator A2 interrupts).	

Note 1. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.
Note 2. Steps 1, 3, 5, and 6 are not required if operation is with the setting to select the comparator A2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR0.LVD2DFDIS and LVD2FSAMP bits or LVD2CR1.LVD2IRQSEL and LVD2IDTSEL bits after operation is stopped or if restarting is in a case where the settings related to the comparator A2 circuit were not changed after operation was stopped.

Table 33.8 Procedures for Stopping Bits Related to the Comparator A2 Interrupt

Step No.	
1	Set the LVD2CR0.LVD2RIE bit to 0 (disable comparator A2 interrupts).
2	Clear the LVD2CR0.LVD2CMPE bit to 0 (comparator A2 circuit comparison result output is disabled).
3 *1	Clear the LVCMPCR.LVD2E bit to 0 (comparator A2 circuit disabled).
4	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD2E, LVD2CR0.LVD2RIE, and LVD2CR0.LVD2CMPE.

Note 1. Step 3 is not required if operation is with the setting to select the comparator A2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR0.LVD2DFDIS and LVD2FSAMP bits or LVD2CR1.LVD2IRQSEL and LVD2IDTSEL bits after operation is stopped or if restarting is in a case where the settings related to the comparator A2 circuit were not changed after operation was stopped.

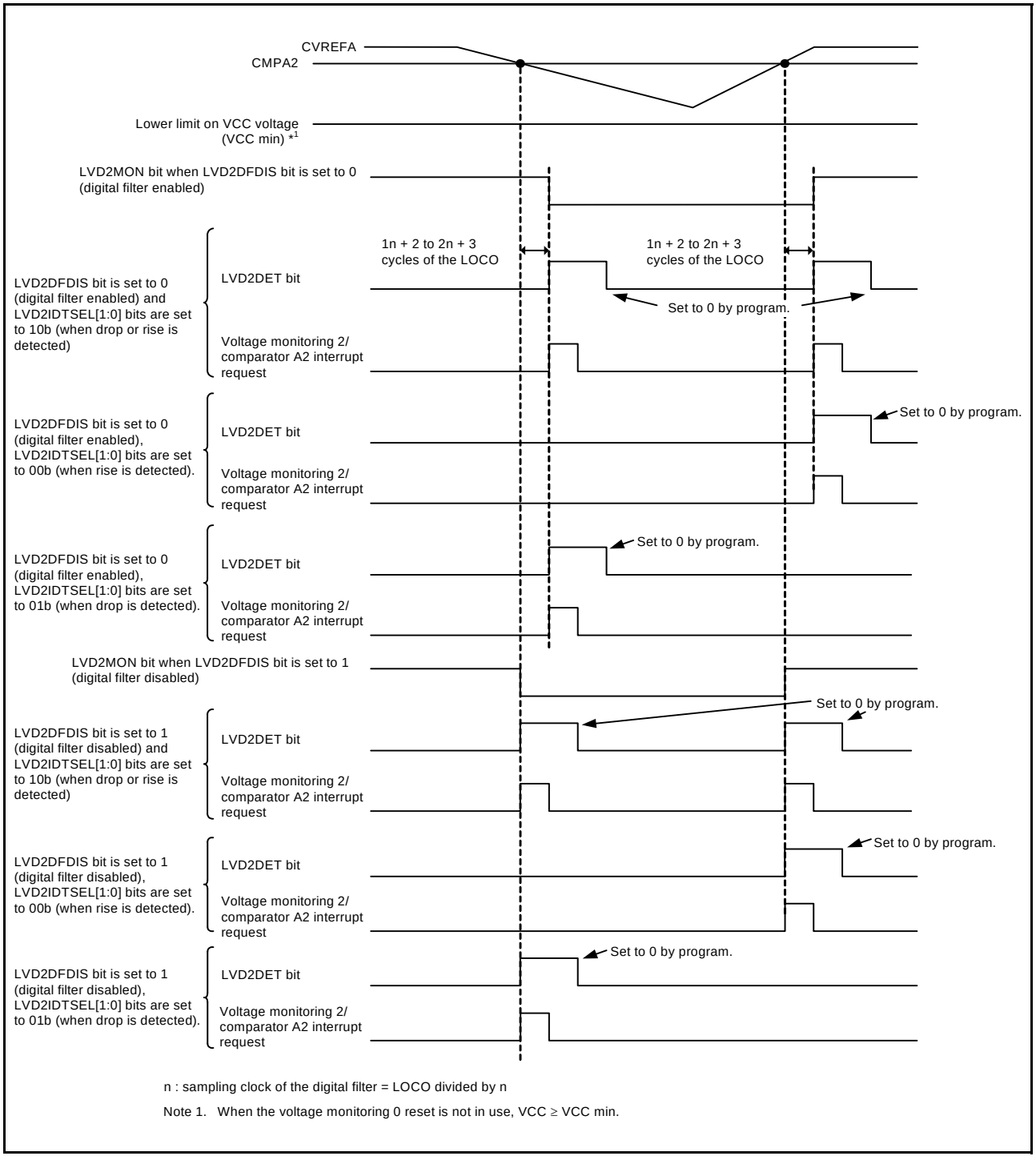


Figure 33.3 Operating Example of Comparator A2

33.5 Comparator A1 and Comparator A2 Interrupts

The two interrupt requests (from comparators A1 and A2) are generated with the following timing.

(1) When the input voltage on the CMPA1 pin has crossed the reference voltage on the CVREFA pin

(2) When the input voltage on the CMPA2 pin has crossed the reference voltage on the CVREFA pin

The LVDiCR1.LVDiIDTSEL[1:0] bits are used to select interrupt request generating conditions: rising above, falling below, or either of the two.

Non-maskable or maskable can be selected for each interrupt type.

For interrupts, refer to section 14, Interrupt Controller (ICUb).

33.5.1 Non-maskable Interrupts

When the LVDiCR1.LVDiIRQSEL bit is 0, the comparator Ai interrupt functions as a non-maskable interrupt. When the selected interrupt request timing occurs, the LVDiSR.LVDiMON bit is set to 1. At this time, if the NMIER.LVDiEN bit of the interrupt controller (ICU) is 1, a non-maskable interrupt request for comparator Ai is generated.

33.5.2 Maskable Interrupts

When the LVDiCR1.LVDiIRQSEL bit is 1, the comparator Ai interrupt functions as a maskable interrupt.

If the condition for the selected interrupt request is satisfied when the LVDiCR0.LVDiRIE bit is 1 and the LVDiCR0.LVDiDFDIS is 0, the LVDiSR.LVDiMON bit becomes 1 and the comparator Ai interrupt request is generated. At this time, if the setting of the IER0B.IEN0 or IER0B.IEN1 bit in the ICU is 1 (interrupt enabled) and the setting of the IPR88.IPR[3:0] and IPR89[3:0] bits is higher than the level indicated by the PSW.IPL[3:0] bits in the CPU, the IR088.IR or IR089.IR bit becomes 1 (interrupt requested), and comparator Ai generates a maskable interrupt request.

For information on the IEN0B, IR088, IR089, IPR088, and IPR089 registers and on the interrupt vectors, see section 14, Interrupt Controller (ICUb).

33.6 Event Link Output

Comparator A1 outputs the event signals for the event link controller (ELC) in the following timing, and these can be used to initiate operations of other modules selected in advance.

(1) When the input voltage on the CMPA1 pin has crossed the reference voltage on the CVREFA pin

In the same way as for the interrupt sources, the conditions for generation of the event signals for the ELC can be selected as crossing while rising, crossing while falling, or either of the two by setting the LVD1IDTSEL[1:0] bits in LVD1CR1.

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

33.7 Interrupt Handling and Event Linking

Output of event signals to the event link controller (ELC) is independent of the output of interrupt requests to the interrupt controller. Accordingly, event signals are output to the ELC regardless of the settings of the LVD1RIE and LVD1RI bits in LVD1CR0 and the LVD1IRQSEL bit in LVD1CR1.

34. Data Operation Circuit (DOC)

34.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.
Table 34.1 lists the data operation circuit specifications and Figure 34.1 shows a block diagram of the data operation circuit.

- 16-bit data comparison and interrupt generation on a specified condition
- 16-bit data addition
- 16-bit data subtraction

Table 34.1 Specifications of Data Operation Circuit

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Interrupts	• The condition selected by the DOCR.DCSEL bit being met • The result of data addition being greater than FFFFh • The result of data subtraction being less than 0000h
Event link function (output)	• The condition selected by the DOCR.DCSEL bit being met • The result of data addition being greater than FFFFh • The result of data subtraction being less than 0000h
Power consumption reduction function	Module stop state can be set.

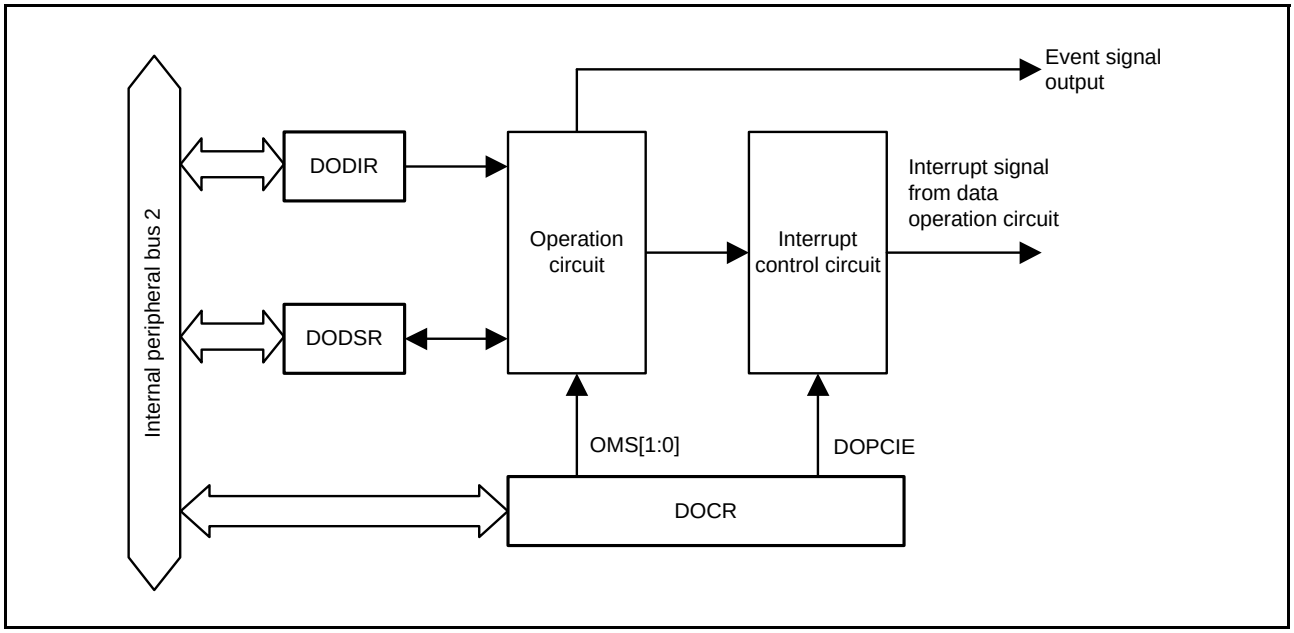


Figure 34.1 Block Diagram of Data Operation Circuit

34.2 Register Descriptions

34.2.1 DOC Control Register (DOCR)

Address: 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	0: Detects mismatch as a result of data comparison. 1: Detects match as a result of data comparison.	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. This bit is only valid in data comparison mode.

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 only valid in data comparison mode.
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 being met
- A result of data addition being greater than FFFFh
- A result of data subtraction being 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.

34.2.2 DOC Data Input Register (DODIR)

Address: 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.

34.2.3 DOC Data Setting Register (DODSR)

Address: 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 in which 16-bit data for use as a reference in data comparison mode are stored. This register also stores the results of operations in data addition and data subtraction modes.

34.3 Operation

34.3.1 Data Comparison Mode

Figure 34.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

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. On setting of the DOCR.DOPCIE bit to 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

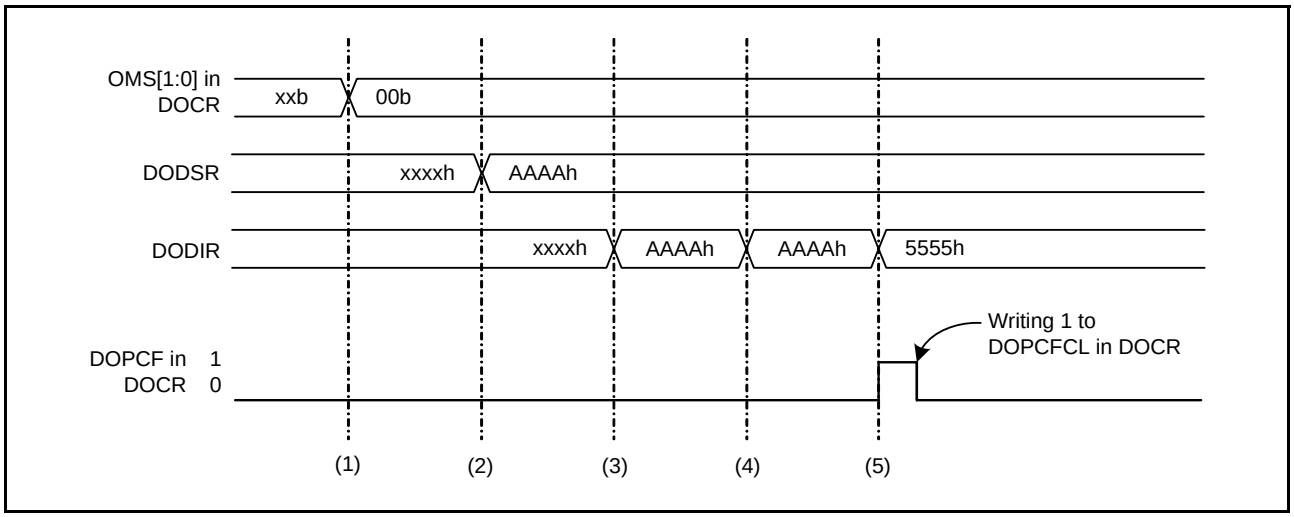


Figure 34.2 Example of Operation in Data Comparison Mode

34.3.2 Data Addition Mode

Figure 34.3 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. A 16-bit initial value is set in DODSR.
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. On setting of the DOCR.DOPCIE bit to 1, a data operation circuit interrupt is also generated.

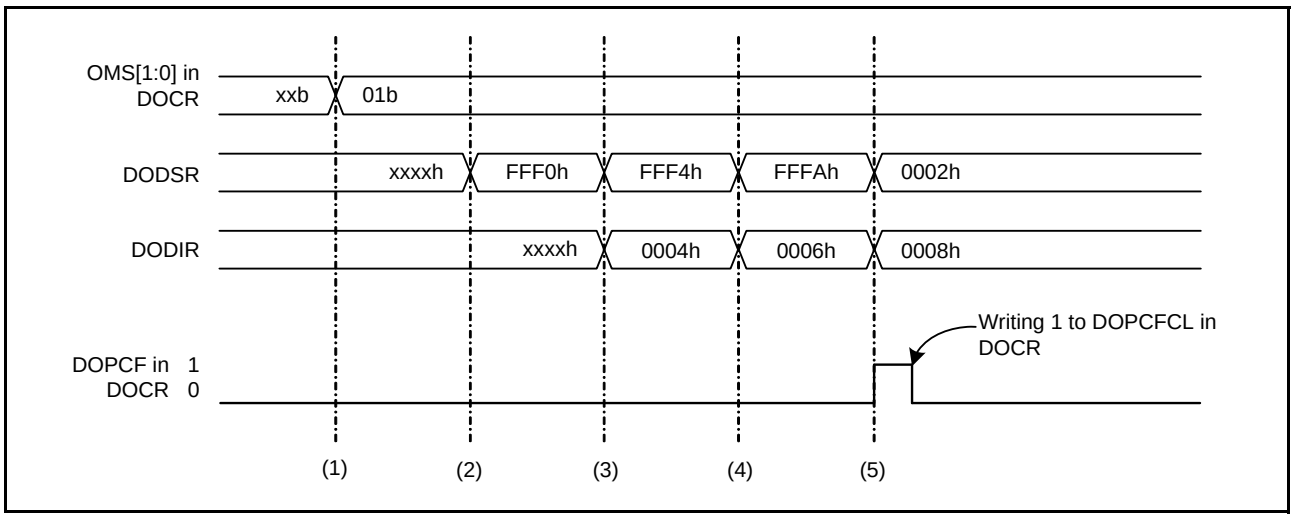


Figure 34.3 Example of Operation in Data Addition Mode

34.3.3 Data Subtraction Mode

Figure 34.4 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. A 16-bit initial value is set in DODSR.
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. On setting of the DOCR.DOPCIE bit to 1, a data operation circuit interrupt is also generated.

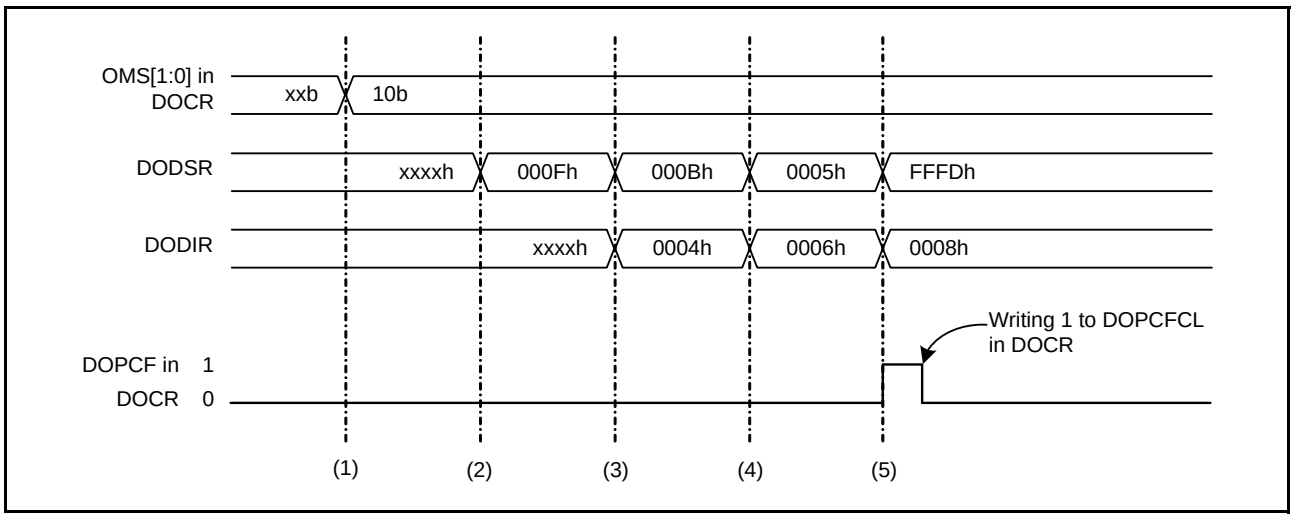


Figure 34.4 Example of Operation in Data Subtraction Mode

34.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt condition arises, the data operation circuit flag corresponding to the interrupt is set to 1. Table 34.2 describes the interrupt request.

Table 34.2 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Conditions
Data operation circuit interrupt	DOPCF	• The condition selected by the DOCR.DCSEL bit being met • The result of data addition being greater than FFFFh • The result of data subtraction being less than 0000h

34.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The condition selected by the DOCR.DCSEL bit being met
- The result of data addition being greater than FFFFh
- The result of data subtraction being less than 0000h

34.5.1 Interrupt Handling and Event Linking

The interrupt produced by the DOC 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.

34.6 Usage Note

34.6.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled by using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be halted. Register access is enabled by canceling the module stop state. For details, see section 11, Low Power Consumption.

35. RAM

The RX220 Group has an on-chip high-speed static RAM.

35.1 Overview

Table 35.1 lists the specifications of the RAM.

Table 35.1 Specifications of RAM

Item	Description
RAM capacity	Max. 16 Kbytes (RAM0: 16 Kbytes)*2
Access	• Single-cycle access is possible for both reading and writing. • On-chip RAM can be enabled or disabled.*1
Low power consumption function	The module stop state is selectable.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.3, System Control Register 1 (SYSCR1).
Note 2. The capacity of RAM differs depending on the products.

RAM Capacity	RAM Address
16 Kbytes	RAM0: 0000 0000h to 0000 3FFFh
8 Kbytes	RAM0: 0000 0000h to 0000 1FFFh
4 Kbytes	RAM0: 0000 0000h to 0000 0FFFh

35.2 Operation

35.2.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to RAM.

Stopping supply of the clock signal places the RAM in the module stop state. The RAM operates after initialization by a reset.
The RAM is not accessible in the module stop state. Do not allow transitions to the module stop state while access to RAM is in progress.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

36. ROM (Flash Memory for Code Storage)

The RX220 Group has a maximum 256-Kbyte flash memory for storing code (ROM).
This section explains the flash memory for code storage. For details on the E2 DataFlash, see section 37, E2 DataFlash Memory (Flash Memory for Data Storage).

36.1 Overview

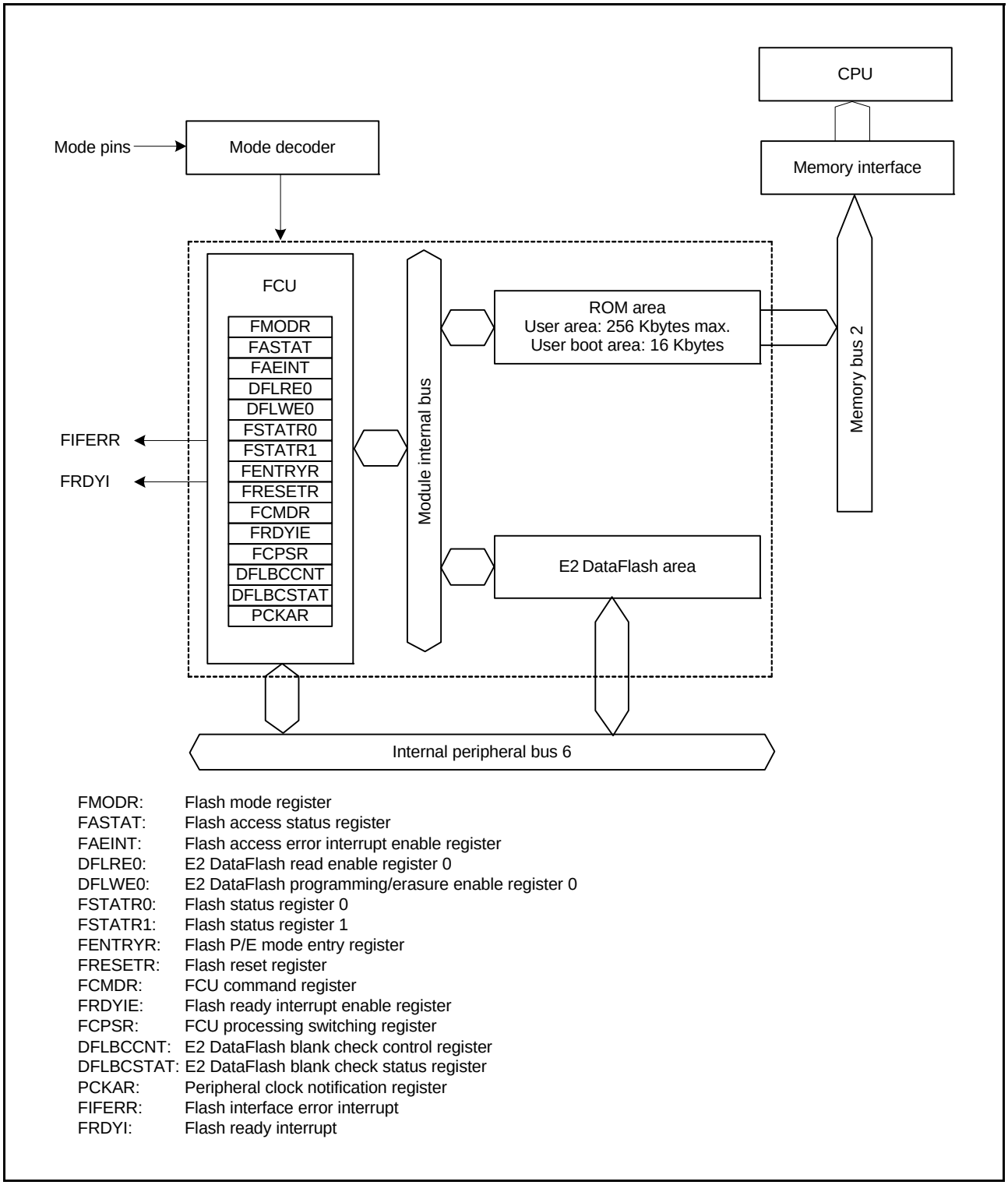
Table 36.1 lists the specifications of the ROM, Table 36.2 lists the correspondence between ROM capacity and ROM addresses, and Figure 36.1 shows a block diagram of the ROM, E2 DataFlash, and related modules.

Table 36.1 Specifications of ROM

Item		Description
Memory capacity		• User area: 256 Kbytes max. • User boot area: 16 Kbytes
High-speed reading		A read operation takes one cycle of ICLK
Programming/erasing method		• The chip incorporates a dedicated sequencer (FCU) for programming of the ROM. • Programming and erasing the ROM are handled by issuing commands to the FCU. • FFFF FFFFh is read from the erased ROM in 32 bits.
BGO (background operation)		• Execution of program code from the ROM is possible while the E2 DataFlash memory is being programmed or erased. • The CPU is able to execute program code from areas other than the ROM or E2 DataFlash while the ROM 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. • Programming and erasure of the ROM can be restarted (resumed) after suspension.
Units of programming and erasure		• Units of programming for the user area or user boot area: 2, 8, or 128 bytes • Units of erasure for the user area: In block units • Units of erasure for the user boot area: 16 Kbytes
On-board programming (three types)		Reprogramming in boot mode • The clock synchronous serial interface (SCI1) is used. • The bit rate is automatically adjusted. • The user boot area is also programmable.
		Reprogramming in user boot mode • The user-specific boot program can be programmed.
		Reprogramming using the ROM reprogramming routine in the user program • ROM is reprogrammable without resetting the system.
Protection	Software-controlled protection	The FENTRYR.FENTRY0, FWEPROR.FLWE[1:0], and lock bits can be set to prevent unintentional programming.
	Command-locked state	When abnormal operations are detected during, programming/erasure, this function disables any further programming/erasure.
Programming/erasure time, number of programming		See section 38, Electrical Characteristics.

Table 36.2 Correspondence between ROM Capacity and ROM Addresses

ROM Capacity	ROM Addresses
32 Kbytes	FFFF 8000h to FFFF FFFFh
64 Kbytes	FFFF 0000h to FFFF FFFFh
128 Kbytes	FFFE 0000h to FFFF FFFFh
256 Kbytes	FFFC 0000h to FFFF FFFFh



FMODR: Flash mode register
FASTAT: Flash access status register
FAEINT: Flash access error interrupt enable register
DFLRE0: E2 DataFlash read enable register 0
DFLWE0: E2 DataFlash programming/erasure enable register 0
FSTATR0: Flash status register 0
FSTATR1: Flash status register 1
FENTRYR: Flash P/E mode entry register
FRESETR: Flash reset register
FCMDR: FCU command register
FRDYIE: Flash ready interrupt enable register
FCPSR: FCU processing switching register
DFLBCCNT: E2 DataFlash blank check control register
DFLBCSTAT: E2 DataFlash blank check status register
PCKAR: Peripheral clock notification register
FIFERR: Flash interface error interrupt
FRDYI: Flash ready interrupt

Figure 36.1 Block Diagram of ROM

36.2 Register Descriptions

Although some registers include bits related to E2 DataFlash, this section deals only with the bits related to ROM. For details on the bits related to the E2 DataFlash, see section 37.2, Register Descriptions (in section 37, E2 DataFlash Memory (Flash Memory for Data Storage)).
P/E indicates programming and erasure.

36.2.1 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, b0	FLWE[1:0]	Flash Programming/ Erasure	b1 b0 0 0: Flash programming/erasure disabled. 0 1: Flash programming/erasure enabled. 1 0: Flash programming/erasure disabled. 1 1: Flash programming/erasure disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FWEPROR is initialized by a reset due to the signal on the RES# pin, by transitions to software standby mode and by the power supply voltage falling below the threshold for detection.

FLWE[1:0] Bits (Flash Programming/Erasure)

These bits protect the execution of the flash programming/erasure with software.

36.2.2 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 Mode Select	0: Memory Area Reading Method Set this bit to 0 when reading ROM lock bits in ROM lock bit read mode. 1: Register Reading Method Set this bit to 1 when using lock bit read 2 command to read ROM lock bits.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FRDMD Bit (FCU Read Mode Select)

This bit is used to select a method for reading lock bits.

If the blank check command for the E2 DataFlash is to be used, this bit has to be set for the register read mode (see section 37, E2 DataFlash Memory (Flash Memory for Data Storage)).

36.2.3 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	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	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	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	R/(W) *1
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation	0: No ROM access error 1: ROM access error	R/(W) *1

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

When any bit from among the DFLWPE, DFLRPE, DFLAE, and ROMAE bits in FASTAT is set to 1, the FSTATR0.ILGLERR bit is set to 1, which places the FCU in the command-locked state (see section 36.8.2, Command-Locked State). To clear the command-locked state, a status register clear command must be issued to the FCU after setting FASTAT to 10h.

CMDLK Bit (FCU Command Lock)

This bit indicates that the FCU is in the command-locked state (see section 36.8.2, Command-Locked State).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU issues a status register clear command under conditions where FASTAT is set to 10h

ROMAE Bit (ROM Access Violation)

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, placing the FCU in the command-locked state.

[Setting conditions]

- Read access to a ROM programming/erasure address when the FCU is in ROM P/E normal mode

ROM Capacity	ROM programming/erasure address ranges FENTRY0 bit is 1
32 Kbytes	00FF 8000h to 00FF FFFFh
64 Kbytes	00FF 0000h to 00FF FFFFh
128 Kbytes	00FE 0000h to 00FF FFFFh
256 Kbytes	00FC 0000h to 00FF FFFFh

- Read/Write access to a ROM programming/erasure address

ROM Capacity	ROM programming/erasure address ranges FENTRY0 bit is 0
32 Kbytes	00FF 8000h to 00FF FFFFh
64 Kbytes	00FF 0000h to 00FF FFFFh
128 Kbytes	00FE 0000h to 00FF FFFFh
256 Kbytes	00FC 0000h to 00FF FFFFh

- Read access to a ROM-reading address while FENTRYR has placed the ROM in ROM P/E mode

ROM Capacity	ROM programming/erasure address ranges
32 Kbytes	FFFF 8000h to FFFF FFFFh
64 Kbytes	FFFF 0000h to FFFF FFFFh
128 Kbytes	FFFE 0000h to FFFF FFFFh
256 Kbytes	FFFC 0000h to FFFF FFFFh

[Clearing condition]

- When 0 is written after reading 1

36.2.4 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	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	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	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the FASTAT.CMDLK bit is set to 1 1: FIFERR interrupt requests enabled when the FASTAT.CMDLK bit is set to 1	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	0: FIFERR interrupt requests disabled when the FASTAT.ROMAE bit is set to 1 1: FIFERR interrupt requests enabled when the FASTAT.ROMAE bit is set to 1	R/W

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an FCU command lock occurs and the FASTAT.CMDLK bit is set to 1.

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.

36.2.5 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	0: Other than the status described below 1: During programming suspend processing or programming suspended	R
b1	ERSSPD	Erasure Suspend Status	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	0: P/E suspend commands cannot be received 1: P/E suspend commands can be received	R
b4	PRGERR	Programming Error	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	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	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 (See section 37, E2 DataFlash Memory (Flash Memory for Data Storage)). 1: Processing described above is not performed	R

FSTATR0 is initialized by a reset, or when the FRESETR.FRESET bit is set to 1

PRGSPD Bit (Programming Suspend Status)

This bit is used to indicate that the FCU enters the programming suspend processing state or programming suspended state. For details, see section 36.7, 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)

This bit is used to indicate that the FCU enters the erasure suspend processing state or erasure suspended state. For details, see section 36.7, 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)

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 FCU enters the command-locked state.

PRGERR Bit (Programming Error)

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 FCU is placed in the command-locked state. For details, see section 36.8.2, 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)

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 FCU is placed in the command-locked state. For details, see section 36.8.2, 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)

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 FCU is placed in the command-locked state. For details, see section 36.8.2, 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 the FCU processes a status register clear command under conditions where FASTAT is set to 10h

FRDY Bit (Flash Ready)

This bit is used to check the processing status of the FCU.

36.2.6 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	0: No error occurs in the FCU processing 1: An error occurs in the FCU processing	R

FSTATR1 is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

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, valid data 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)

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.

36.2.7 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

FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is to enable/disable a FRDYI 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).

36.2.8 Flash P/E Mode Entry Register (FENTRYR)

Address: 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	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: 32, 64, 128, or 256 Kbytes of ROM are in ROM read mode 1: 32, 64, 128, or 256 Kbytes of ROM are in ROM P/E mode	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	See section 37, E2 DataFlash Memory (Flash Memory for Data Storage).	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, either the FENTRYD or FENTRY0 bit must be set to 1. Note that if more than one of these bits are set to 1, the FSTATR0.ILGLERR bit is set to 1 and the FCU enters the command-locked state.
After writing to the FENTRYR to initiate a transition to ROM-reading mode, read the register and confirm that it actually holds the new setting before reading the ROM.
FENTRYR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place 32 Kbytes (read addresses: FFFF 8000h to FFFF FFFFh, programming/erasure addresses: 00FF 8000h to 00FF FFFFh), 64 Kbytes (read addresses: FFFF 0000h to FFFF FFFFh, programming/erasure addresses: 00FF 0000h to 00FF FFFFh), 128 Kbytes (read addresses: FFFE 0000h to FFFF FFFFh, programming/erasure addresses: 00FE 0000h to 00FF FFFFh), or 256 Kbytes (read addresses: FFFC 0000h to FFFF FFFFh, programming/erasure addresses: 00FC 0000h to 00FF FFFFh) of ROM in ROM P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- 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 FENTRY0 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 FENTRY0 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

36.2.9 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 initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

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.

36.2.10 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.

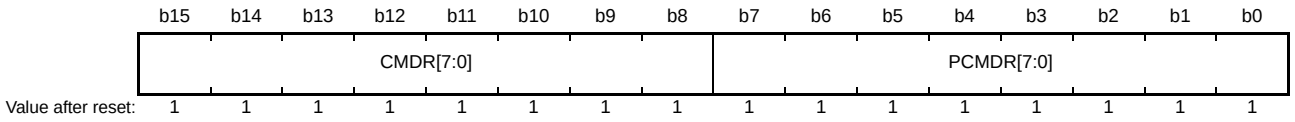
FRESET Bit (Flash Reset)

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 (see section 38, Electrical Characteristics) when initializing the FCU. While the FRESET bit is kept 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.

36.2.11 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 initialized by a reset, or when the FRESETR.FRESET bit is set to 1.
Table 36.3 lists the states of FCMDR after receiving each command. For details on the blank check processing, see section 37.6, Programming and Erasing the E2 DataFlash Memory.

Table 36.3 States of FCMDR after Receiving Each Command

Command	CMDR[7:0]	PCMDR[7:0]
P/E 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

36.2.12 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	Programming/Erasure Suspend Mode	0: Suspension priority mode 1: Programming/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 initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

ESUSPMD Bit (Programming/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 36.7, Suspending Operation.

36.2.13 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 initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

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/E2 DataFlash.

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.

36.2.14 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

PCKAR is initialized by a reset, or when the FRESETR.FRESET bit is set to 1.

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 frequency during programming/erasure for the ROM/E2 DataFlash.
Calculate the setting value as follows:

- Convert an operating frequency represented in MHz units to binary and write it to the PCKA[7:0] bits.

For example, when the operating frequency of the FlashIF clock is 29.9 MHz, the setting value is calculated as follows:

- Round 29.9 off to a whole number.
- Convert 30 to binary and set the upper bits and lower bits of the PCKA[7:0] bits to 00h and 1Eh (0001 1110b).

Note: • 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: • 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: • Please note that the programming time depends on the frequency to some extent even if the PCKA[7:0] bits are used.

36.3 Configuration of Memory Areas for the ROM

The ROM of products in the RX220 Group is configured of a maximum 256-Kbyte user area and a 16-Kbyte user boot area. Figure 36.2 shows the configuration of the ROM memory area.
Note that for the user area, the address range for reading differs from the address range for programming and erasure.

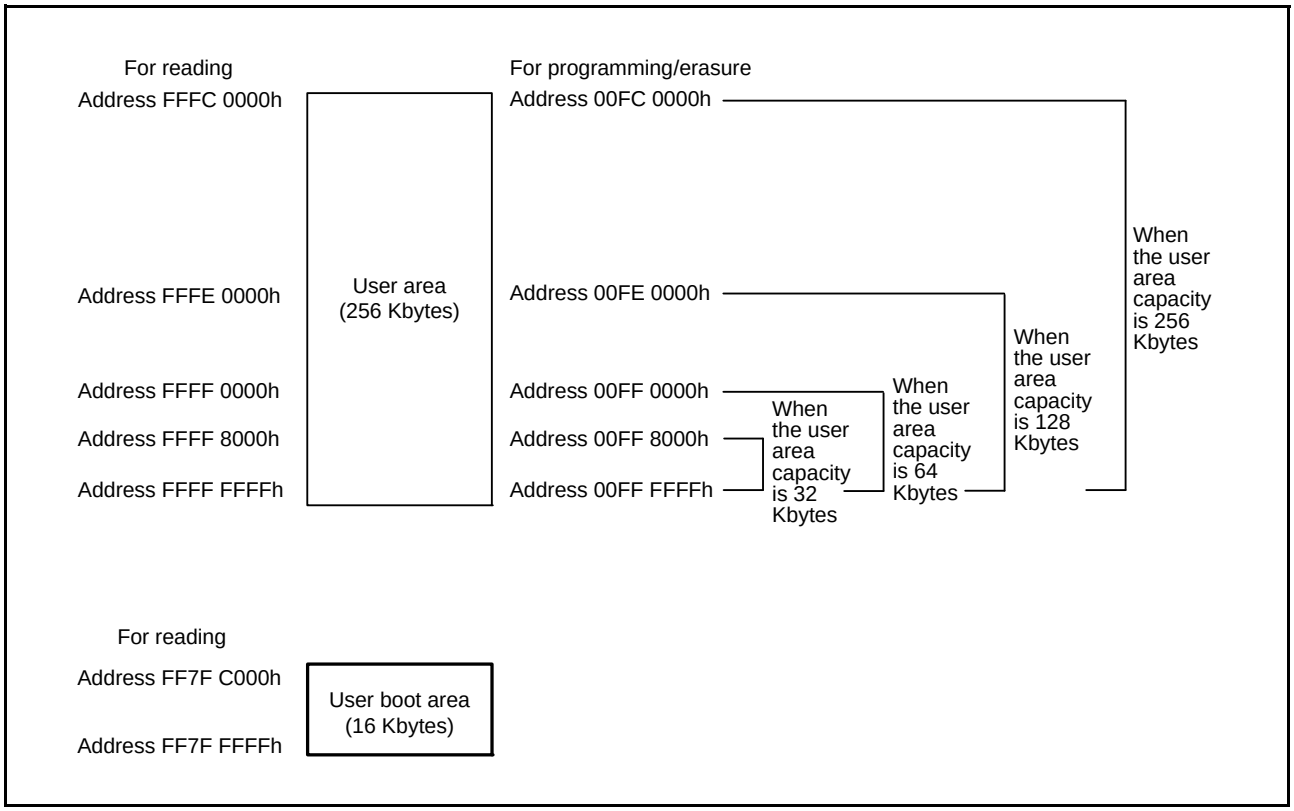


Figure 36.2 Memory Area Configuration of ROM

36.4 Block Configuration

The configuration of erasure blocks for the user area is shown in Figure 36.3.
The user area is divided into following blocks according to the ROM capacity and erasure is executed in block units.
Programming is executed in 2-/8-/128-byte units, where each unit starts at 00h.

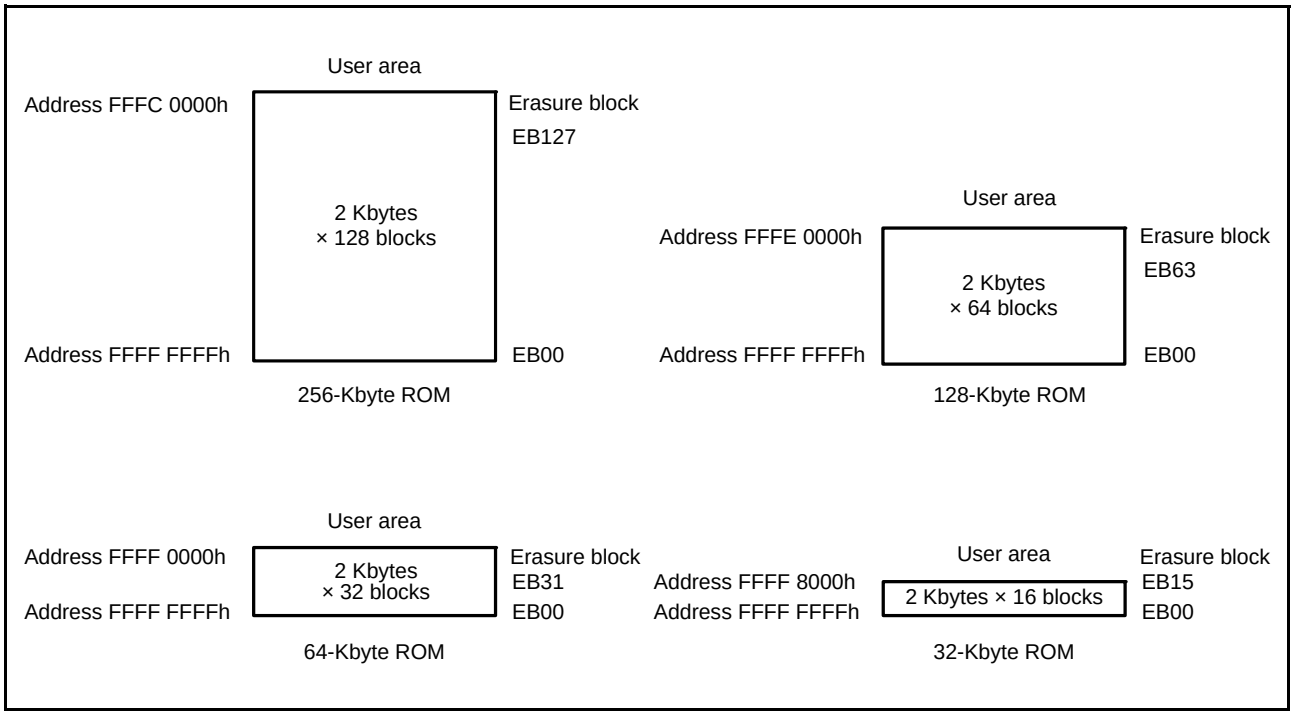


Figure 36.3 Configuration of Erasure Blocks for the User Area

36.5 Operating Modes Associated with the ROM

For information on the relationship between the setting of the level on the MD pin and the operating mode for the RX220 Group, refer to section 3, Operating Modes.
The ROM can be read, programmed, and erased on board in boot mode or single-chip mode.
Which areas are programmable and erasable, the area from which booting up proceeds after a reset, etc., differs with the mode. The differences between modes are indicated in Table 36.4.

Table 36.4 Differences between Modes

Item	Boot Mode	User Boot Mode	Single-Chip 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 36.10.4, 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, RAM is employed for the boot program. For this reason, preserving the contents of RAM is not possible in this case.
- In user boot mode, booting up from the user boot area and then programming and reading of the user area and data area via a desired interface become possible (after booting up from the boot area).

36.6 Programming and Erasing the ROM

The ROM is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.
The mode transitions required to program or erase the ROM and the system of commands are described below. The descriptions apply in common to boot mode, user boot mode, and single-chip mode.

36.6.1 FCU Modes

The FCU has five modes. Transitions between modes are caused by modifying FENTRYR or issuing FCU commands. Figure 36.4 is a diagram of the FCU mode transitions.

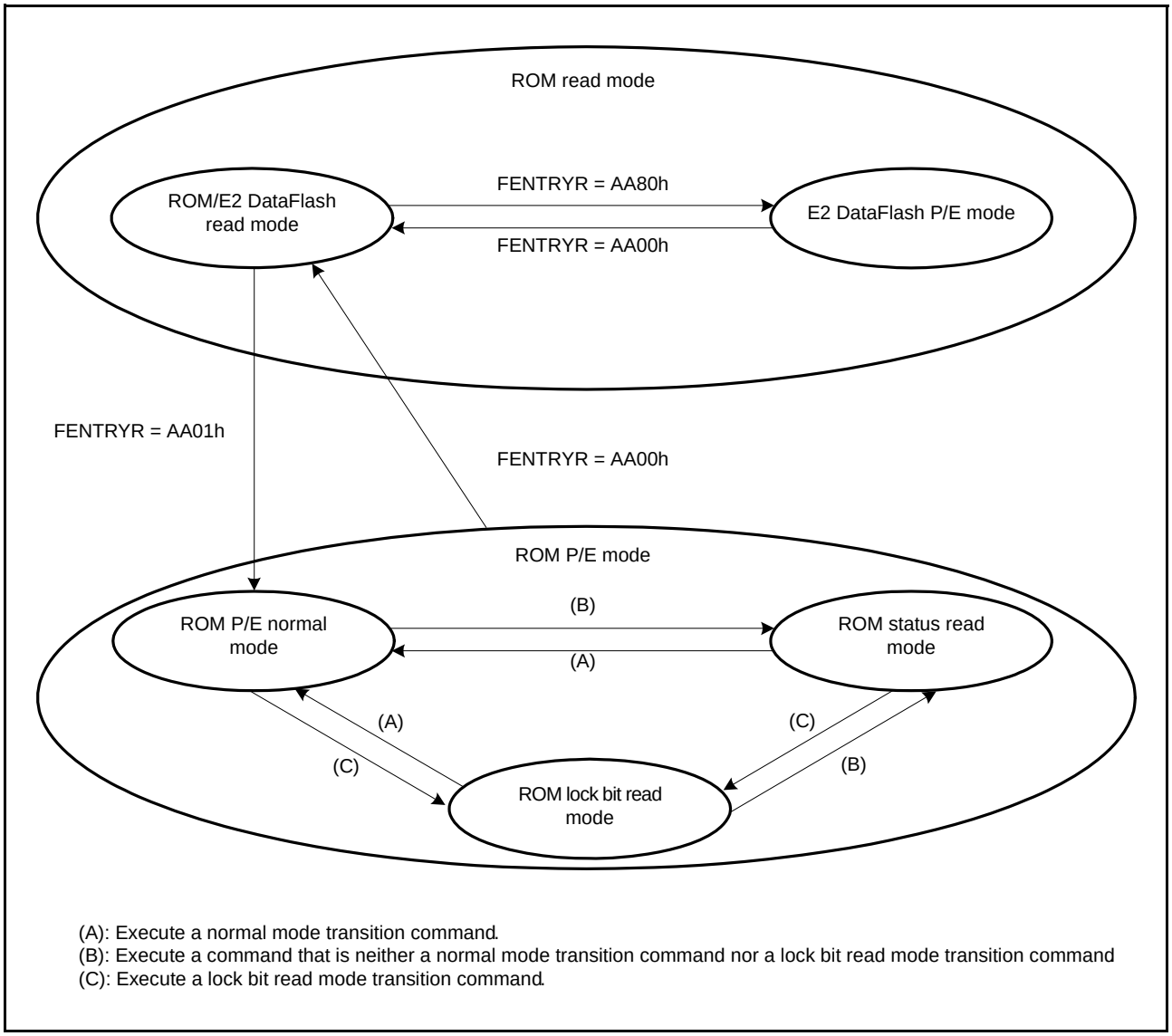


Figure 36.4 Mode Transitions of the FCU (Associated with the ROM)

36.6.1.1 ROM Read Modes

The ROM read modes are for high-speed reading of the ROM. Access to 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.

(1) 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.FENTRY0 bit is set to 0 with the FENTRYR.FENTRYD bit set to 0.

(2) E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. High-speed reading of the ROM is also possible. Although the FCU accepts FCU commands related to the E2 DataFlash memory in this mode, it does not accept FCU commands related to the ROM. The FCU enters these modes when the FENTRYR.FENTRY0 bit is set to 0 with the FENTRYR.FENTRYD bit set to 1.
For details on the E2 DataFlash P/E modes, see section 37.6.1, FCU Modes (in section 37, E2 DataFlash Memory (Flash Memory for Data Storage)).

36.6.1.2 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. Read access to an address within the range for reading causes a ROM-access violation, and the FCU enters the command-locked state (see section 36.8.2, Command-Locked State).
ROM P/E normal mode, ROM status read mode, and ROM lock-bit read mode are the 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 the FENTRYR.FENTRY0 bit set to 1 in ROM read mode, or when the normal mode transition command is received in ROM P/E modes. Table 36.7 lists the acceptable commands in this mode.
Read access to an address within the range for programming and erasure while the FENTRYR.FENTRY0 bit is set to 1 causes a ROM-access violation, and the FCU enters the command-locked state (see section 36.8.2, 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 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 command-locked state after an error has occurred. Table 36.7 lists the acceptable commands in this mode.
Read access to an address within the range for programming and erasure while the FENTRYR.FENTRY0 bit is 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 36.7 lists the acceptable commands in this mode.
Read access to an address within the range for programming and erasure while the FENTRYR.FENTRY0 bit is 1 allows the value of the lock bit of the erasure block including the accessed address to be read from all the read bits.

36.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and commands for programming and erasure. Table 36.5 lists the FCU commands for use with the ROM.

Table 36.5 FCU Commands for Use with the ROM

Command	Description
P/E normal mode transition	Shifts to normal mode (see section 36.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Shifts to status read mode (see section 36.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Shifts to lock bit read mode (see section 36.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the peripheral clock
Programming	ROM programming (in 2-, 8-, or 128-byte units)
Block erase	ROM erasure (in block units, with the lock bit being erased simultaneously)
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 releases the FCU from the command locked state
Lock bit read 2/blank check	Reads the lock bit of a specified erasure block (the value of the lock bit is reflected in the FSTATR1.FLOCKST bit)/blank checking of the E2 DataFlash memory
Lock bit programming	Programs the lock bit of a specified erasure block

The lock bit read 2 command is 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 (see section 37, E2 DataFlash Memory (Flash Memory for Data Storage)).

Commands for the FCU are issued by write access to addresses within the range for ROM programming and erasure. Table 36.6 lists the formats of the FCU commands. Write access as listed in Table 36.6 and in accordance with certain conditions causes the FCU to execute processing for the corresponding command.

For details on the conditions for the acceptance of the individual FCU commands, see section 36.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 36.6.4, FCU Command Usage.

Table 36.6 FCU Command Formats

Command	Number of Bus Cycles	1st Cycle		2nd Cycle		3rd Cycle		4th to 5th Cycle		6th Cycle		7th to (N+2)th Cycle		(N+3)th Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
P/E normal mode transition	1	RA	FFh	—	—	—	—	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	70h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	RA	71h	—	—	—	—	—	—	—	—	—	—	—	—
Peripheral clock notification	6	RA	E9h	RA	03h	WA	0F0Fh	RA	0F0Fh	RA	D0h	—	—	—	—
Programming (2-byte programming; N = 1)	4	RA	E8h	RA	01h	WA	WDn	—	—	—	—	—	—	RA	D0h
Programming (8-byte programming; N = 4)	7	RA	E8h	RA	04h	WA	WDn	RA	WDn	RA	WDn	—	—	RA	D0h
Programming (128-byte programming; N = 64)	67	RA	E8h	RA	40h	WA	WDn	RA	WDn	RA	WDn	RA	WDn	RA	D0h
Block erase	2	RA	20h	BA	D0h	—	—	—	—	—	—	—	—	—	—
P/E suspend	1	RA	B0h	—	—	—	—	—	—	—	—	—	—	—	—
P/E resume	1	RA	D0h	—	—	—	—	—	—	—	—	—	—	—	—
Status register clear	1	RA	50h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	71h	BA	D0h	—	—	—	—	—	—	—	—	—	—
Lock bit programming	2	RA	77h	BA	D0h	—	—	—	—	—	—	—	—	—	—

Address column RA: ROM programming/erasure address
When the FENTRYR.FENTRY0 bit is 1 with 32-Kbyte ROM: An address from 00FF 8000h to 00FF FFFFh
When the FENTRYR.FENTRY0 bit is 1 with 64-Kbyte ROM: An address from 00FF 0000h to 00FF FFFFh
When the FENTRYR.FENTRY0 bit is 1 with 128-Kbyte ROM: An address from 00FE 0000h to 00FF FFFFh
When the FENTRYR.FENTRY0 bit is 1 with 256-Kbyte ROM: An address from 00FC 0000h to 00FF FFFFh
WA: ROM programming-destination address
Start address for programming of 2/8/128 bytes of data
BA: ROM erasure block address
An address within the target erasure block (specified as an address in the range for programming and erasure)
Data column WDn: nth word of data for programming (n = 1 to 64)

36.6.3 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 36.7. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 36.8.2, 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 36.7 Acceptable Commands and the State and Mode (ROM 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 during Erasure Suspended State	Processing to Suspend Programming or Erasure	Lock Bit Read 2 Processing	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**	**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**
FSTATR0.ERSSPD bit	**0**	**1**	**0**	**0**	**1**	**0/1**	**0**	**0**	**1**	**0/1**	**0/1**	**0**	**0**	**1**	**0**
FSTATR0.PRGSPD bit	**1**	**0**	**0**	**0**	**0**	**0/1**	**0**	**1**	**0**	**0/1**	**0/1**	**0**	**1**	**0**	**0**
FASTAT.CMDLK bit	**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
Lock bit read 2	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit programming	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A

A: Acceptable
*: Only programming is acceptable for blocks other than the block where erasure was suspended
×: Not acceptable

36.6.4 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 36.6.3, Connections between FCU Modes and Commands.

36.6.4.1 Mode Transitions

This subsection covers the commands for mode transitions. For an illustration of the various transitions between modes, see Figure 36.4.

(1) Switching to ROM P/E Mode

A transition to ROM P/E mode is required before executing an FCU command for the ROM becomes possible.
Setting the FENTRY0 bit in FENTRYR to 1 causes a transition to ROM P/E mode for programming and erasure of the corresponding address range.
Before proceeding to program or erase the ROM, enable programming and erasure by writing 01h as a byte to FWEPROR (see section 36.2.1, Flash Write Erase Protection Register (FWEPROR)).

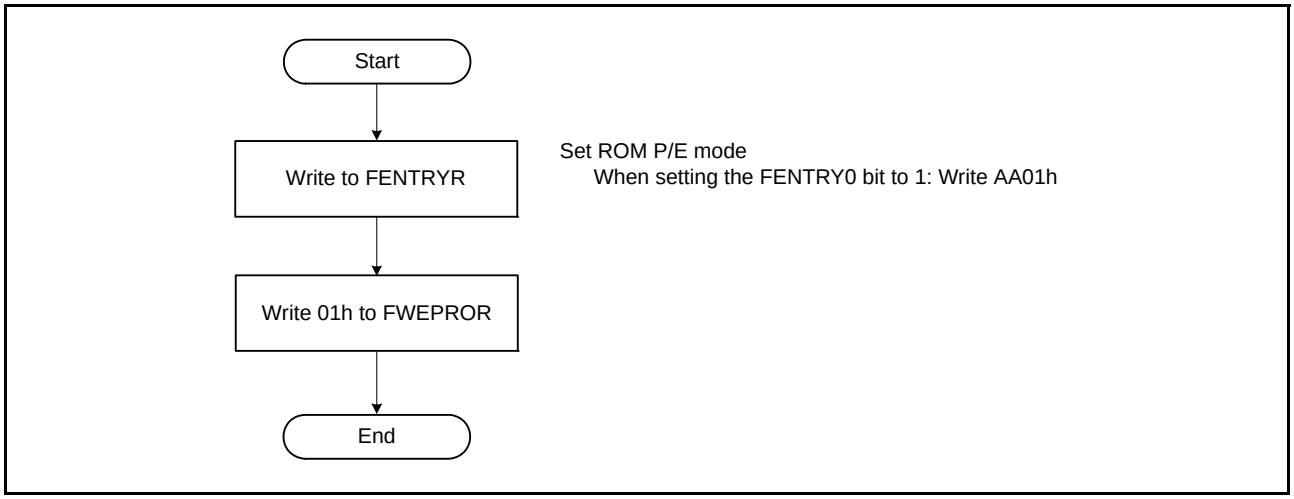
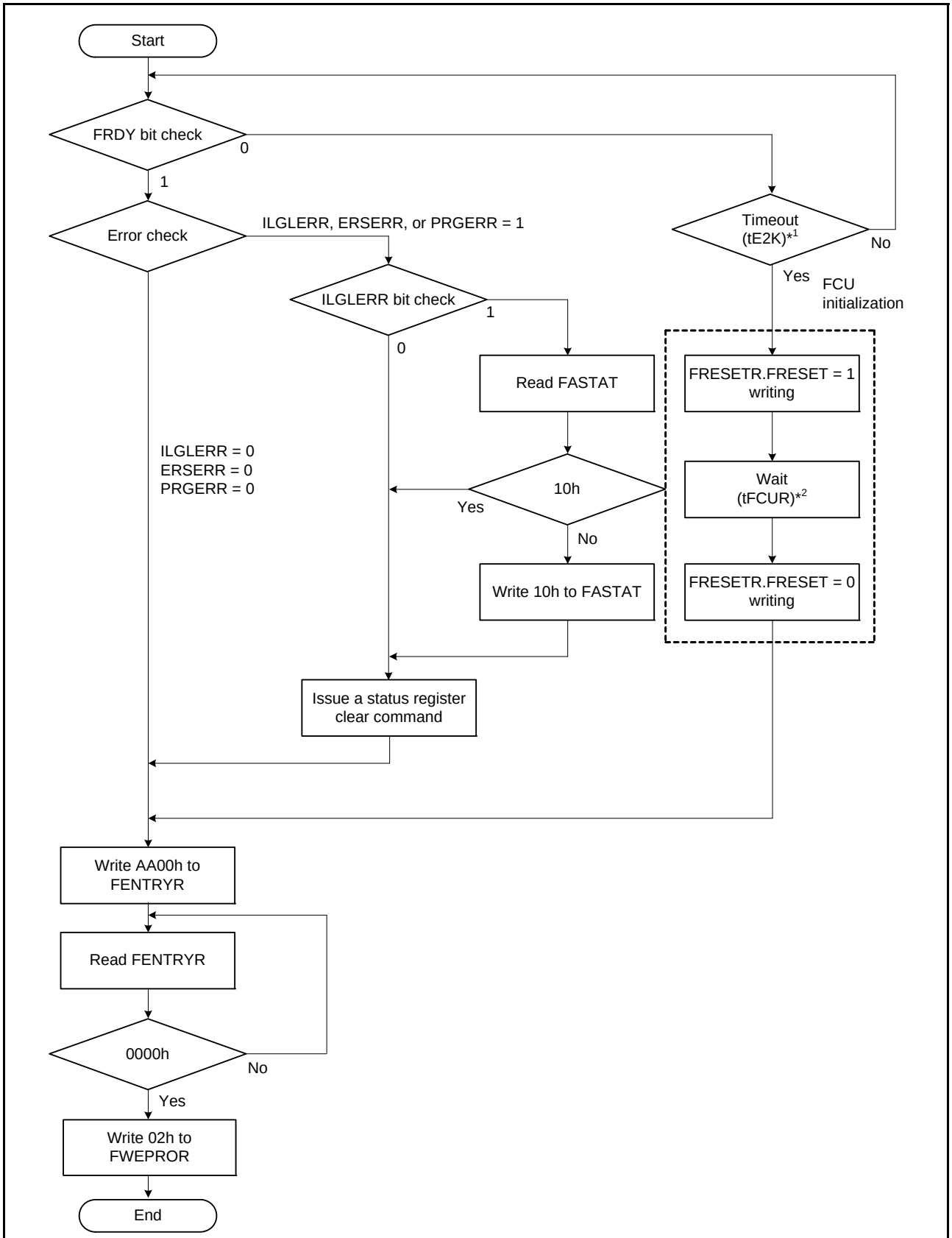


Figure 36.5 Procedure for Transition to ROM P/E Mode

(2) Switching to ROM Read Mode

High-speed reading of the ROM requires clearing of the FENTRY0 bit in FENTRYR, 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 36.2.1, 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.



Note 1. tE2K: Erasure time for a 2-Kbyte erasure block (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.6 Procedure for Transition to ROM Read Mode

(3) Switching to ROM P/E Normal Mode

Two methods are available for the transition to ROM P/E normal mode: setting FENTRYR while the FCU is in ROM read mode (see section 36.6.1, FCU Modes), or issuing the normal mode transition command while the FCU is in ROM P/E mode (see Figure 36.7). The normal mode transition command is issued by writing FFh as a byte to a ROM programming/erasure address.

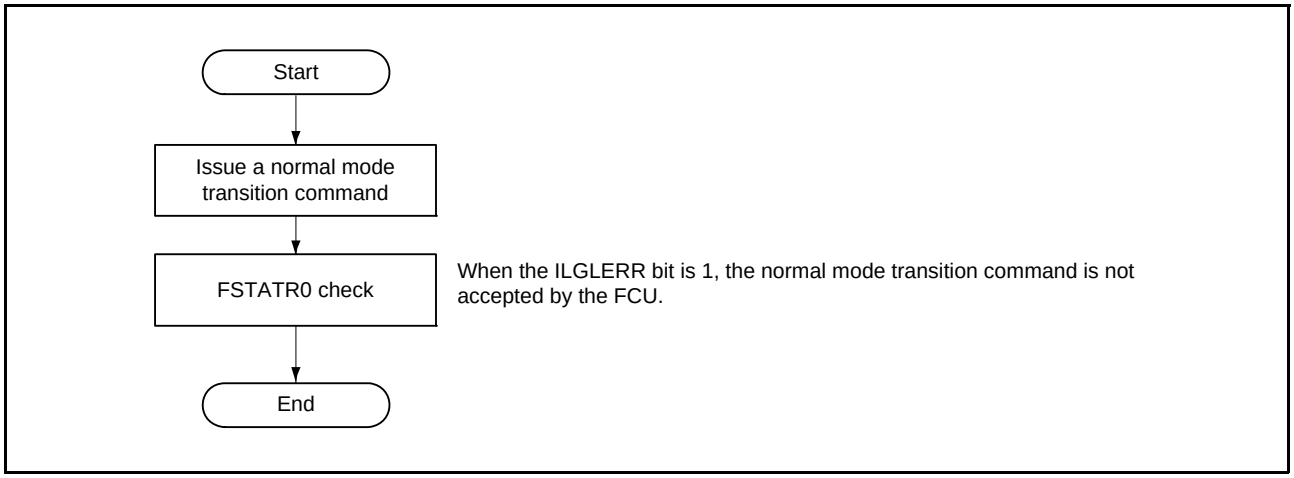


Figure 36.7 Procedure for Transition to ROM P/E Normal Mode

(4) Switching to ROM Status Read Mode

Issuing an FCU command other than a normal mode transition or lock bit read mode transition command places the FCU in ROM status read mode. The same transition can be obtained by issuing the status read mode transition command. Figure 36.8 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 read access to a ROM programming/erasure address and then checked.

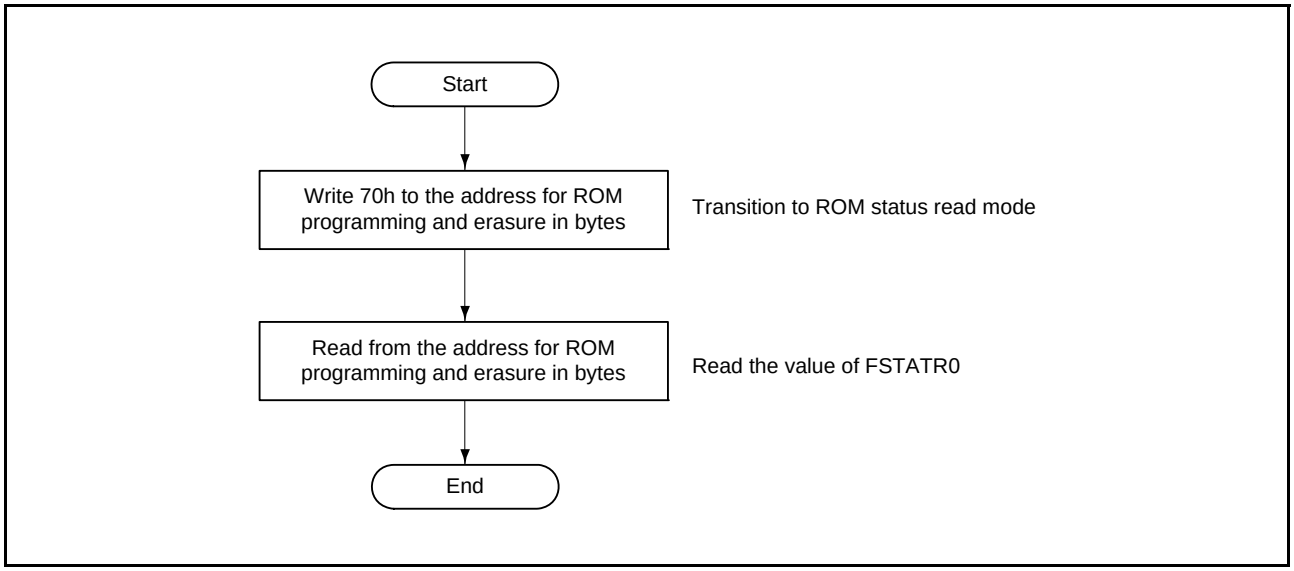


Figure 36.8 Procedure for Transition to ROM Status Read Mode and the Status Checking

(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 ROM lock bit read mode, the lock bit value is obtained by read access to a ROM programming/erasure address. All bits of a value thus read have the value of the lock bit of the erasure block that contains the accessed address (Figure 36.9).

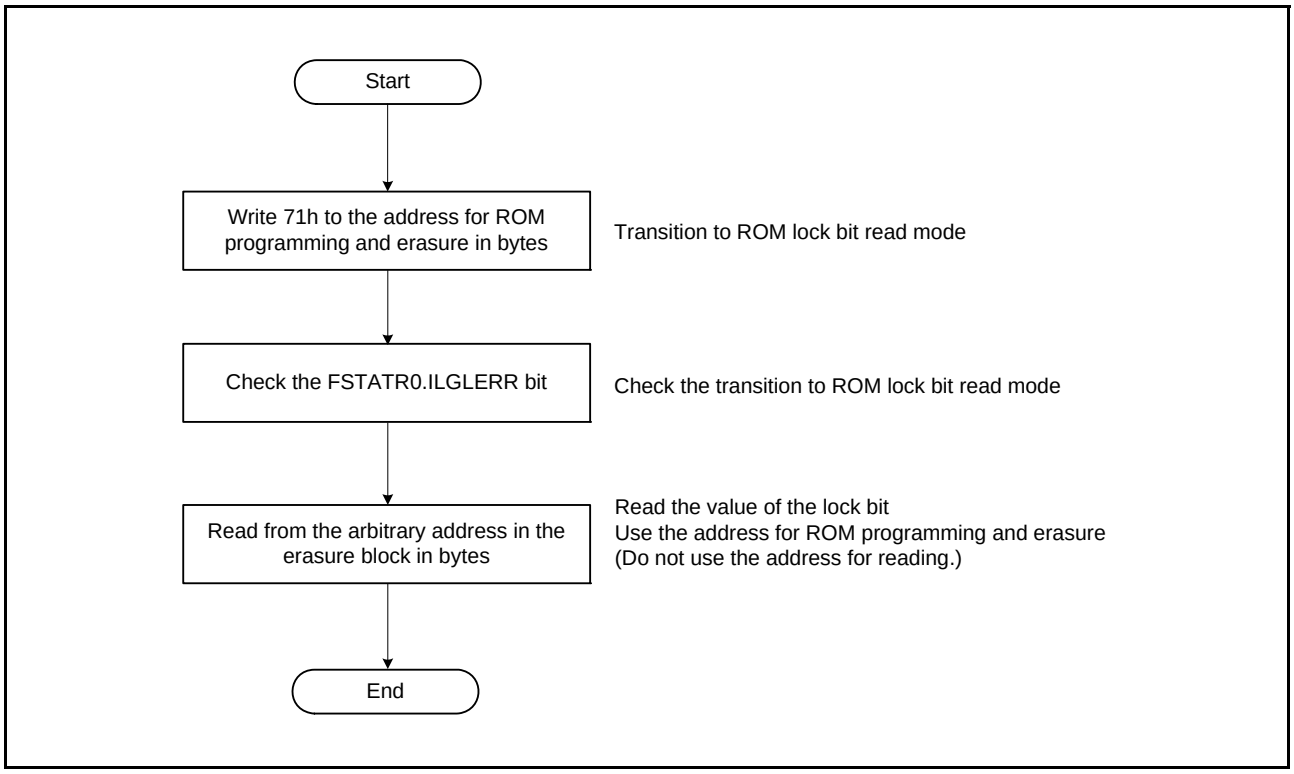
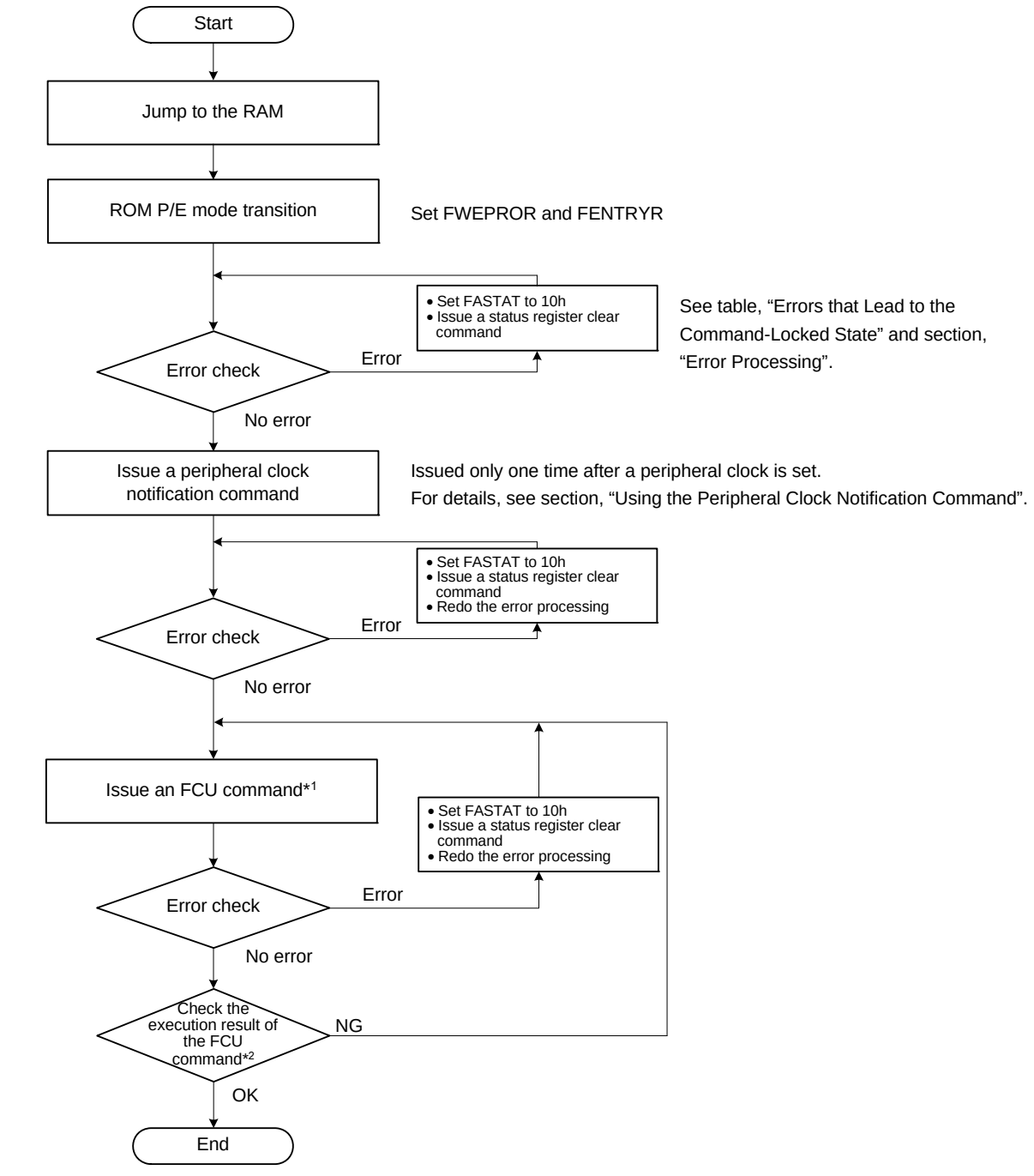


Figure 36.9 Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading

36.6.4.2 Programming and Erasure Procedures

The following passages describe the flow of procedures for programming or erasing the ROM. For details on the acceptance of commands by the FCU, see section 36.6.3, Connections between FCU Modes and Commands. Figure 36.10 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 and read data from the ROM (see section, “Switching to ROM Read Mode”).

Figure 36.10 Simple Flowchart of the Procedure for Programming and Erasure

(1) Jumping to Locations in RAM

Since fetching instructions from the ROM is not possible while the ROM is being programmed or erased, code has to be executed from an area other than the ROM. Copy the required instruction code to RAM and then make a jump to the address where the code starts in RAM.

(2) Transition to ROM P/E Mode

The FCU is placed in ROM P/E mode by setting the FENTRY0 bit in FENTRYR and FWEPROR register. For details, see section 36.6.4.1, (1) Switching to ROM P/E Mode.

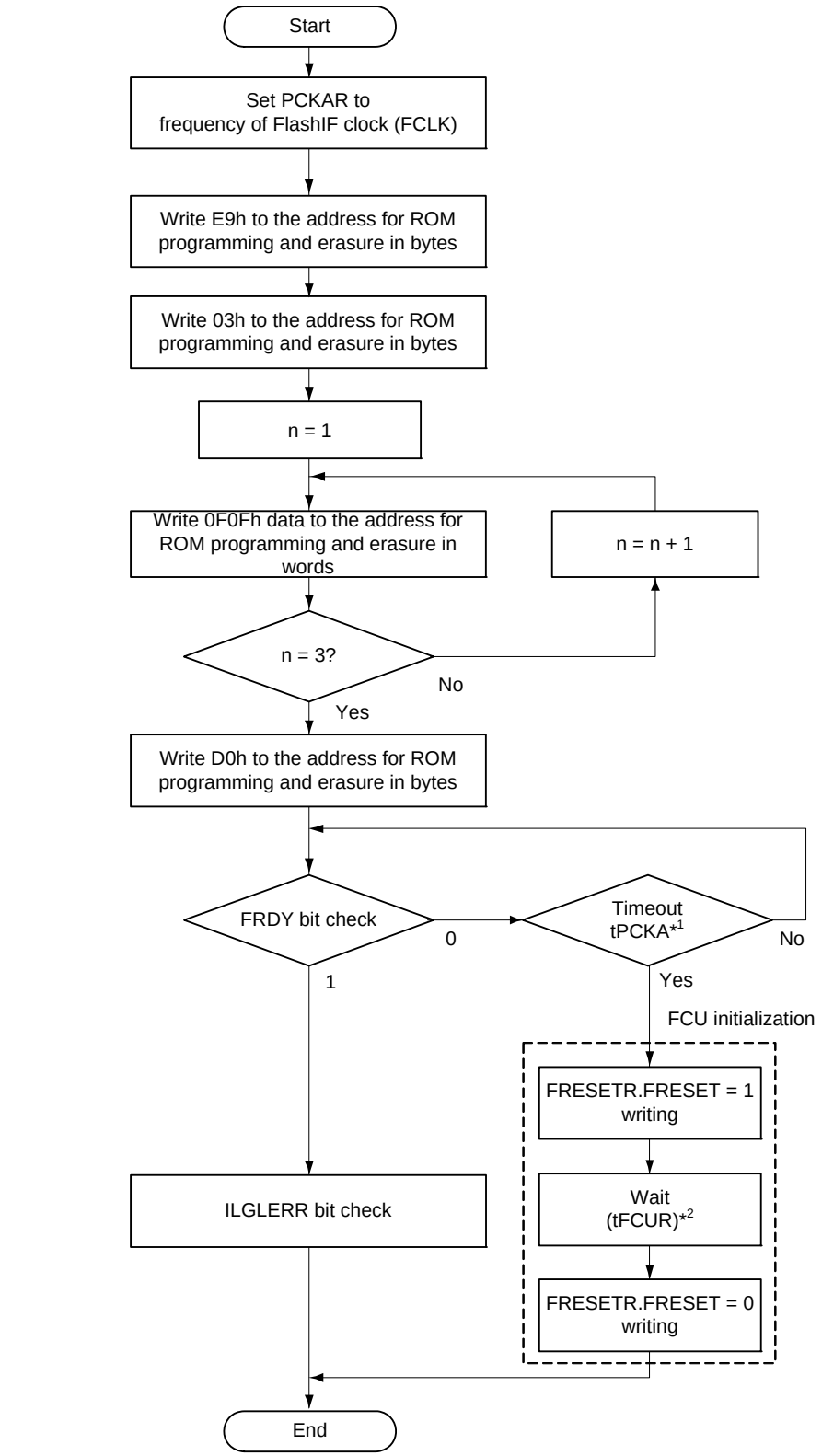
(3) Using the Peripheral Clock Notification Command

The FlashIF clock (FCLK) is used in programming and erasing the ROM, so the frequency of this clock has to be set in PCKAR. Frequencies in the range from 4 MHz to 32 MHz are selectable. If a frequency within this range has not been set, the FCU will detect the error and enter the command-blocked state (see section 36.8.2, 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 as a byte to the address range for programming and erasure of the ROM. Word-size writing is used 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, the process of the FCU setting the frequency of the peripheral clock starts once the value D0h has been written as a byte in the sixth cycle. The FSTATR0.FRDY bit can be used to check whether or not the settings have been completed.

Addresses that can be used in the first to sixth cycles differ according to the settings of the FENTRY0 bit in FENTRYR. Ensure that the addresses suit the settings of the FENTRY0 bit in FENTRYR. If issuing of the command is attempted with an erroneous combination of the settings of the FENTRY0 bit in FENTRYR and the specified addresses, the FCU will detect the error and enter the command-blocked state (see section 36.8.2, 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, this setting is also valid for the next FCU command.

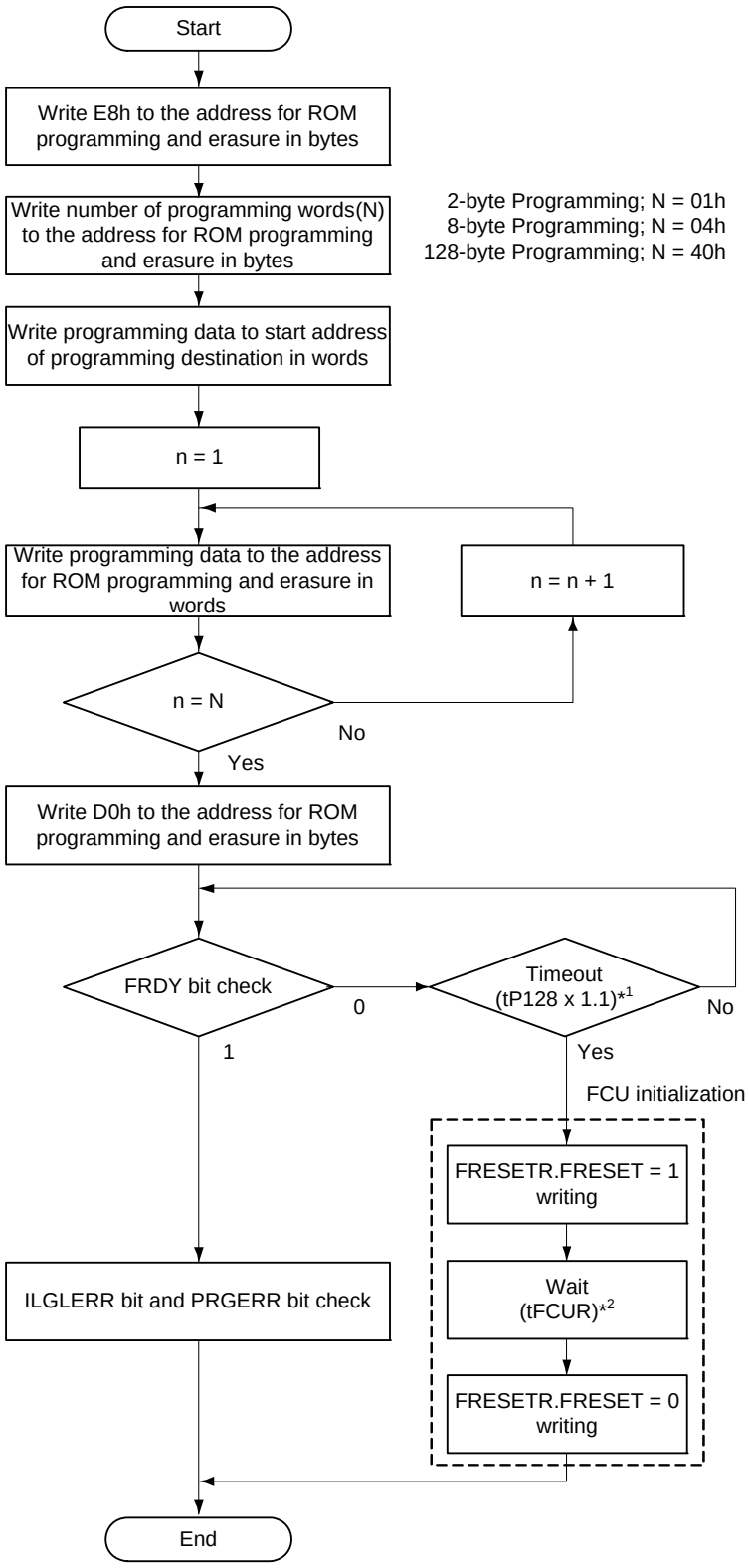


Note 1. tPCKA: 120 µs when FCLK = 25 MHz
Note 2. tFCUR: 960 µs when FCLK = 4 MHz, reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.11 Using the Peripheral Clock Notification Command

(4) Programming

The programming command is used to write data to the ROM.
In the first cycle of the programming command, the value E8h is written as a byte to the address range for programming and erasure of the ROM. In the second cycle, the values, 01h (for 2-byte programming), 04h (for 8-byte programming), or 40h (for 128-byte programming) are written to the same address range. 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.
In the case of 128-byte programming, 128 bytes (64 words) of data are written to the ROM in the third to the 66th cycles, divided into 64 rounds. The start address of 128 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 128. The address range specified in the fourth to the 66th cycles does not need to be the address range for actual programming.
In the case of 8-byte programming, 8 bytes (4 words) of data are written to the ROM in the third to the sixth cycles, divided into four rounds. The start address of 8 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 8. The address range specified in the fourth to the sixth cycles does not need to be the address range for actual programming.
In the case of 2-byte programming, the address range and data for programming are specified in the third cycle. The address range must be an even number.
Once the value D0h has been written as a byte 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.FENTRY0 bit. Ensure that the addresses suit the setting of FENTRYR.FENTRY0 bit. If issuing of the command is attempted with an erroneous combination of the setting of the FENTRYR.FENTRY0 bit and the specified addresses, the FCU will detect the error and enter the command-blocked state (see section 36.8.2, 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. To execute programming with lock bit protection disabled, proceed with programming after setting the FPROTR.FPROTCN bit to 1.

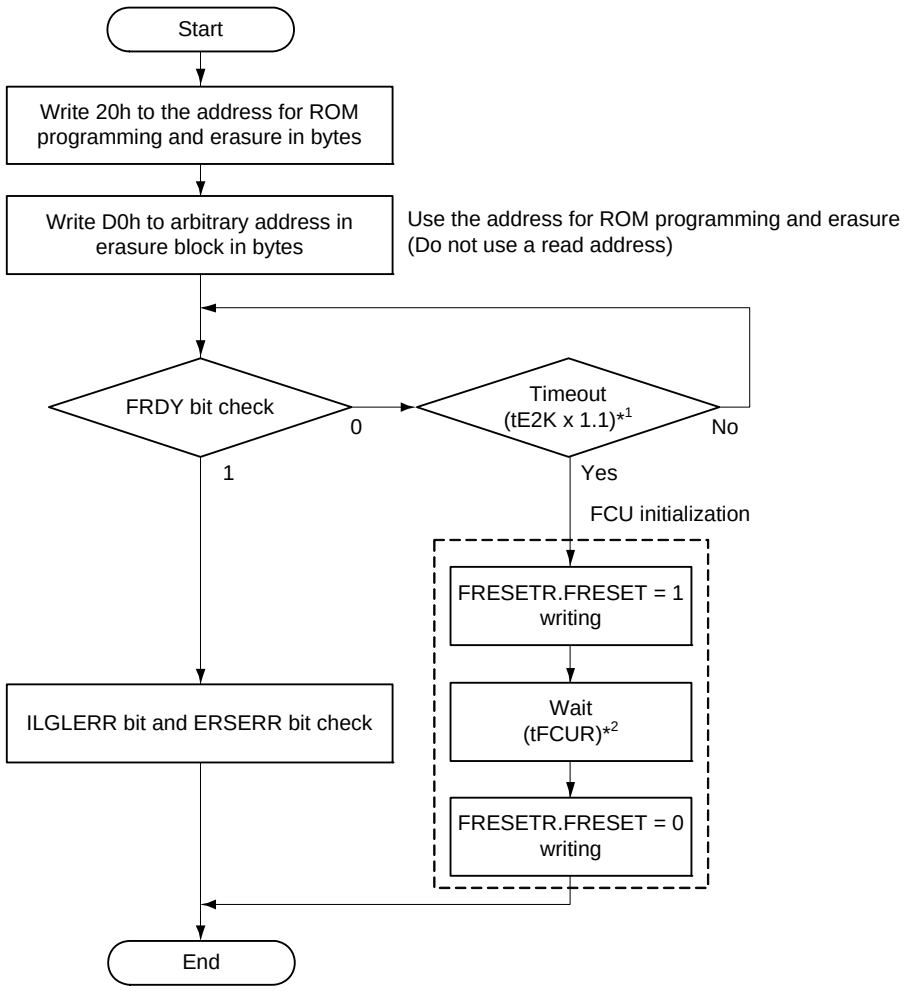


Note 1. tP128: Programming time for 128-byte data (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.12 Procedure for ROM Programming

(5) Erasure

To erase data from the ROM, use the block erase command.
Write 20h to the ROM programming/erasure address in byte access in the first cycle of the block erase command. When D0h is written to an arbitrary address in a target erasure block in byte access in the second cycle, the FCU starts the erasure processing for the ROM. Whether erasure is completed can be checked with the FSTATR0.FRDY bit. Reading the erased ROM by the CPU returns FFFF FFFFh in 32 bits.
To execute an erasure with lock bit protection disabled, set the FPROTR.FPROTCN bit before erasure.

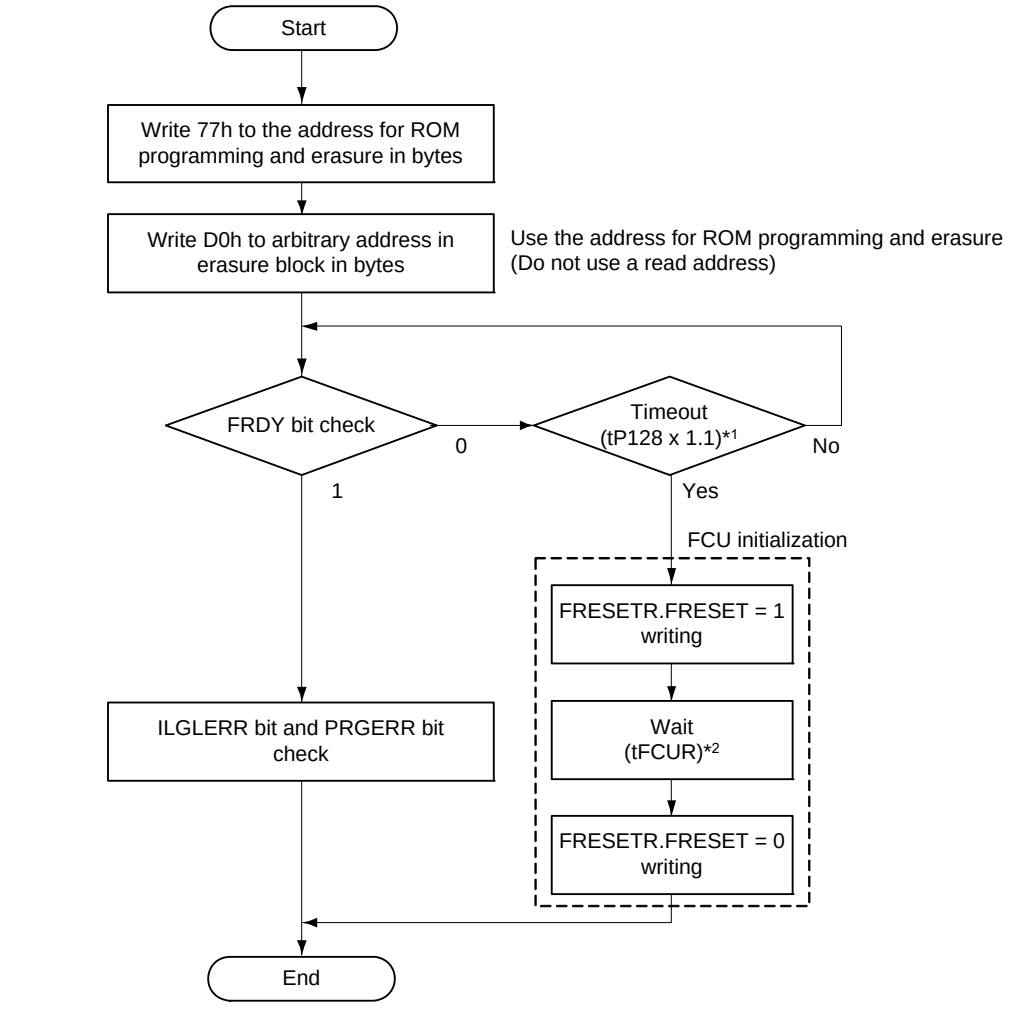


Note 1. tE2K: Erasure time for a 2-Kbyte erasure block (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.13 Procedure for ROM Erasure

(6) Writing to/Erasing Lock Bits

Each erasure 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 as a byte. When D0h is written to an arbitrary address in an erasure block whose lock bit is to be written to in the second cycle as a byte, the FCU starts the processing to write to the lock bit. Whether writing is completed can be checked with the FSTATR0.FRDY bit.



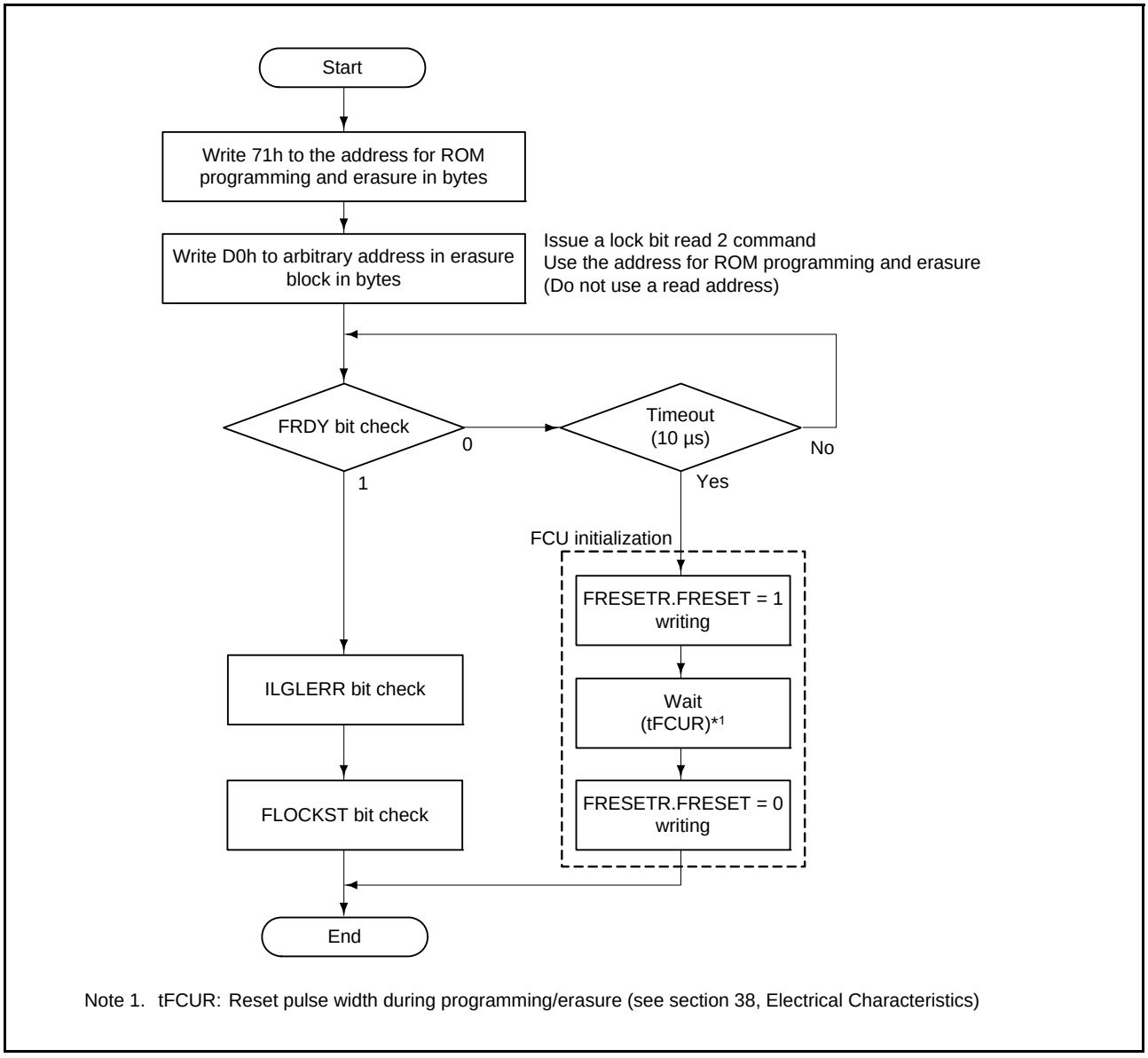
Note 1. tP128: Programming time for 128-byte data (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.14 Procedure for Programming the Lock Bit

To erase a lock bit, use the block erase command.
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 erasure block. It is impossible to erase only a lock bit.

(7) Reading Lock Bits

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 erasure 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 as bytes; once these values have been written, the value of the lock bit for the specified erasure 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 section 36.6.4.1, (5) Switching to ROM Lock-Bit Read Mode.



Note 1. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.15 Procedure for Reading Lock Bit in Register Read Mode

36.6.4.3 Error Processing

The following passages describe the flow of error processing. For details on errors, see section 36.8, Protection.

(1) 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 36.6.4.1, (4) Switching to ROM Status Read Mode.

(2) 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 FCU is placed in the command-locked state and 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.

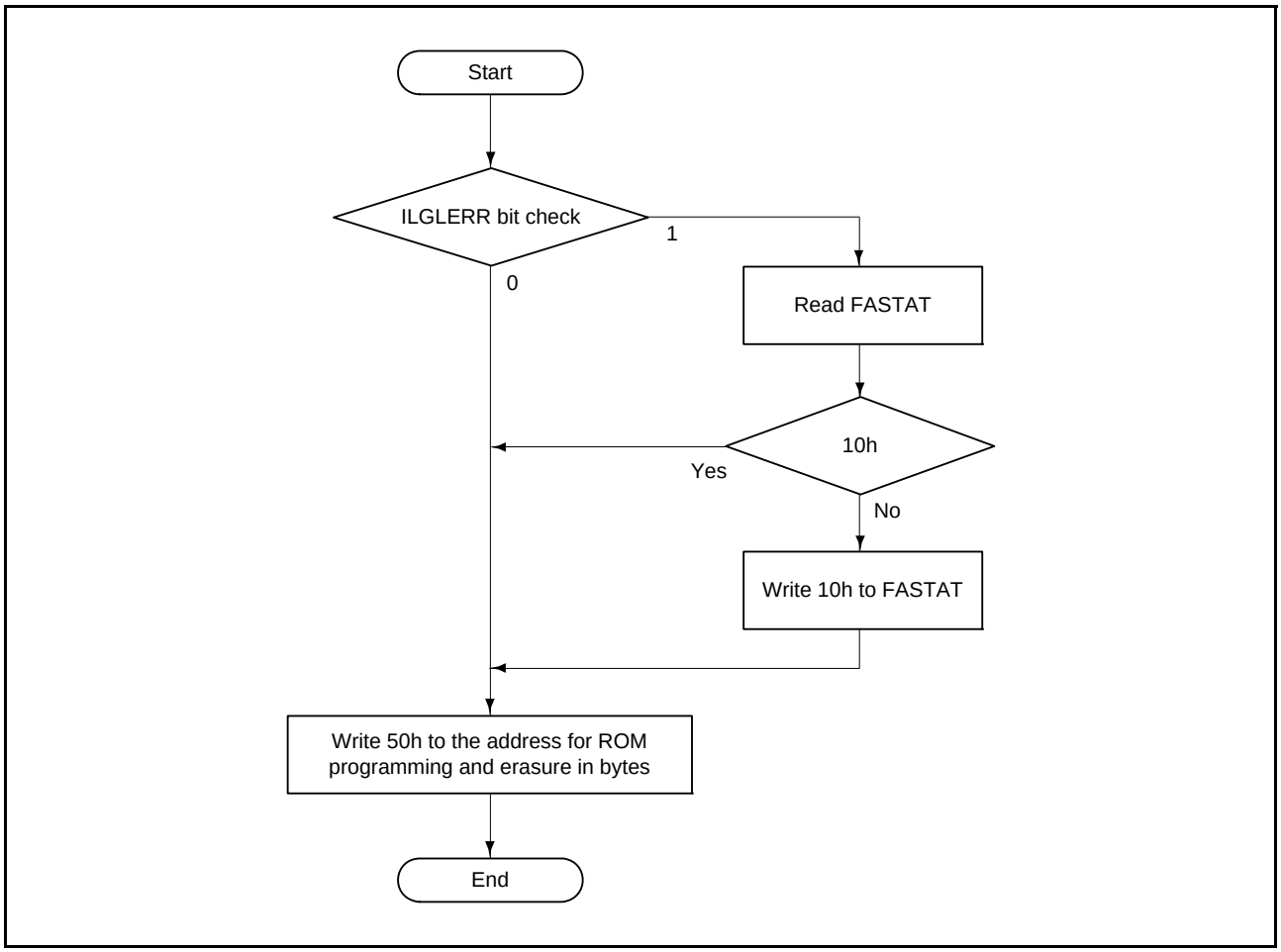


Figure 36.16 Procedure for Clearing FSTATR0

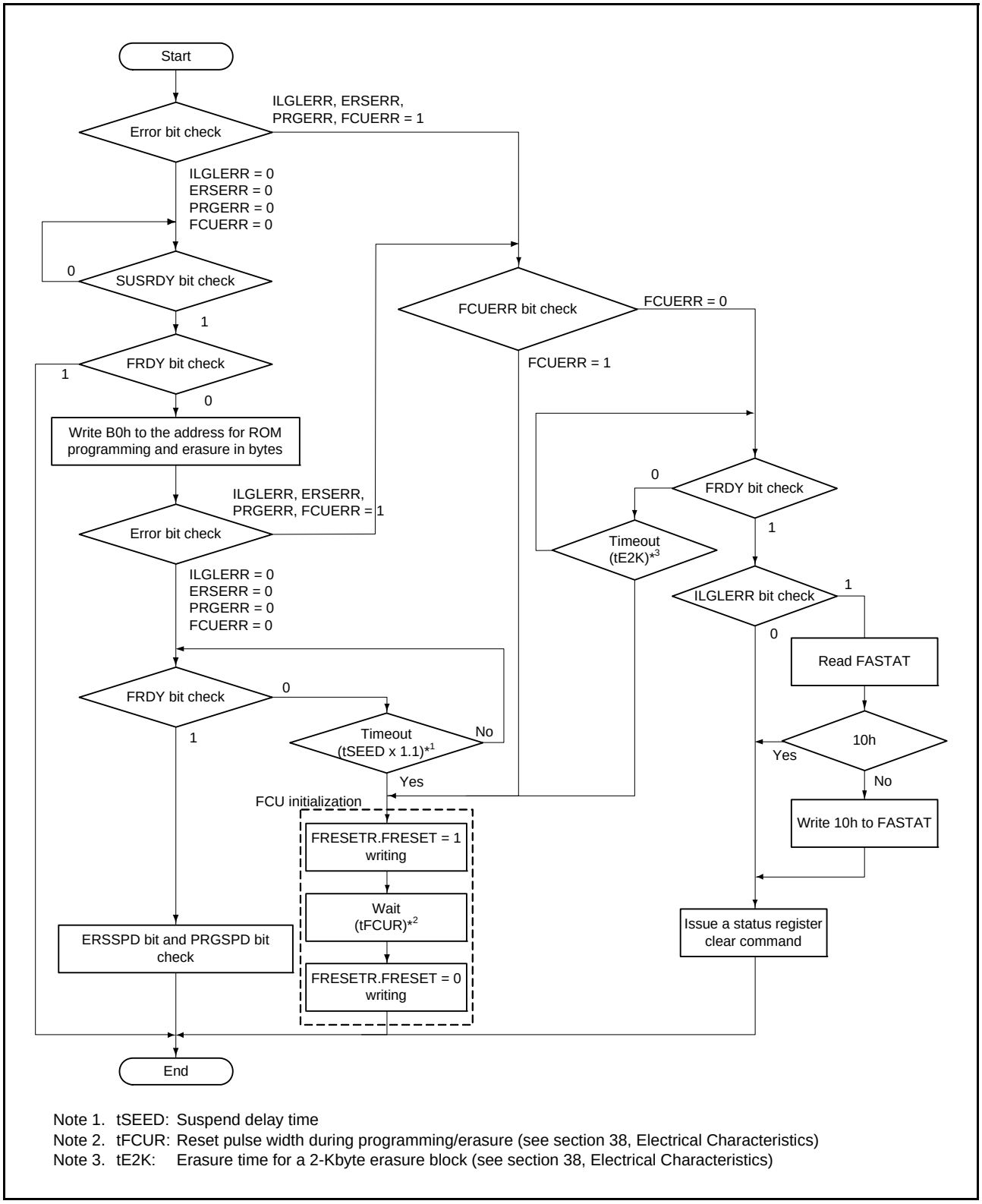
(3) Initializing the FCU

When a timeout leads to the FSTATR0.FRDY bit not being set to 1 after an FCU command has been issued, FRESETR must be used to initialize the FCU. FCU initialization by FRESETR is also necessary when the FSTATR1.FCUERR bit is 1. In either case, maintain the FRESETR.FRESET bit set to 1 for a period of tFCUR (see section 38, 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 36.10.

36.6.4.4 Suspension and Resumption

(1) Suspending Programming or Erasure

To suspend programming/erasure for the ROM, 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 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 FCU is placed in the command-locked state (see section 36.8.2, 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 36.7, Suspending Operation.

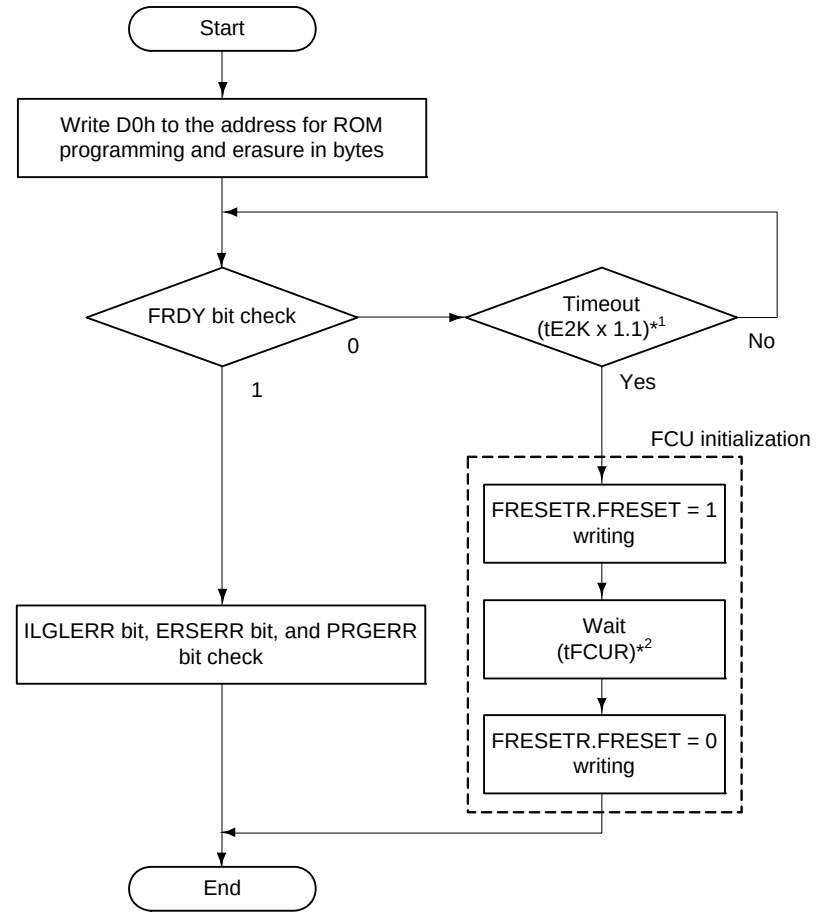


Note 1. tSEED: Suspend delay time
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)
Note 3. tE2K: Erasure time for a 2-Kbyte erasure block (see section 38, Electrical Characteristics)

Figure 36.17 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 FENTRYR register settings are changed during suspension, reset the FENTRYR register to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command.



Note 1. tE2K: Erasure time for a 2-Kbyte erasure block (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 36.18 Procedure for Resuming Programming or Erasure

36.7 Suspending Operation

The ROM cannot be read during programming/erasure. The ROM can be read by suspending the ROM programming/erasure with the P/E suspend command. The P/E suspend command includes two programming modes (suspension priority mode and programming/erasure priority mode) and two erasure modes (suspension priority mode and programming/erasure priority mode). The P/E resume command that resumes suspended programming/erasure processing is also provided.

(1) Suspension during Erasure

- Suspension priority mode: Erasure can be suspended one time per pulse*1.
- Programming/erasure priority mode: Suspension is performed after completion of one programming/erasure pulse*1 (see Figure 36.19).

(2) Suspension during Programming

- Suspension priority mode: Programming can be suspended one time per pulse*1.
- Programming/erasure priority mode: Suspension is performed after completion of one programming/erasure pulse*1.

Note 1. Programming/erasure is performed by generating multiple pulses for a single programming/erasure command.

36.7.1 Suspension during Programming/Erasure (Suspension Priority Mode)

Figure 36.19 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, the FSTATR0.SUSRDY bit is set 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 erasure suspend mode 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.

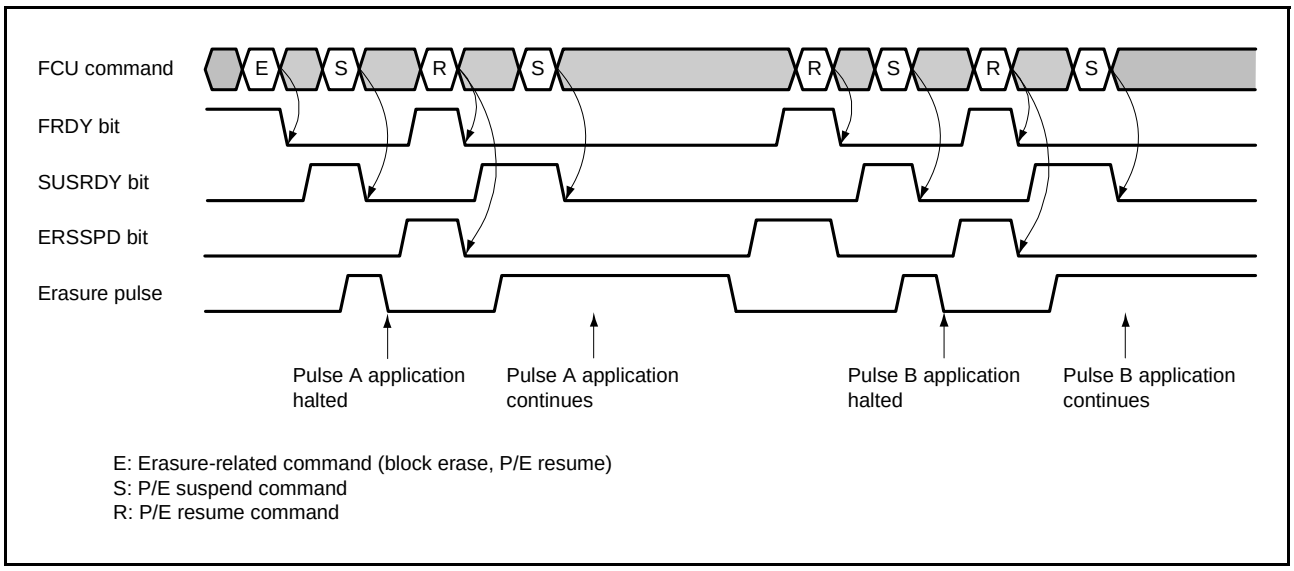


Figure 36.19 Suspension during Erasure (Suspension Priority Mode)

36.7.2 Suspension during Programming/Erasure (Programming/Erasure Priority Mode)

Figure 36.20 shows the suspend operation of erasure when the erasure suspend mode is set to the programming/erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in programming/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.

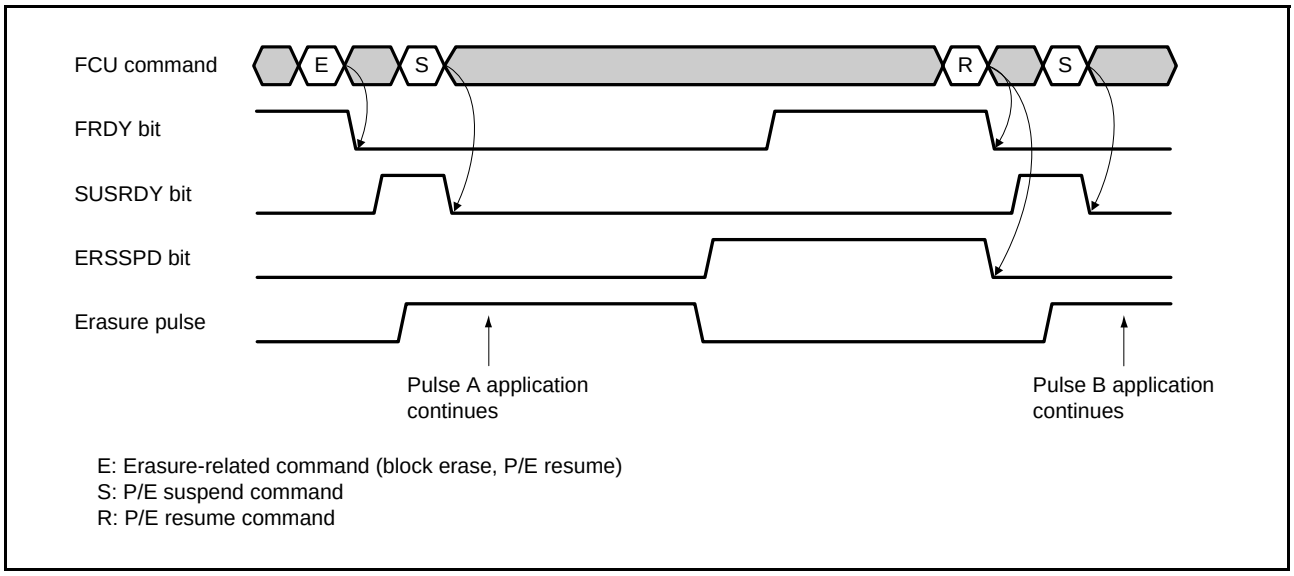


Figure 36.20 Suspension during Erasure (Programming/Erasure Priority Mode)

36.8 Protection

Protection against programming/erasure for the ROM includes software protection and command-locked state.

36.8.1 Software Protection

With the software protection, the ROM 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 programming/erasure-related command is issued, the FCU detects an error and enters the 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.FENTRY0 bit is 0, ROM read mode is selected. Because the FCU command cannot be received in ROM read mode, ROM programming/erasure is prohibited. When an FCU command is issued in ROM read mode, the FCU detects an illegal command error and is placed in the command-locked state (see section 36.8.2, Command-Locked State).

(3) Protection through Lock Bit

Each erasure block in the user area includes a lock bit. When the FPROTR.FPROTCN bit is 0, erasure blocks whose lock bit is set to 0 are prohibited from being programmed/erased. To program or erase erasure 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 enters the command-locked state (see section 36.8.2, Command-Locked State).

36.8.2 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 FASTAT.ROMAE bit) is set to 1, the FCU will be in the command-locked state (FASTAT.CMDLK bit is set to 1), so programming and erasure of the ROM are prohibited. To clear the command-locked state, a status register clear command must be issued with FASTAT set to 10h.
While the FAEINT.CMDLKIE bit is set to 1, if the FCU is placed in the command-locked state (FASTAT.CMDLK bit is set to 1), a flash interface error (FIFERR) interrupt occurs. While the FAEINT.ROMAEIE bit is set to 1, if the FASTAT.ROMAE bit is set to 1, an FIFERR interrupt occurs.
Table 36.8 lists the relationship between the contents of the ROM-related command-locked state and status bit values (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.ROMAE bit) 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 36.8 Errors that Lead to the Command-Locked State (Types Dedicated to ROM and Types Common to ROM and E2 DataFlash)

Type	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FENTRYR setting error	More than one bit is set to 1 among the FENTRYD and FENTRY0 bits in FENTRYR	1	0	0	0	0
	The FENTRYR setting at suspension disagrees with that at resumption	1	0	0	0	0
Illegal command error	Undefined code is specified in the first cycle of an FCU command	1	0	0	0	0
	Other than D0h is specified in the last cycle of a multi-cycle FCU command	1	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
	A command other than the suspend command is issued during programming/erasure	1	0	0	0	0
	A suspend command is issued during processing other than programming/erasure	1	0	0	0	0
	A suspend command is issued in the suspended state	1	0	0	0	0
	A resume command is issued in other than the suspended state	1	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
	A block erase command is issued in the erasure suspended state	1	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
	Other than 01h, 04h, or 40h is specified in the second cycle of a programming command	1	0	0	0	0
	A command is issued in the command-locked state	1	0/1	0/1	0/1	0/1
Erasure error	An error occurs during erasure	0	1	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
Programming error	An error occurs during programming	0	0	1	0	0
	When the FPROTR.FPROTCN bit is 0, a programming or lock bit programming command is issued to an erasure block whose lock bit is set to 0	0	0	1	0	0
FCU error	An error occurs during FCU internal processing	0	0	0	1	0
ROM access violation	When the FENTRYR.FENTRY0 bit is 1 in ROM P/E normal mode, a read access command is issued for addresses: 00FF 8000h to 00FF FFFFh when the user area capacity is 32 Kbytes, 00FF 0000h to 00FF FFFFh when the user area capacity is 64 Kbytes, 00FE 0000h to 00FF FFFFh when the user area capacity is 128 Kbytes, or 00FC 0000h to 00FF FFFFh when the user area capacity is 256 Kbytes.	1	0	0	0	1
	When the FENTRYR.FENTRY0 bit is 0, an access command is issued for addresses: 00FF 8000h to 00FF FFFFh when the user area capacity is 32 Kbytes, 00FF 0000h to 00FF FFFFh when the user area capacity is 64 Kbytes, 00FE 0000h to 00FF FFFFh when the user area capacity is 128 Kbytes, or 00FC 0000h to 00FF FFFFh when the user area capacity is 256 Kbytes.	1	0	0	0	1
	When FENTRYR has set ROM in ROM P/E mode, a read access command is issued for addresses FFFF 8000h to FFFF FFFFh when the user area capacity is 32 Kbytes, FFFF 0000h to FFFF FFFFh when the user area capacity is 64 Kbytes, FFFE 0000h to FFFF FFFFh when the user area capacity is 128 Kbytes, or FFFC 0000h to FFFF FFFFh when the user area capacity is 256 Kbytes.	1	0	0	0	1

36.9 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.

36.10 Boot Mode

36.10.1 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 RX220 in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.
When the RX220 is activated in boot mode, the program on the area that holds the boot program is executed. This program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.
Figure 36.21 shows the system configuration for operations in boot mode.
Input and output pins associated with the ROM are listed in Table 36.9.

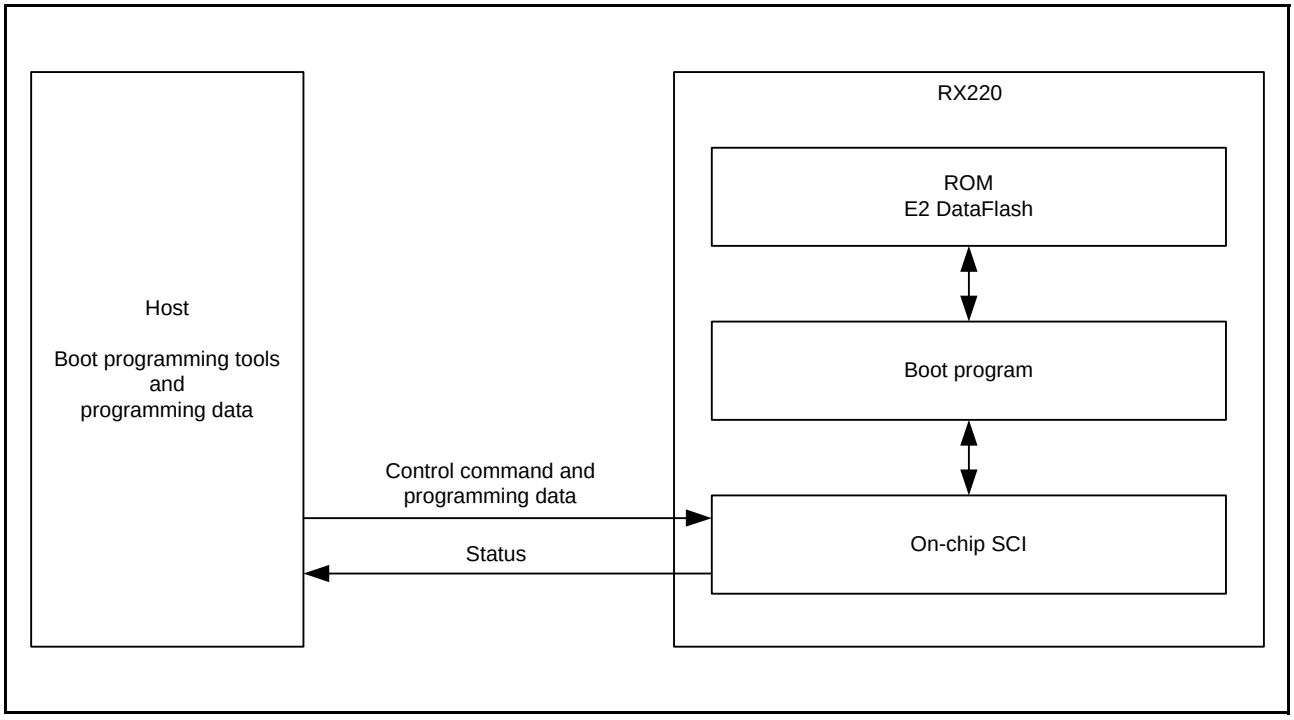


Figure 36.21 System Configuration for Operations in Boot Mode

Table 36.9 Input and Output Pins Associated with the ROM

Pin Name	I/O	Description
MD	Input	Selection of operating mode
PC7	Input	Selection of boot mode or User boot mode
P30/RXD1	Input	Used in boot mode to receive data via SCI1 (for host communications)
P26/TXD1	Output	Used in boot mode to transmit data from SCI1 (for host communications)

36.10.2 State Transitions in Boot Mode

Figure 36.22 is a diagram of the state transitions in boot mode.

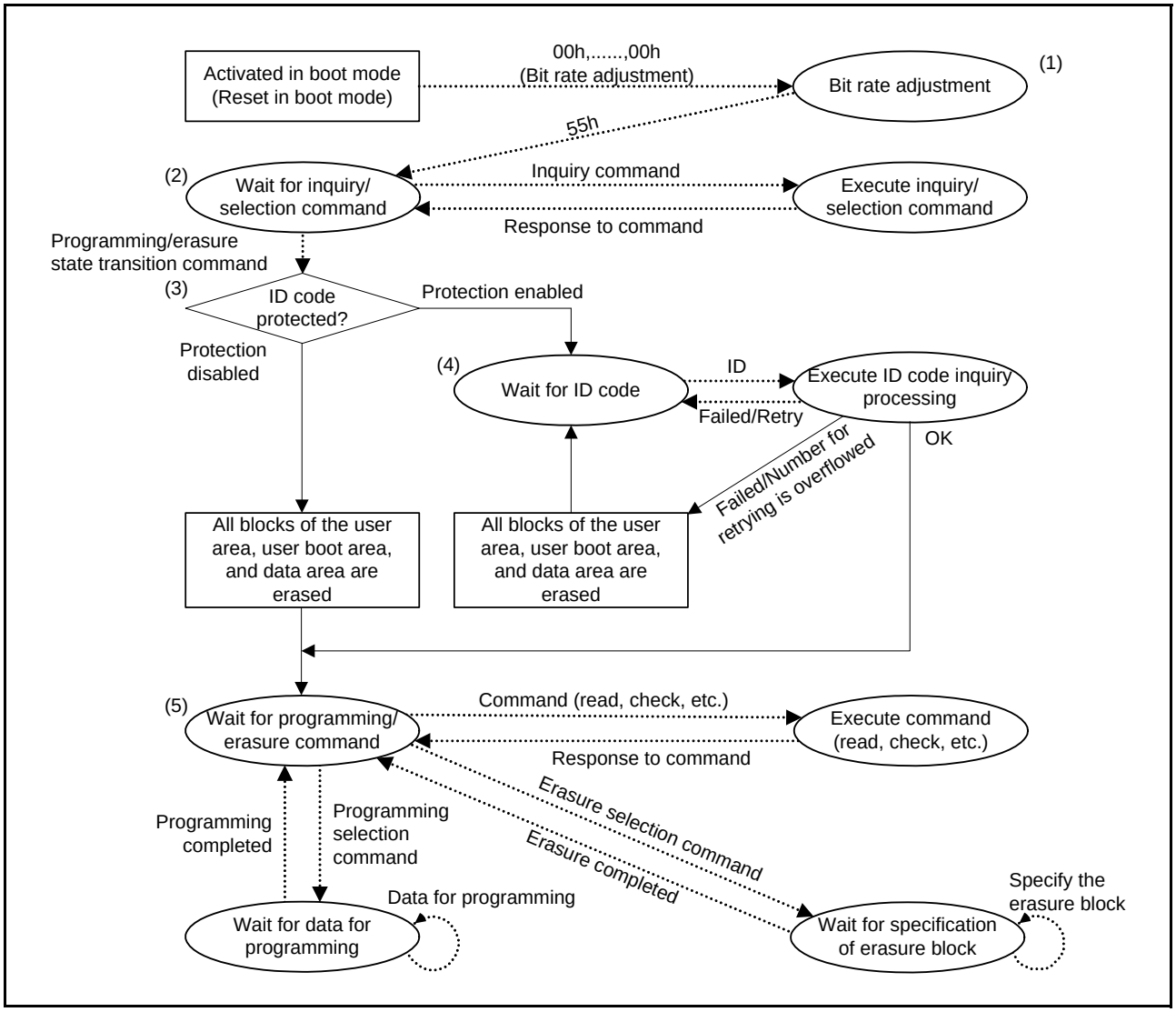


Figure 36.22 State Transitions in Boot Mode

(1) Matching the Bit Rates

When the RX220 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, the RX220 transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, the RX220 enters of wait for a command for inquiry or selection. For details on matching of the bit rates, see section 36.10.3, Automatic Adjustment of the Bit Rate.

(2) Waiting for a Command for Inquiry or Selection

This state is for inquiries on area size, area configuration, the addresses where areas start, the state of support, etc., and for selection of the device, clock mode, and bit rate. The RX220 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 36.10.6, 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 36.10.4, 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. A reset is required to release the system from this state due to non-matching ID codes. For details on the control code and ID code, see section 36.10.4, 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, the RX220 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 the RX220 receives a programming selection command, it enters the wait 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 the RX220 receives an erasure selection command, it enters the wait 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 (to confirm erasure), reading from memory, and acquiring status information.

36.10.3 Automatic Adjustment of the Bit Rate

When the RX220 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. The RX220 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 the RX220. If successful reception of 00h by the host is not possible, reboot the RX220 in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by the RX220 is successful, it responds by sending E6h to the host, and if successful reception of 55h by the RX220 is not possible, it responds by sending FFh to the host.

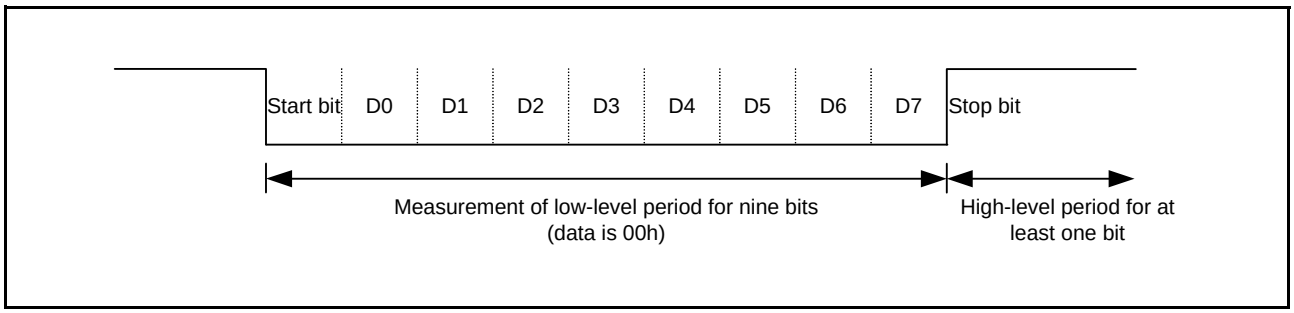


Figure 36.23 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

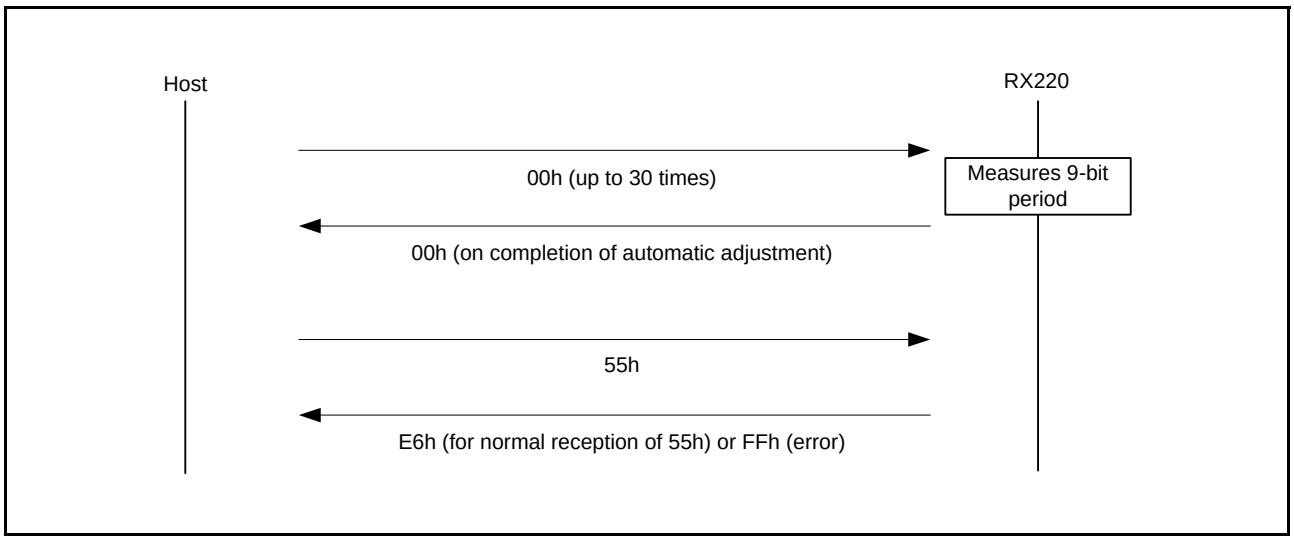


Figure 36.24 Sequence of Transfer between Host and RX220

Set the bit rate for SCI transfer on the host to satisfy either of the conditions Table 36.10.

Table 36.10 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host
9,600 bps
19,200 bps

36.10.4 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 36.25 shows the configuration of the control code and ID code. The ID code should be set in 32-bit units.

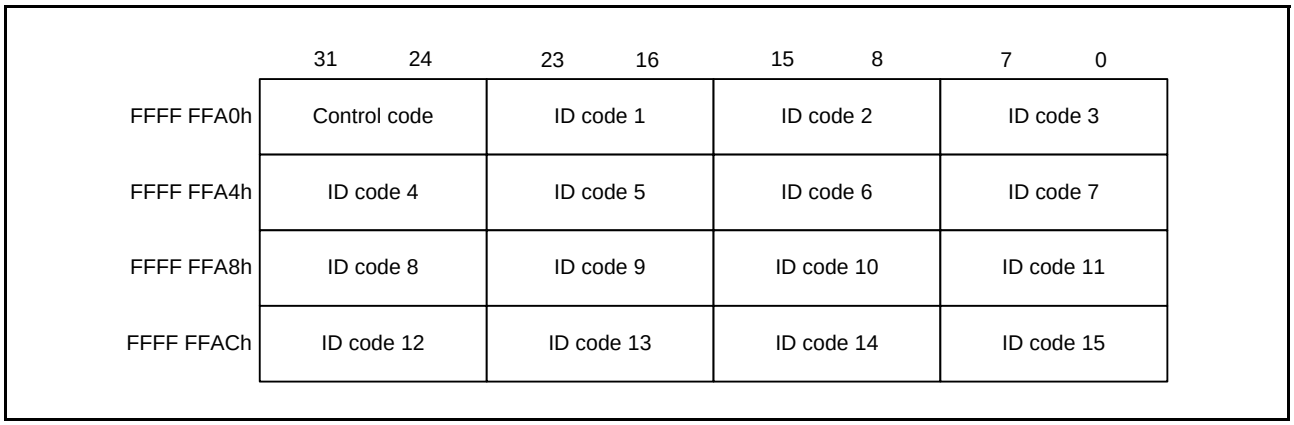


Figure 36.25 Configuration of Control Code and ID Code in ROM

(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 36.11 lists how the control code determines the method of authentication

Table 36.11 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 host 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)	Always 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 always 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
```

36.10.5 UB Code

For the UB code, see section 7.3, UB Code.

36.10.6 Inquiry/Selection Command Wait

Table 36.12 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 36.12 Inquiry/Selection Commands

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 areas and the start and end addresses
User area information inquiry	Inquires regarding the number of user areas and the start and end addresses
Erasure block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
New bit rate selection	Modifies the bit rate of SCI communications between the host and RX220
Programming/erasure state transition	Enters the state for determining ID code protection
Boot program status inquiry	Inquires regarding the processing state

If the host has sent an undefined command, the RX220 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

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 the RX220 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, the RX220 returns a response indicating a command error. Figure 36.26 shows an example of the procedure to use commands in the inquiry/selection command wait.

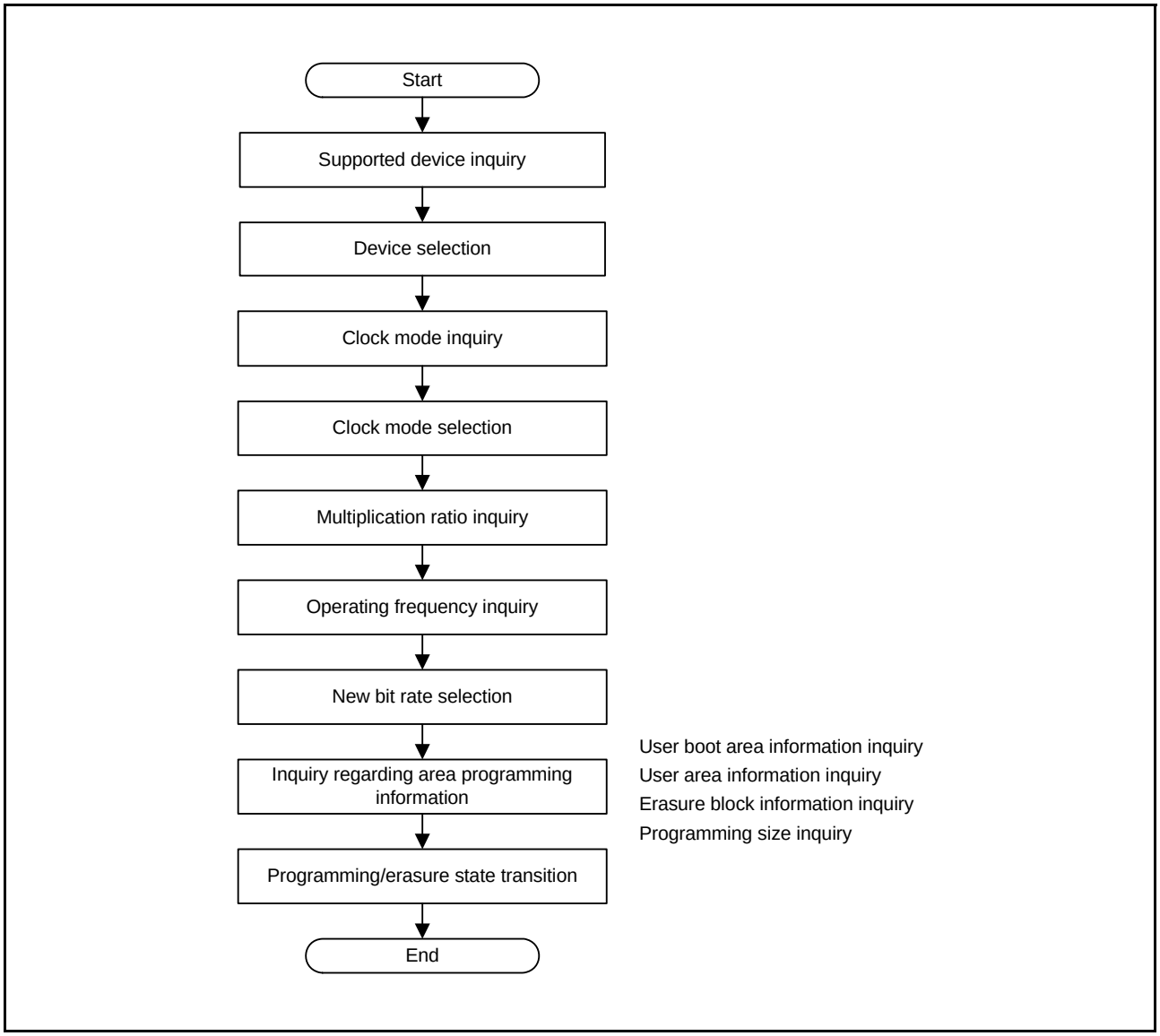


Figure 36.26 Example of Procedure to Use Inquiry/Selection Commands for User Area/User Boot Area

Each command is described in detail below. The “command” in the description indicates a command sent from the host to the RX220 and the “response” indicates a response sent from the RX220 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX220 becomes 00h.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, the RX220 returns the information concerning the devices supported by the boot program. If the supported device inquiry command comes after the host has selected a device, the RX220 only returns the information concerning the selected device. In response to supported device inquiry commands, the RX220 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
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, the RX220 checks if the selected device is supported. When the selected device is supported, the RX220 returns a response (06h). If the selected device is not supported or the sent command is illegal, the RX220 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, the RX220 returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, the RX220 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, the RX220 checks if the selected clock mode is supported. When the selected mode is supported, the RX220 specifies this clock mode for use and returns a response (06h). If the selected mode is not supported or the sent command is illegal, the RX220 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)
21h: Clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, the RX220 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, the RX220 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, the RX220 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

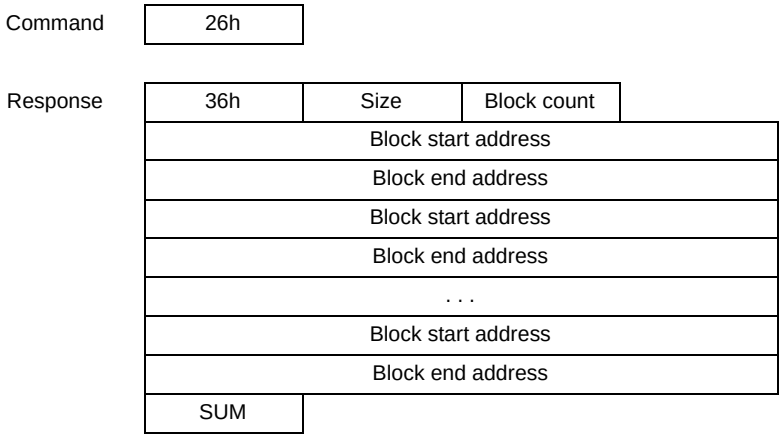
In response to a user area information inquiry command sent from the host, the RX220 returns the number of user areas and their addresses.

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) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, the RX220 returns the number of total erasure blocks in the user area and data area, and their addresses.



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, the RX220 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) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, the RX220 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, the RX220 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, the RX220 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 the RX220 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.

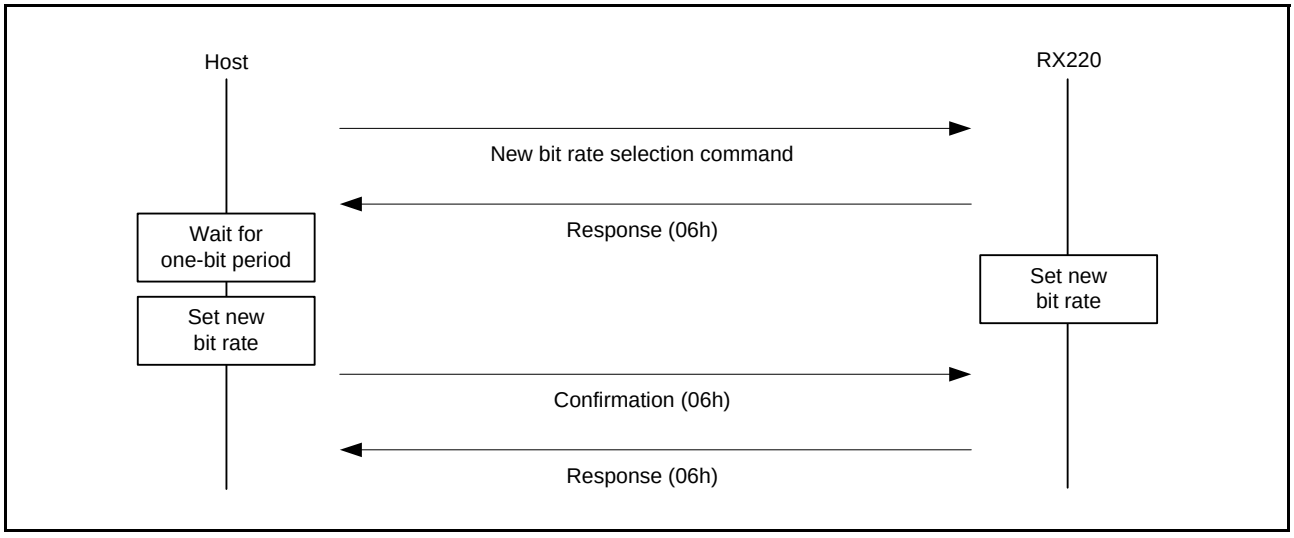


Figure 36.27 New Bit Rate Selection Sequence

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 the RX220 (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 (for example, 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 (PCLK) 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 the RX220 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 the RX220 cannot operate at the operating frequencies selected through a new bit rate selection command. The RX220 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.

(12) Programming/Erasure State Transition

In response to a programming/erasure state transition command sent from the host, the RX220 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, the RX220 returns a response (16h) and waits for the ID code. When ID code protection is disabled, the RX220 erases the entire area of each of the user area, user boot area and data area. After completing entire erasure, the RX220 returns a response (26h) and waits for a programming/erasure command. If the RX220 has failed to complete erasure due to an error, it returns an error response (C0h, 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

(13) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, the RX220 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	SUM

Size (1 byte): Total number of bytes in the status and error fields (fixed at 2)

Status (1 byte): Current status of RX220 (see Table 36.13)

Error (1 byte): Error status of RX220 (see Table 36.14)

SUM (1 byte): Checksum

Table 36.13 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 36.14 Error Code

Code	Description
00h	No error
11h	Checksum error
21h	Incorrect device code error
22h	Incorrect 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

36.10.7 ID Code Wait State

Table 36.15 shows the command available in the ID code wait state.

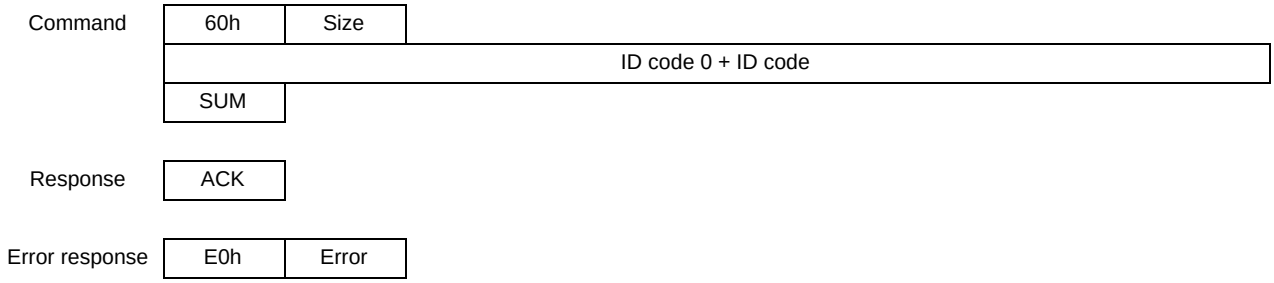
Table 36.15 ID Code Check Command

Command Name	Function
ID code check	Performs the ID code check

If the host has sent an undefined command, the RX220 returns a response indicating a command error. For the contents of a command error, see section 36.10.6, Inquiry/Selection Command Wait.

(1) ID Code Check

In response to an ID code check command sent from the host, the RX220 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.

36.10.8 Programming/Erasure Command Wait

Table 36.16 lists the commands available in the programming/erasure command wait.

Table 36.16 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
User boot area blank check	Checks whether the user boot area is blank
User area blank check	Checks whether the user 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 RX220

If the host has sent an undefined command, the RX220 returns a response indicating a command error. For the contents of a command error, see section 36.10.6, Inquiry/Selection Command Wait.

To program the ROM, issue a programming selection command (user/data area programming selection, user boot area programming selection) and then a 256-byte programming command from the host. Upon reception of a programming selection command, the RX220 enters the programming data wait state (see section 36.10.2, State Transitions in Boot Mode). In response to a 256-byte programming command sent from the host in this state, the RX220 starts programming the ROM. When the host sends a 256-byte programming command specifying FFFF FFFFh as the programming start address, the RX220 detects it as the end of programming and enters the programming/erasure command wait.

To erase the ROM, issue an erasure selection command and then a block erase command from the host. Upon reception of an erasure selection command, the RX220 enters the erasure block selection wait state (see section 36.10.2, State Transitions in Boot Mode). In response to a block erase command sent from the host in this state, the RX220 erases the specified block in the ROM. When the host sends a block erase command specifying FFh as the block number, the RX220 detects it as the end of erasure and enters the programming/erasure command wait.

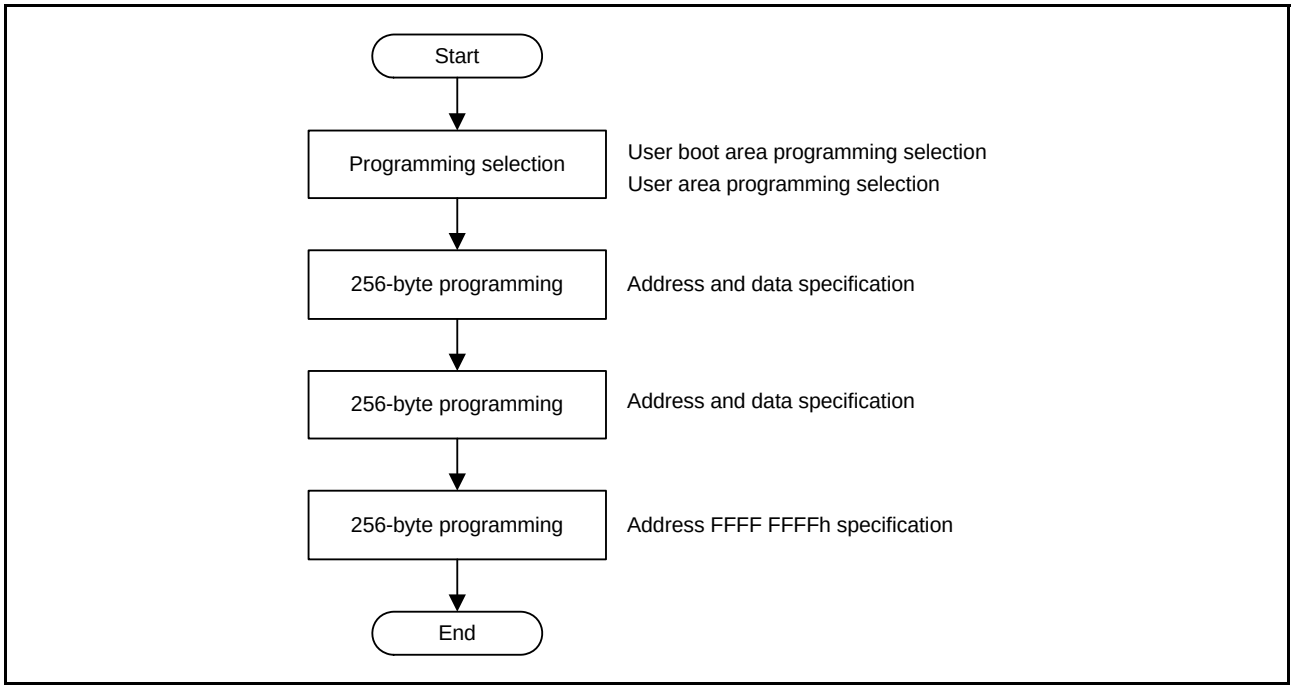


Figure 36.28 Procedure for ROM Programming in Boot Mode

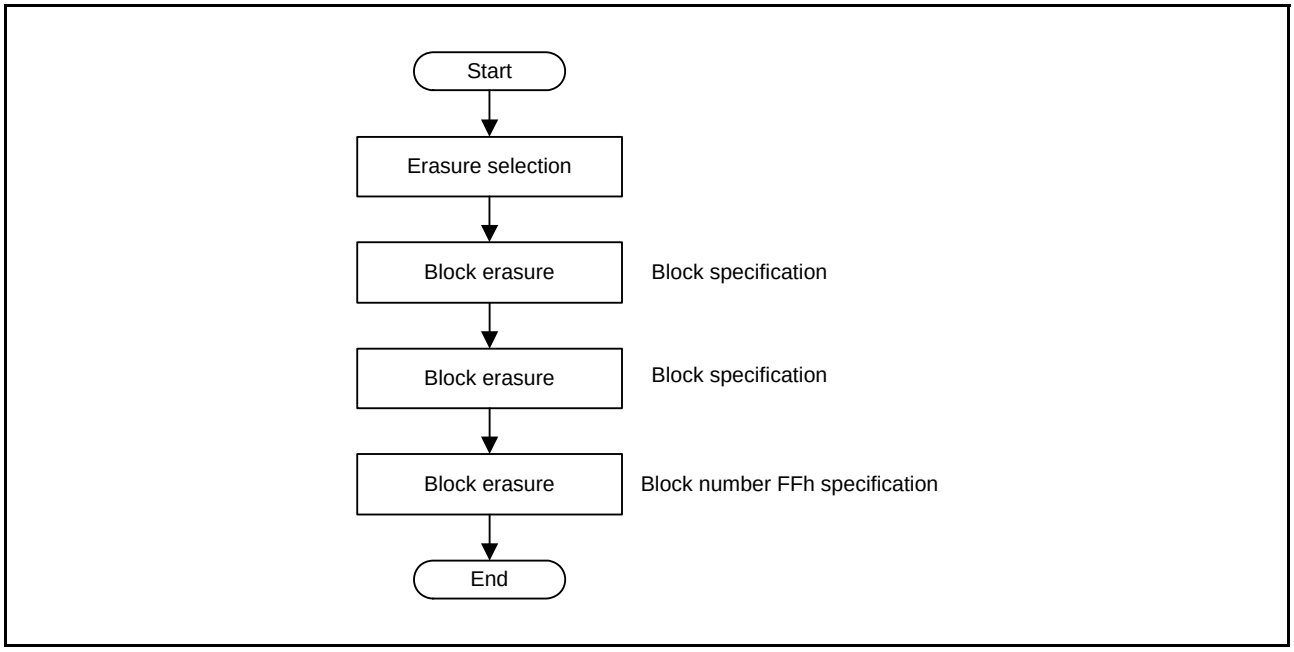


Figure 36.29 Procedure for ROM Erasure in Boot Mode

Each command is described in detail below. The “command” in the description indicates a command sent from the host to the RX220 and the “response” indicates a response sent from the RX220 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX220 becomes 00h.

(1) User Boot Area Programming Selection

In response to a user boot area programming selection command sent from the host, the RX220 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, the RX220 selects the program for user area programming and waits for programming data.

Command	43h
Response	06h

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, the RX220 programs the ROM. After completing ROM programming successfully, the RX220 returns a response (06h). If an error has occurred during ROM programming, the RX220 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 program the ROM, 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, the RX220 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, the RX220 erases the ROM. After completing ROM erasure successfully, the RX220 returns a response (06h). If an error has occurred during ROM erasure, the RX220 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 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, the RX220 reads data from the ROM. After completing ROM reading successfully, the RX220 returns the data stored in the address specified by the memory read command. If the RX220 has failed to read the ROM, the RX220 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
Read start address (4 bytes):	Start address of the area to be read
Reading size (4 bytes):	Size of data to be read (bytes)
SUM (1 byte):	Checksum
Data (1 byte):	Data read from the ROM
Error (1 byte):	Error code 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, the RX220 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, the RX220 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) User Boot Area Blank Check

In response to a user boot area blank check command sent from the host, the RX220 checks whether the user boot area is completely erased. When the user boot area is completely erased, the RX220 returns a response (06h). If the user boot area has an unerased area, the RX220 returns an error response (CCh, 52h).

Command	4Ch	
Response	06h	
Error response	CCh	52h

(10) User Area Blank Check

In response to a user area blank check command sent from the host, the RX220 checks whether the user area is completely erased. When the user area is completely erased, the RX220 returns a response (06h). If the user area has an unerased area, the RX220 returns an error response (CDh, 52h).

Command	4Dh	
Response	06h	
Error response	CDh	52h

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, the RX220 reads data from the lock bit. After completing the lock bit reading successfully, the RX220 returns the data stored in the address specified by the read lock bit status command. If the RX220 has failed to read the lock bit, the RX220 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
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 15 to 8)
A23 to A16 (1 byte):	A23 to A16 of the last address in the specified block (bits 23 to 16)
A31 to A24 (1 byte):	A31 to A24 of the last address in the specified block (bits 31 to 24)
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)

(12) Lock Bit Program

In response to a lock bit program command sent from the host, the RX220 writes to a lock bit and locks the specified block. After completing the lock bit blocking successfully, the RX220 returns a response (06h). If the RX220 has failed to lock, the RX220 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 the RX220)
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

(13) Lock Bit Enable

In response to a lock bit enable command sent from the host, the RX220 enables a lock bit.

Command	7Ah
Response	06h

(14) Lock Bit Disable

In response to a lock bit disable command sent from the host, the RX220 disables a lock bit.

Command	75h
Response	06h

(15) Boot Program Status Inquiry

For details, refer to section 36.10.6, Inquiry/Selection Command Wait.

36.11 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 always 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 always considered to match, and connection of the on-chip debugger is allowed. See Figure 36.25 for the configuration of ID codes in flash memory.

Table 36.17 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 always matches and connection to the on-chip debugger is permitted.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h	Protection enabled	The ID code is always 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.

36.12 Usage Notes

(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 or Erasure

In cases where the input of a reset on the RES# pin during programming or erasure is not avoidable, only deassert the reset signal after at least the reset input period (tRESWF; see section 38, Electrical Characteristics) has elapsed. When the FCU is reset by using the FRESETR.FRESET bit during programming or erasure, make sure that the reset state is maintained over tFCUR (see section 38, Electrical Characteristics). Do not attempt to read the ROM during the FCU reset.
Maintaining the periods described above is not relevant to usage in the case of IWDT, or software reset during programming or erasure.

(5) Prohibition of Non-Maskable Interrupts during Programming or Erasure

If a non-maskable interrupt (an NMI pin, oscillation stop detection, IWDT underflow/refresh error, voltage monitoring 1, or voltage monitoring 2 interrupt) is generated during programming or erasure, the vector read is undefined because it is fetched from the ROM. Therefore, avoid the generation of non-maskable interrupts during programming and erasure of the ROM (this is only applicable in the case of the ROM).

(6) Interrupt Vector Assignment During Programming or Erasure

The generation of interrupts during programming or erasure 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/Erasure in Low-Speed Operating Modes 1 and 2

Do not program or erase the flash memory when low-speed operating mode 1 or 2 is selected with the operating power control register (OPCCR).

(8) Programming/Erasure Abnormal End

If, during programming or erasure, the operating voltage swings above the allowed range, the FCU is reset by using the FRESETR.FRESET bit, the command-locked state is entered due to the detection of an error, or one of the prohibitions under item (9) below is violated, and programming or erasure is not completed normally as a consequence, the value of the lock bit may become 0 (corresponding to the protected state). In this case, erase the lock bit by issuing the block erase command with the FPROTR.FPROTCN bit set to 1. After that, repeat the incomplete programming so that it is completed normally.

(9) Actions Prohibited during Programming and Erasure

Since they may damage the flash memory, avoid all of the actions, events, and operations listed below during programming and erasure.

- Applying a power-supply voltage beyond the allowed range to the RX220
- Changing the values of the FWEPROR.FLWE[1:0] bits
- Changing the values of the OPCCR.OPCM[2:0] bits
- Changing the clock source select bits in the SCKCR3 register
- Enabling the sleep mode return clock source switching function by setting the RSTCKCR.RSTCKEN bit
- Changing the division ratio of the FlashIF clock (FCLK)
- Setting the PCKAR for a different frequency from that of FCLK
- Making a transition to all-module clock stop mode or software standby mode

37. E2 DataFlash Memory (Flash Memory for Data Storage)

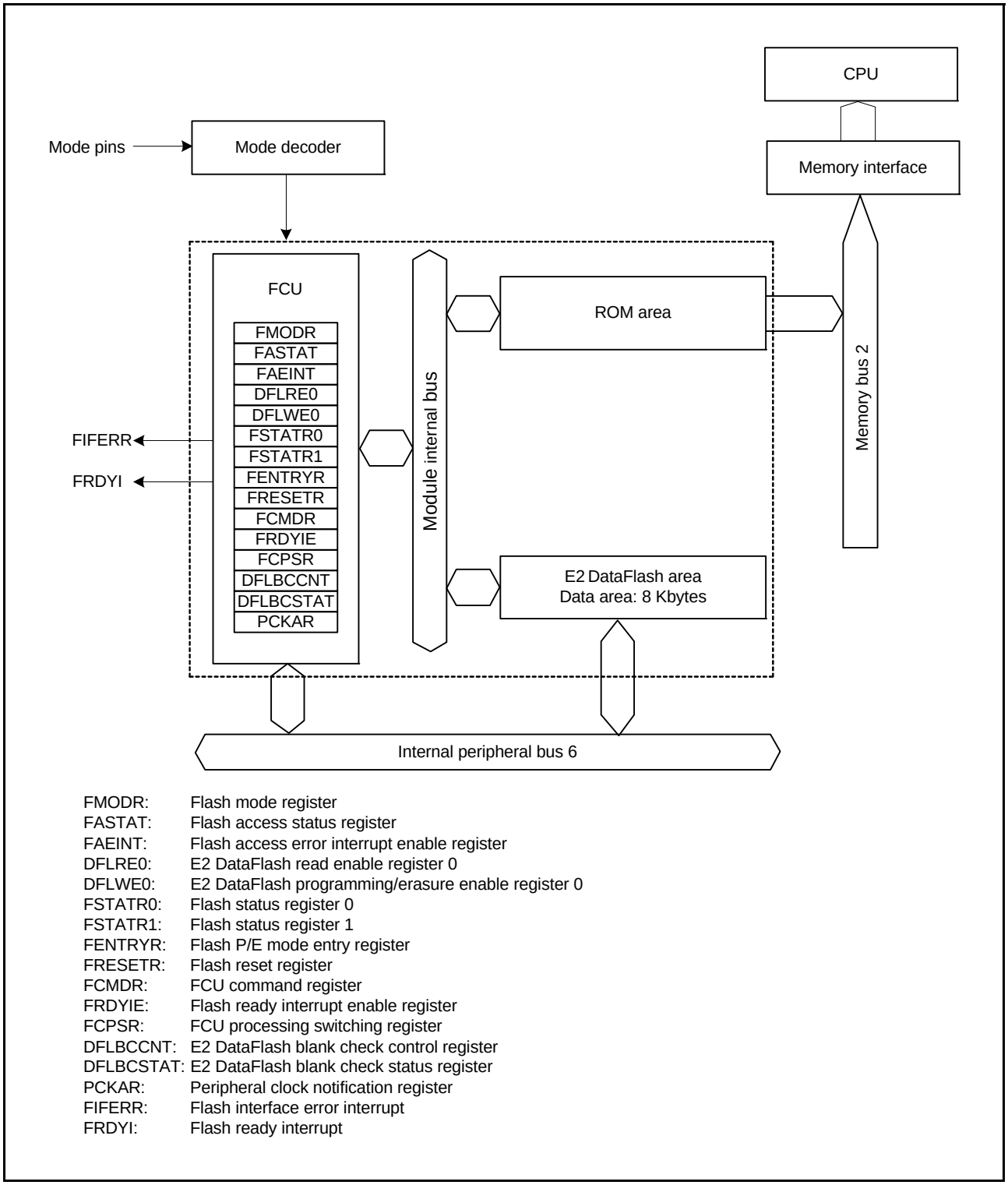
The RX220 Group has an 8-Kbyte flash memory for storing data (E2 DataFlash).
This section covers the E2 DataFlash memory. For the ROM, see section 36, ROM (Flash Memory for Code Storage).

37.1 Overview

Table 37.1 lists the specifications of the E2 DataFlash memory, and Figure 37.1 is a block diagram of the ROM, E2 DataFlash memory, and related modules.

Table 37.1 Specifications of E2 DataFlash Memory

Item		Description
Memory capacity		Data area: 8 Kbytes
Reading via the peripheral bus		A read operation takes four cycles of FCLK in words or bytes
Programming/erasing method		• The chip incorporates a dedicated sequencer (FCU) for programming of the ROM and E2 DataFlash. • Programming and erasing the ROM and E2 DataFlash are handled by issuing commands to the FCU.
BGO (background operation)		• Execution of program code from the ROM is possible while the E2 DataFlash memory is being programmed or erased. • The CPU is able to execute program code from areas other than the ROM or E2 DataFlash while the ROM is being programmed or erased.
Suspension and resumption		• The CPU is able to execute program code from the E2 DataFlash during suspension of programming or erasure. • Programming and erasure of the E2 DataFlash can be restarted (resumed) after suspension.
Units of programming and erasure		• Unit of programming for the data area: 2 or 8 bytes • Unit of erasure for the data area: 128 bytes
Blank checking function		• The blank checking command can be executed to check the erasure state of E2 DataFlash. • The size of the area to be blank-checked is 2 bytes or 2 Kbytes.
On-board programming (three types)		Reprogramming in boot mode • The clock synchronous serial interface (SCI1) is used. • The bit rate is automatically adjusted. • The user boot area is also programmable.
		Reprogramming in user boot mode • The user-specific boot program can be programmed.
		Reprogramming using the ROM reprogramming routine in the user program • ROM is reprogrammable without resetting the system.
Protection	Software-controlled protection	• The FENTRYR.FENTRYD bit, FWEPROR.FLWE[1:0] bits, DFLRE0, and DFLWE0 register can be used to prevent unintentional programming or reading. • Protection with the DFLRE0 and DFLWE0 registers is performed on a 2-Kbyte basis.
	Command-locked state	Prevention of further programming or erasure after the detection of abnormal operations during programming or erasure
Programming/erasure time, number of programming		See section 38, Electrical Characteristics.



FMODR: Flash mode register
FASTAT: Flash access status register
FAEINT: Flash access error interrupt enable register
DFLRE0: E2 DataFlash read enable register 0
DFLWE0: E2 DataFlash programming/erasure enable register 0
FSTATR0: Flash status register 0
FSTATR1: Flash status register 1
FENTRYR: Flash P/E mode entry register
FRESETR: Flash reset register
FCMDR: FCU command register
FRDYIE: Flash ready interrupt enable register
FCPSR: FCU processing switching register
DFLBCCNT: E2 DataFlash blank check control register
DFLBCSTAT: E2 DataFlash blank check status register
PCKAR: Peripheral clock notification register
FIFERR: Flash interface error interrupt
FRDYI: Flash ready interrupt

Figure 37.1 Block Diagram of E2 DataFlash Memory

For input and output pins associated with the E2 DataFlash, see Table 36.9, Input and Output Pins Associated with the ROM.

37.2 Register Descriptions

Some registers also have bits related to the ROM, but this section deals only with the bits that are relevant to the E2 DataFlash. For registers containing bits with common functions for the ROM and E2 DataFlash (FRDYIE, FSTATR0, FSTATR1, FRESETR, FCMDR, FCPSR, PCKAR, and FWEPROR) and details on the functions of bits dedicated to the ROM, see section 36.2, Register Descriptions.
P/E indicates programming/erasure.

37.2.1 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 Mode Select	0: Memory Area Reading Method This is the setting when a transition of E2 DataFlash lock bit read mode is made. 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 reading mode is made. 1: Register Reading Method This is the setting when the blank checking command is to be used.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set the FRDMD bit to 1 if the blank checking command is to be used.

FRDMD Bit (FCU Read Mode Select)

This bit is used to select processing for a transition of E2 DataFlash lock bit read mode or for blank checking.
The FRDMD bit is used to select the method of reading when lock bit values for the ROM are read (see section 36, ROM (Flash Memory for Code Storage)).

37.2.2 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	0: An E2 DataFlash programming/erasure command is not issued which conflicts with the DFLWE0 settings 1: An E2 DataFlash programming/erasure command is issued which conflicts with the DFLWE0 settings	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation	0: There is no such E2 DataFlash reading that conflicts with the DFLRE0 settings 1: There is such an E2 DataFlash reading that conflicts with the DFLRE0 settings	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	0: No E2 DataFlash access violation 1: E2 DataFlash access violation	R/(W) *1
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation	See section 36, ROM (Flash Memory for Code Storage).	R/(W) *1

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

When any bit from among the DFLWPE, DFLRPE, DFLAE, and ROMAE bits in FASTAT is set to 1, the FSTATR0.ILGLERR bit is set to 1, which places the FCU in the command-locked state (see section 37.7.2, Command-Locked State). To clear the command-locked state, a status register clearing command must be issued to the FCU after setting FASTAT to 10h.

DFLWPE Bit (E2 DataFlash Programming/Erasure Protection Violation)

This bit is used to indicate whether or not the programming/erasure protection set by DFLWE0 is violated.
When the DFLWPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, placing the FCU in the command-locked state. For FSTATR0, see section 36.2.5, Flash Status Register 0 (FSTATR0).

[Setting condition]

- A programming/erasure command is issued for an E2 DataFlash area for which programming or erasure is disabled by DFLWE0.

[Clearing condition]

- When 0 is written after reading 1

DFLRPE Bit (E2 DataFlash Read Protection Violation)

This bit is used to indicate whether or not the reading protection set by DFLRE0 is violated.
When the DFLRPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, placing the FCU in the command-locked state.
For FSTATR0, see section 36.2.5, Flash Status Register 0 (FSTATR0).

[Setting condition]

- A read command is issued for an E2 DataFlash area for which reading is disabled by DFLRE0.

[Clearing condition]

- When 0 is written after reading 1

DFLAE Bit (E2 DataFlash Access Violation)

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, placing the FCU in the command-locked state.
For FSTATR0, see section 36.2.5, Flash Status Register 0 (FSTATR0).

[Setting conditions]

- A read command is issued for an E2 DataFlash area in E2 DataFlash P/E normal mode and when the FENTRYD bit in FENTRYR is set to 1.
- A write command is issued for an E2 DataFlash area when the FENTRYD bit is set to 0.
- A command is issued for an E2 DataFlash area when the FENTRY0 in FENTRYR is set to 1.

[Clearing condition]

- When 0 is written after reading 1

CMDLK Bit (FCU Command Lock)

This bit indicates that the FCU is in the command-locked state (see section 37.7.2, Command-Locked State).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU has processed a status register clearing command

37.2.3 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	0: FIFERR interrupt requests disabled when the DFLWPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLWPE bit in FASTAT is set to 1	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLRPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLRPE bit in FASTAT is set to 1	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	0: FIFERR interrupt requests disabled when the DFLAE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLAE bit in FASTAT is set to 1	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the CMDLK bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the CMDLK bit in FASTAT is set to 1	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	See section 36, ROM (Flash Memory for Code Storage).	R/W

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 a FCU command-locked state occurs and the CMDLK bit in FASTAT is set to 1.

37.2.4 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]								—	—	—	—	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	DB00 to DB15 Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE01	DB16 to DB31 Block Read Enable		R/W
b2	DBRE02	DB32 to DB47 Block Read Enable		R/W
b3	DBRE03	DB48 to DB63 Block Read Enable		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control 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 DB00 to DB63 blocks of the data area (see Figure 37.3) to be read.

DBREj Bit (DBj Block Read Enable) (j = 00 to 03)

This bit is used to enable or disable the DB00 to DB63 blocks of the data area to be read.

The DBREj bit is used to control reading of the DB00 to DB63 blocks.

37.2.5 E2 DataFlash Programming/Erasure 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]								—	—	—	—	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	DB00 to DB15 Block Programming/ Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE01	DB16 to DB31 Block Programming/ Erasure Enable		R/W
b2	DBWE02	DB32 to DB47 Block Programming/ Erasure Enable		R/W
b3	DBWE03	DB48 to DB63 Block Programming/ Erasure Enable		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control 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 DB00 to DB63 blocks of the data area (see Figure 37.3) to be programmed or erased.

DBWEj Bit (DBj Block Programming/Erasure Enable) (j = 00 to 03)

This bit is used to enable or disable the DB00 to DB63 blocks of the data area to be programmed or erased.
The DBWEj bit is used to control programming/erasure of the DB00 to DB63 blocks.

37.2.6 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	—	—	—	—	—	—	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	See section 36, ROM (Flash Memory for Code Storage).	R/W
b6 to b1	—	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 or FENTRY0 bit must be set to 1. If more than one bit is set to 1, the ILGLERR bit is set in FSTATR0 and the FCU enters the command-locked state.
After writing to the FENTRYR to initiate a transition to ROM-reading mode, read the register and confirm that it actually holds the new setting before proceeding to read the ROM.
FENTRYR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.
For FSTATR0, see section 36.2.5, Flash Status Register 0 (FSTATR0).
For FRESETR, see section 36.2.10, Flash Reset Register (FRESETR).

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)]

- 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.

37.2.7 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 initialized by a reset, or when the FRESET bit in FRESETR is set to 1.
For FRESETR, see section 36.2.10, Flash Reset Register (FRESETR).

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).
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.

37.2.8 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 initialized by a reset, or when the FRESET bit in FRESETR is set to 1.
For FRESETR, see section 36.2.10, Flash Reset Register (FRESETR).

37.3 Configuration of Memory Area for the E2 DataFlash Memory

The E2 DataFlash memory of products in the RX220 Group is configured as an 8-Kbyte data area. Figure 37.2 shows the configuration of the data area for the E2 DataFlash.
Note that for the data area, the address range for reading is the same as the address range for programming and erasure.

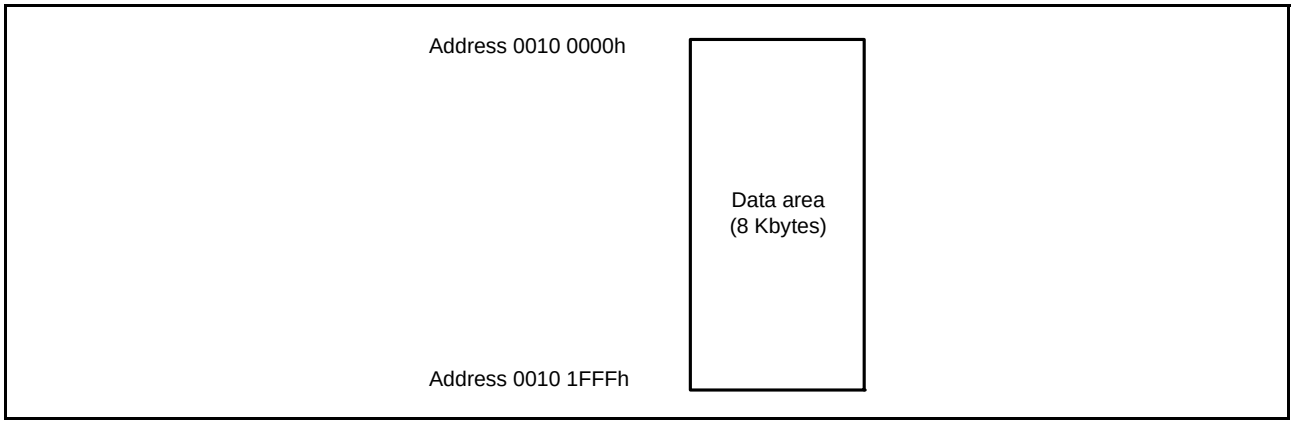


Figure 37.2 Configuration of the Data Area for the E2 DataFlash Memory

37.4 Block Configuration

Figure 37.3 shows how the erasure blocks of the data area are configured. The data area is divided into 64 blocks of 128 bytes, and erasure proceeds in these block units. Programming proceeds in 2- or 8-byte units. For programming in 2-byte units, each unit starts at an address whose two low-order bits are 0h or 2h. For programming in 8-byte units, each unit starts at an address whose three low-order bits are all 0.

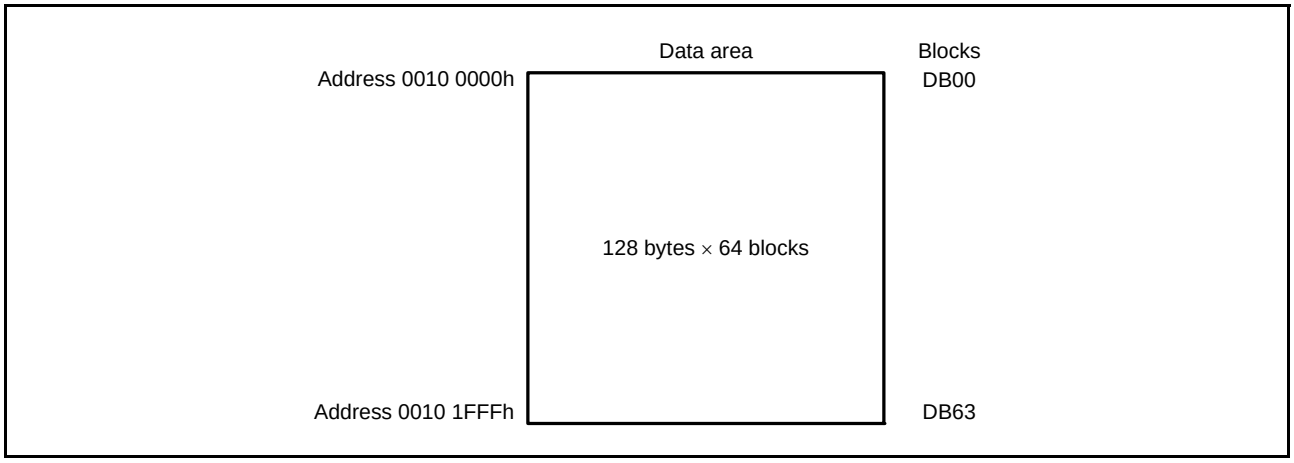


Figure 37.3 Division of the Data Area into Blocks

37.5 Operating Modes Associated with the E2 DataFlash

For the transitions between operating modes, see section 36.5, Operating Modes Associated with the ROM.
Reading, programming, and erasing of the data area in an on-board device can proceed if the device is in boot mode, user boot mode, or single-chip mode.
The differences between modes are indicated in Table 37.2.

Table 37.2 Differences between Modes

Item	Boot Mode	User Boot Mode	Single-Chip Mode
Environment for programming and erasure	On-board programming		
Programmable and erasable area	Data area	Data 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. All flash memory areas may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 36.10.4, ID Code Protection (Boot Mode), section 36.10.2, State Transitions in Boot Mode

- In boot mode, a host is able to program/erase or read the data area via an SCI.
- In boot mode, RAM is employed for use in the boot program. For this reason, preserving the contents of RAM is not possible in this case.
- In user boot mode, any desired interface can be used for programming or reading of the user area and data area following booting up from the user boot area.

37.6 Programming and Erasing the E2 DataFlash Memory

The E2 DataFlash memory is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.
The mode transitions required to program or erase the E2 DataFlash and the system of commands are described below. The descriptions apply in common to boot mode, user boot mode, and single-chip mode.

37.6.1 FCU Modes

The FCU has five modes or sets of modes. Transitions between modes are caused by writing to the FENTRYR register or issuing FCU commands. Figure 37.4 is a diagram of the FCU mode transitions.

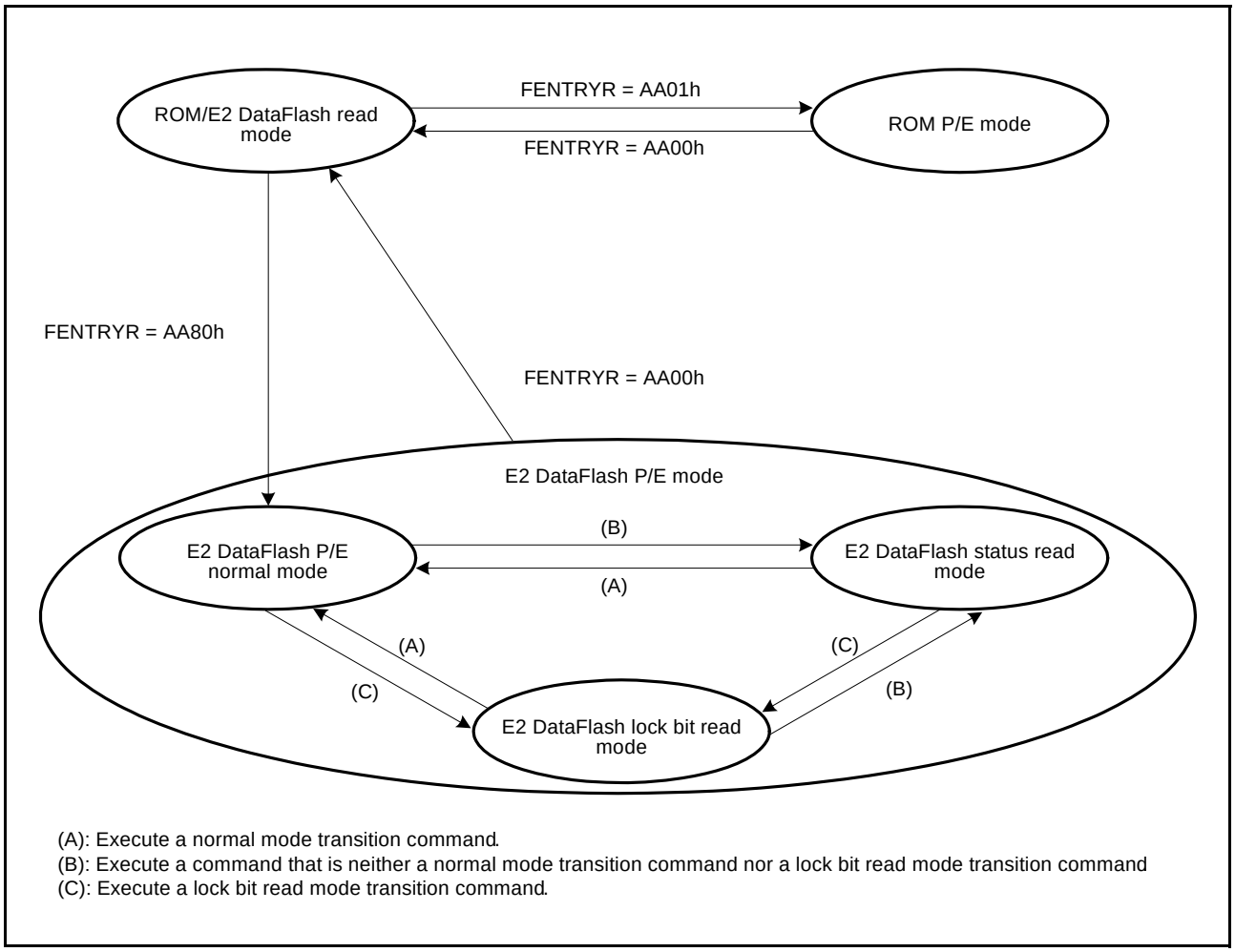


Figure 37.4 Mode Transitions of the FCU (Associated with the E2 DataFlash)

37.6.1.1 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM.
For details on the ROM P/E modes, see section 36.6.1.2, ROM P/E Modes.

37.6.1.2 ROM/E2 DataFlash Read Mode

This mode is for reading the ROM or E2 DataFlash memory. The FCU does not accept commands.
The FCU enters this mode when the FENTRYD and FENTRY0 bits in FENTRYR are set to 0.

37.6.1.3 E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. Reading out the E2 DataFlash is not possible.
E2 DataFlash P/E normal mode, E2 DataFlash status read mode, and E2 DataFlash lock-bit read mode are the 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 FENTRYD bit in FENTRYR is set to 1 and the FENTRY0 bit in FENTRYR is 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 37.5 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 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 command-locked state after an error has occurred.Table 37.5 lists the acceptable commands in this mode.
Read access to 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

The E2 DataFlash lock-bit read mode is for reading the values of the lock bits of the E2 DataFlash. However, this is not possible because the E2 DataFlash does not have lock bits.
The FCU enters this mode when a lock-bit read mode transition command is received in E2 DataFlash P/E modes. Table 37.5 lists the acceptable commands in this mode.
Since the E2 DataFlash does not have lock bits, data read in read access to addresses within the E2 DataFlash area are undefined. However, the access does not lead to an E2 DataFlash-access violation. High-speed reading of the ROM is possible.

37.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and of commands for programming and erasure. Table 37.3 lists the FCU commands for use with the E2 DataFlash.

Table 37.3 FCU Commands for Use with E2 DataFlash Memory

Command	Description
P/E normal mode transition	Changes the mode to normal mode (see section 37.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Changes the mode to status read mode (see section 37.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Changes the mode to lock bit read mode (see section 37.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the FlashIF clock (FCLK)
Programming	E2 DataFlash programming (in 2 or 8 byte units)
Block erasure	E2 DataFlash erasure (in 128 units)
P/E suspension	Suspends programming/erasure
P/E resumption	Resumes programming/erasure
Status register clearing	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and releases the FCU from the command-locked state
Lock bit read 2/blank checking	Checks whether the specified block of E2 DataFlash memory has been erased (is blank)

Commands other than the blank-checking command are also for use with the ROM.
The blank-checking command for the E2 DataFlash memory is also used as the lock bit read 2 command for the ROM. That is, when the same command is issued for the ROM, a lock bit of the ROM is read.
Commands for the FCU are issued by write access to addresses within the E2 DataFlash area.
Table 37.4 lists the formats of the programming commands and the blank checking command. For the formats of FCU commands other than programming and blank checking commands, see section 36.6.2, FCU Commands.
Write access as listed in Table 37.4 and in accord with certain conditions causes the FCU to execute processing for the corresponding command. For details on the conditions for acceptance of the individual FCU commands, see section 37.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 37.6.4, FCU Command Usage.

Table 37.4 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory)

Command	Number of Bus Cycles	First Cycle		Second Cycle		Third Cycle		4th to (N+2)th Cycle		(N+3)th Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Programming (2-byte programming; N = 1)	4	EA	E8h	EA	01h	WA	WDn	—	—	EA	D0h
Programming (8-byte programming; N = 4)	7	EA	E8h	EA	04h	WA	WDn	EA	WDn	EA	D0h
Blank checking	2	EA	71h	BA	D0h	—	—	—	—	—	—

Address column EA: Address within the E2 DataFlash area. Any address in the range from 0010 0000h to 0010 1FFFh.
WA: Start address of programming data
BA: Block start address within the E2 DataFlash area (2-Kbyte block)
Data columns WDn: nth word of data for programming (n = 1 to N)

37.6.3 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 varies according to 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 37.5. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 37.7.2, Command-Locked State).
Issuing of an FCU command must follow checking of the values of the acceptable FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FCUERR bit in FSTATR1 after transitions of the FCU mode. Furthermore, the CMDLK bit in FASTAT can be checked to see if an error has occurred. The value of the CMDLK bit in FASTAT is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1.

Table 37.5 Acceptable Commands and the State and Mode (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 during Erasure Suspended State	Processing to Suspend programming or Erasure	Blank Checking	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	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
FSTATR0.ERSSPD bit	0	1	0	0	1	0/1	0	0	1	0/1	0/1	0	0	1	0
FSTATR0.PRGSPD bit	1	0	0	0	0	0/1	0	1	0	0/1	0/1	0	1	0	0
FASTAT.CMDLK bit	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P/E normal mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock-bit read mode transition (lock bit read 1)	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
Programming	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Block erasure	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspension	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×
P/E resumption	A	A	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clearing	A	A	A	×	×	×	×	A	A	×	A	A	A	A	A
Blank checking	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

37.6.4 FCU Command Usage

This section shows how to program and erase the E2 DataFlash memory by using programming and block erasure commands, respectively, and how to check the state of erasure of the E2 DataFlash by using the blank check command. For the ways to use other FCU commands, see section 36.6.4, FCU Command Usage.

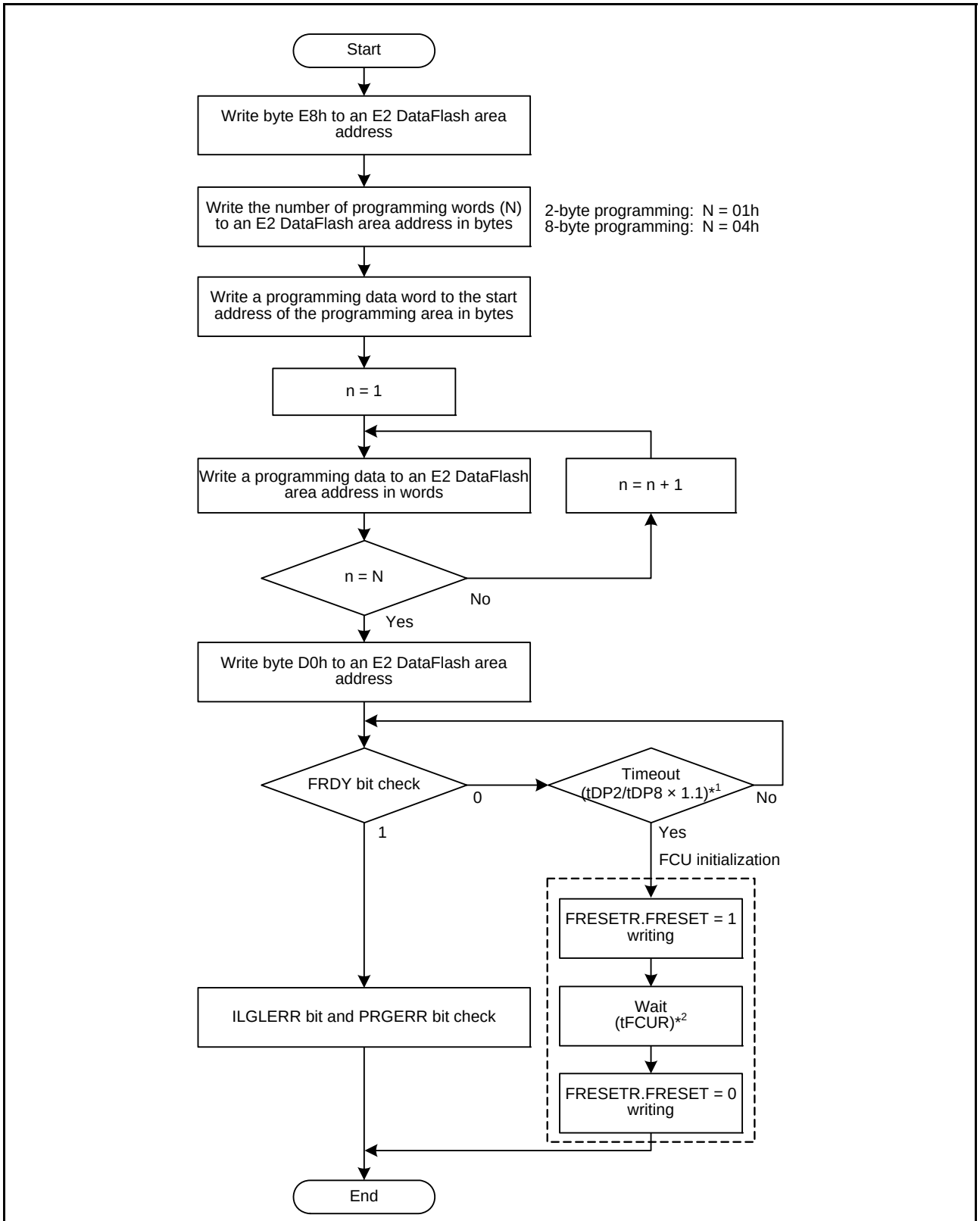
(1) Using the Peripheral Clock Notification Command

This command handles notification of the frequency of the peripheral clock. For details, see section 36.6.4, FCU Command Usage. Set the FENTRYD bit in FENTRYR to 1 and make settings to indicate an address within the E2 DataFlash area.

(2) Programming

To program the E2 DataFlash, use one of the programming commands.
Use byte access to write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and the number of words (N)*1 to be programmed in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the first word of data for programming to the address where the target area for programming starts.
In the case of 8-byte programming, 8 bytes (4 words) of data are written to the E2 DataFlash in the third to the sixth cycles, divided into four rounds. The start address of 8 bytes for programming is specified in the third cycle. The address range specified at this time must be an integral multiple of 8. The address range specified in the fourth to the sixth cycles does not need to be the address range for actual programming.
In the case of 2-byte programming, the address range and data for programming are specified in the third cycle. The address range must be an even number.
After writing words to addresses in the E2 DataFlash area N times, write byte D0h to an address within the E2 DataFlash area in cycle N + 3; 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.
If the area accessed in the third cycle to cycle N + 2 includes addresses that do not require programming, write FFFFh as the programming data for those addresses. Moreover, programming and erasure of the block that contains the address must be enabled by the corresponding setting for protection from programming and erasure in the DFLWE0 register.
Figure 37.5 shows the procedure for E2 DataFlash programming.

Note 1. N = 01h and 04h for 2- and 8-byte programming, respectively.



Note 1. tDP2/tDP8: Time required for programming 2-/8-byte data (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

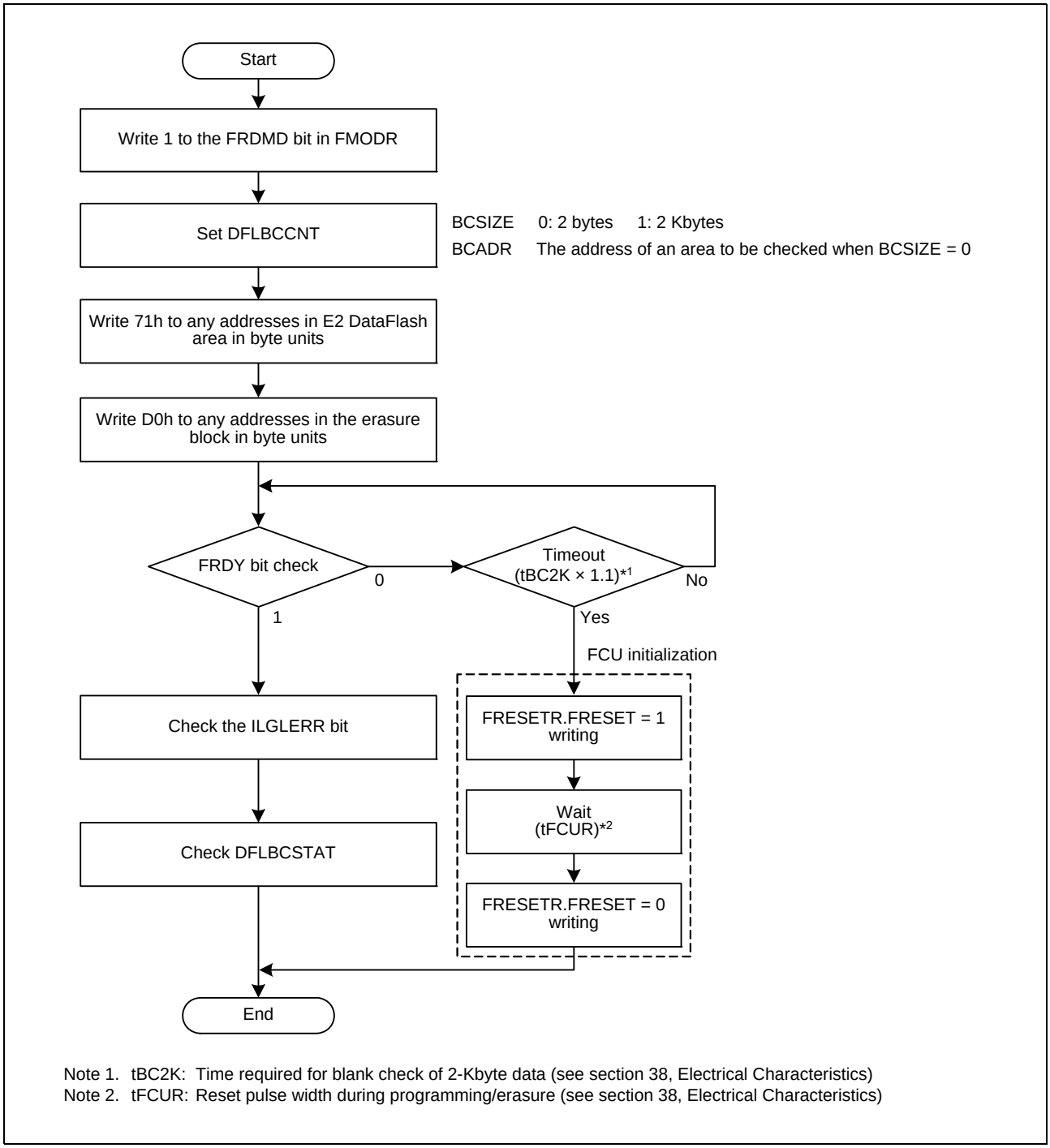
Figure 37.5 Procedure for E2 DataFlash Programming

(3) Erasure

To erase the E2 DataFlash, use the block erasure command. The E2 DataFlash is erased in the same way as the ROM (see section 36, ROM (Flash Memory for Code Storage)).
Note that the E2 DataFlash has a programming and erasure protection function that is controlled by DFLWE0. Erasure can only be performed with protection provided by the DFLWE0 setting disabled, so set the programming/erasure enable bit for the target erasure block to 1 before issuing the erasure command.

(4) Blank Checking

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 BCSIZE bit in DFLBCCNT is set to 1, blank check can be executed for the entire erased block (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 as a byte unit to an address in the E2 DataFlash. In the second cycle, when the value D0h is written as a byte unit 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 37.6 shows the procedure for blank checking of the E2 DataFlash.



Note 1. tBC2K: Time required for blank check of 2-Kbyte data (see section 38, Electrical Characteristics)
Note 2. tFCUR: Reset pulse width during programming/erasure (see section 38, Electrical Characteristics)

Figure 37.6 Procedure for Blank Checking of the E2 DataFlash

37.7 Protection

There are two types of E2 DataFlash programming/erasure protection: software protection and command-locked state.

37.7.1 Software Protection

In the software protection function, control register settings are used to disable E2 DataFlash programming, erasure, and reading. If an attempt is made to issue a programming, an erasure, or a reading command for the E2 DataFlash and the command violates current software protection, the FCU detects the error and enters the command-locked state.

(1) Protection through FWEPROR

If the FLWE[1:0] bits in FWEPROR are not set to 01b, programming cannot be performed in any mode.

(2) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the ROM/E2 DataFlash read mode is selected. Since the FCU does not accept commands in ROM/E2 DataFlash read mode, E2 DataFlash programming and erasure are disabled. If an attempt is made to issue an FCU command for the E2 DataFlash in ROM/E2 DataFlash read mode, the FCU detects an illegal command error and enters the command-locked state (see section 37.7.2, Command-Locked State).

(3) Protection through DFLWE0

When the DBWEj (j = 00 to 03) bit in DFLWE0 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 enters the command-locked state (see section 37.7.2, Command-Locked State).

(4) Protection through DFLRE0

When the DBREj (j = 00 to 03) bit in DFLRE0 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 enters the command-locked state (see section 37.7.2, Command-Locked State).

37.7.2 Command-Locked State

Command-locked state is the detection of errors in the issuing of FCU commands and of prohibited access, and response in the form of notification of the FCU malfunction and prohibition of the reception of further commands by the FCU (the FCU enters the command-locked state). When any bit from among the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FSTATR1.FCUERR bit, and the FASTAT.DFLAE, DFLRPE, and DFLWPE bits) is set to 1, the FCU will be in the command-locked state (FASTAT.CMDLK bit is set to 1), so programming and erasure of the E2 DataFlash are prohibited. To release the FCU from the command-locked state, a status register clearing command must be issued with FASTAT set to 10h.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFERR) interrupt will be generated if the FCU enters the command-locked state (the CMDLK bit in FASTAT becomes 1). While an E2 DataFlash-related interrupt enable bit (DFLAEIE, DFLRPEIE, or DFLWPEIE) in FAEINT is 1, an FIFERR interrupt will also be generated if the corresponding status bit (DFLAE, DFLRPE, or DFLWPE) in FASTAT becomes 1.

Table 37.6 shows the errors that lead to the command-locked state for the E2 DataFlash and the values of the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the DFLAE, DFLRPE, and DFLWPE bits in FASTAT) after the detection of each type of error. For the errors that lead to the command-locked state used in common by the ROM and E2 DataFlash (FENTRYR setting error, most illegal command errors, erasing errors, programming errors, and FCU errors, see section 36.8.2, Command-Locked State.

If the FCU enters the command-locked state due to a command other than a suspension command issued during programming or erasure processing, the FCU continues programming or erasing the E2 DataFlash. In this state, the P/E suspension command cannot suspend programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1.

Table 37.6 Errors that Lead to the Command-Locked State (for E2 DataFlash Only)

Error	Description	ILGLERR	ERSERR	PRGERR	DFLAE	DFLRPE	DFLWPE
Illegal command error	The value specified in the second cycle of a programming command was other than 01h and 04h.	1	0	0	0	0	0
	A lock bit programming command was issued for an area in the E2 DataFlash while the FENTRYD bit of FENTRYR register was set to 1.	1	0	0	0	0	0
E2 DataFlash access error	A read access command was issued for the E2 DataFlash area while FENTRYD = 1 in FENTRYR in E2 DataFlash P/E normal mode.	1	0	0	1	0	0
	A write access command was issued for the E2 DataFlash area while FENTRYD = 0.	1	0	0	1	0	0
	An access command was issued for the E2 DataFlash area while the FENTRY0 bit in FENTRYR was 1.	1	0	0	1	0	0
E2 DataFlash read protect error	A read access command was issued for the E2 DataFlash area while it was protected against reading by the DFLRE0 setting.	1	0	0	0	1	0
E2 DataFlash programming protect error	A program/block erase command was issued for the E2 DataFlash area while it was protected against programming and erasure by the DFLWE0 setting.	1	0	0	0	0	1

37.8 Boot Mode

To program or erase the data area in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, see section 36.10, Boot Mode. This section describes only the commands dedicated for the E2 DataFlash.

37.8.1 Inquiry/Selection Commands

Table 37.7 lists the inquiry/selection commands dedicated to the E2 DataFlash. The data area inquiry and data area information inquiry commands are used in the step of “Inquiry regarding area programming information” in the flowchart shown in Figure 36.26, Example of Procedure to Use Inquiry/Selection Commands for User Area/User Boot Area in section 36.10.6, Inquiry/Selection Command Wait.

Table 37.7 Inquiry/Selection Commands (only for E2 DataFlash)

Command Name	Function
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

Each command is described in detail below. The “command” in the description indicates a command sent from the host to the RX220 and the “response” indicates a response sent from the RX220 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX220 becomes 00h.

(1) Data Area Inquiry

In response to a data area inquiry command sent from the host, the RX220 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

(2) Data Area Information Inquiry

In response to a data area information inquiry command sent from the host, the RX220 returns the number of data area areas 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 areas (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 erasure block information inquiry command (see section 36.10.6, Inquiry/Selection Command Wait).

37.8.2 Programming/Erasing Commands

Table 37.8 lists the programming/erasing commands dedicated to the E2 DataFlash. E2 DataFlash-dedicated commands are provided only for checksum and blank check of the data area; the programming, erasing, and reading commands are used in common for the ROM and E2 DataFlash.

To program the data area, issue from the host a user area programming selection command and then a 256-byte programming command specifying a data area address as the programming address. To erase the data area, issue an erasure selection command and then a block erasure command specifying an erasure block in the data area. The information concerning the erasure block in the data area is included in the response to the erasure block information inquiry command. 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.

For the user area programming selection, user boot area programming selection, 256-byte programming, erasure selection, block erasure, and memory read commands, refer to section 36.10.8, Programming/Erasure Command Wait. For the erasure block information inquiry command, refer to section 36.10.6, Inquiry/Selection Command Wait.

Table 37.8 Programming/Erasure Commands (only for E2 DataFlash)

Command Name	Function
Data area checksum	Performs checksum verification for the data area
Data area blank check	Checks whether the data area is blank

Each command is described in detail below. The “command” in the description indicates a command sent from the host to the RX220 and the “response” indicates a response sent from the RX220 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX220 becomes 00h.

(1) Data Area Checksum

In response to a data area checksum command sent from the host, the RX220 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)

(2) Data Area Blank Check

In response to a data area blank check command sent from the host, the RX220 checks whether the data area is completely erased. When the data area is completely erased, the RX220 returns a response (06h). If the data area has an unerased area, the RX220 returns an error response (E2h, 52h).

Command	62h	
Response	06h	
Error response	E2h	52h

37.9 Usage Notes

(1) Protection of Data Area Immediately after a Reset

As the initial values of DFLRE0 and DFLWE0 are 0000h, programming, erasure, and reading of the data area are disabled immediately after a reset. To read data from the data area, set DFLRE0 appropriately before accessing the data area. To program or erase the data area, set DFLWE0 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 enters the command-locked state.

(2) Other Points to Note

The other points to note are the same as for the ROM. See section 36.12, Usage Notes of section 36, ROM (Flash Memory for Code Storage).
However, blank checking has been added for the E2 DataFlash. Therefore, for the E2 DataFlash, read “programming, erasure, or blank checking” for “programming and erasure” in the usage notes.

38. Electrical Characteristics

38.1 Absolute Maximum Ratings

Table 38.1 Absolute Maximum Ratings

Conditions: VSS = AVSS0 = VREFL0 = 0 V

Item	Symbol	Value	Unit
Power supply voltage	VCC	–0.3 to +6.5	V
Input voltage (except for ports for 5V tolerant*1 and port 4)	V_{in}	–0.3 to VCC +0.3*3	V
Input voltage (port 4)	V_{in}	–0.3 to AVCC0 +0.3*3	V
Input voltage (ports for 5 V tolerant*1)	V_{in}	–0.3 to +6.5	V
Analog power supply voltage	AVCC0*2	–0.3 to +6.5	V
Reference power supply voltage	VREFH0*2	–0.3 to AVCC0 +0.3*3	V
Analog input voltage (except for port 4)	V_{AN}	–0.3 to VCC +0.3*3	V
Analog input voltage (port 4)	V_{AN}	–0.3 to AVCC0 +0.3*3	V
Operating temperature	T_{opr}	–40 to +105	°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.

Connect the VCL pin to a VSS pin via a 0.1 µF (±20% accuracy) capacitor. The capacitor must be placed as close to the pin as possible.

Note 1. Ports 12, 13, 16, and 17 are 5 V tolerant.

Note 2. Set to the same potential as VCC. When the A/D converter is not used, do not leave the AVCC0, VREFH0, AVSS0, and VREFL0 pins open. Connect the AVCC0 and VREFH0 pins to VCC, and the AVSS0 and VREFL0 pins to VSS, respectively.

Note 3. The maximum value is 6.5 V.

38.2 DC Characteristics

Table 38.2 DC Characteristics (1)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus, 5 V tolerant)	V_{IH}	VCC × 0.7	—	5.8	V	
	Ports 12, 13, 16, and 17 (5 V tolerant)		VCC × 0.8	—	5.8		
	Ports 0, 14, 15, 2, 3, 4, 5, A, B, C, D, E, H, J, and RES#		VCC × 0.8	—	VCC + 0.3		
	RIIC input pin (except for SMBus)	V_{IL}	–0.3	—	VCC × 0.3		
	Other than RIIC input pin		–0.3	—	VCC × 0.2		
	RIIC input pin (except for SMBus)	ΔV_T	VCC × 0.05	—	—		
	Other than RIIC input pin		VCC × 0.1	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD pin	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL		VCC × 0.8	—	VCC + 0.3		
	RIIC input pin (SMBus)		2.1	—	VCC + 0.3		
	MD pin	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL		–0.3	—	VCC × 0.2		
	RIIC input pin (SMBus)		–0.3	—	0.8		

Table 38.3 DC Characteristics (2)

Conditions: VCC = AVCC0 = 1.62 to 2.7 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	Ports 12, 13, 16, and 17 (5 V tolerant)	V_{IH}	VCC × 0.8	—	5.8	V	
	Ports 0, 14, 15, 2, 3, 4, 5, A, B, C, D, E, H, and J		VCC × 0.8	—	VCC + 0.3		
	RES#		VCC × 0.9	—	VCC + 0.3		
	Ports 0, 1, 2, 3, 4, 5, A, B, C, D, E, H, and J	V_{IL}	–0.3	—	VCC × 0.2		
	RES#		–0.3	—	VCC × 0.1		
	All input pins	ΔV_T	VCC × 0.01	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD pin	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL		VCC × 0.8	—	VCC + 0.3		
	MD pin	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL		–0.3	—	VCC × 0.2		

Table 38.4 DC Characteristics (3)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES#, MD pin, P35/NMI	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	Other pins except for ports for 5 V tolerant	$\lvert I_{TSI} \rvert$	—	—	0.2	µA	V_{in} = 0 V, VCC
	Ports for 5 V tolerant		—	—	1.0		V_{in} = 0 V, 5.8 V
Input capacitance	All input pins (except for XCIN and XCOUT)	C_{in}	—	—	15	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	XCIN and XCOUT		—	—	3		

Table 38.5 DC Characteristics (4)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC						Unit	Test Conditions
			1.62 to 2.7 V		2.7 to 4.0 V		4.0 to 5.5 V			
			Min.	Max.	Min.	Max.	Min.	Max.		
Input pull-up MOS current	All ports (except for port 35)	I_p	–150	–5	–200	–10	–400	–50	µA	V_{in} = 0 V

Table 38.6 DC Characteristics (5)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item					Symbol	Typ.*9	Max.	Unit	Test Conditions
Supply current*1	Medium-speed operating modes 1A and 1B	Normal operating mode	No peripheral operation*2	ICLK = 32 MHz	I_{CC}	4.6	—	mA	
				ICLK = 20 MHz		3.2	—		
			All peripheral operation: Normal*3	ICLK = 32 MHz		14	—		
				ICLK = 20 MHz		9.5	—		
			All peripheral operation: Max.*3	ICLK = 32 MHz		—	25		
				ICLK = 20 MHz		—	19		
		Sleep mode	No peripheral operation*2	ICLK = 32 MHz		3.8	—		
				ICLK = 20 MHz		3.0	—		
			All peripheral operation: Normal*3	ICLK = 32 MHz		10	—		
				ICLK = 20 MHz		7	—		
		All-module clock stop mode		ICLK = 32 MHz		2.5	—		
				ICLK = 20 MHz		2.0	—		
		Increase during BGO operation*4	Medium-speed operating mode 1A			17	—		
			Medium-speed operating mode 1B			17	—		
	Low-speed operating mode 1	Normal operating mode	No peripheral operation*5	ICLK = 8 MHz		1.4	—		
				ICLK = 4 MHz		0.9	—		
				ICLK = 2 MHz		0.7	—		
			All peripheral operation: Normal*6	ICLK = 8 MHz		4.2	—		
				ICLK = 4 MHz		2.6	—		
				ICLK = 2 MHz		1.8	—		
			All peripheral operation: Max.*6	ICLK = 8 MHz		—	6.5		
				ICLK = 4 MHz		—	3.7		
				ICLK = 2 MHz		—	2.4		
		Sleep mode	No peripheral operation*5	ICLK = 8 MHz		1.5	—		
				ICLK = 4 MHz		1.2	—		
				ICLK = 2 MHz		1.1	—		
			All peripheral operation: Normal*6	ICLK = 8 MHz		3.1	—		
				ICLK = 4 MHz		2.1	—		
				ICLK = 2 MHz		1.5	—		
		All-module clock stop mode		ICLK = 8 MHz		1.4	—		
				ICLK = 4 MHz		1.1	—		
				ICLK = 2 MHz		1.0	—		
	Low-speed operating mode 2	Normal operating mode	No peripheral operation*7	ICLK = 32 kHz		0.027	—		
			All peripheral operation: Normal*8	ICLK = 32 kHz		0.04	—		
			All peripheral operation: Max.*8	ICLK = 32 kHz		—	0.23		
		Sleep mode	No peripheral operation*7	ICLK = 32 kHz		0.024	—		
			All peripheral operation: Normal*8	ICLK = 32 kHz		0.034	—		
		All-module clock stop mode				0.016	—		

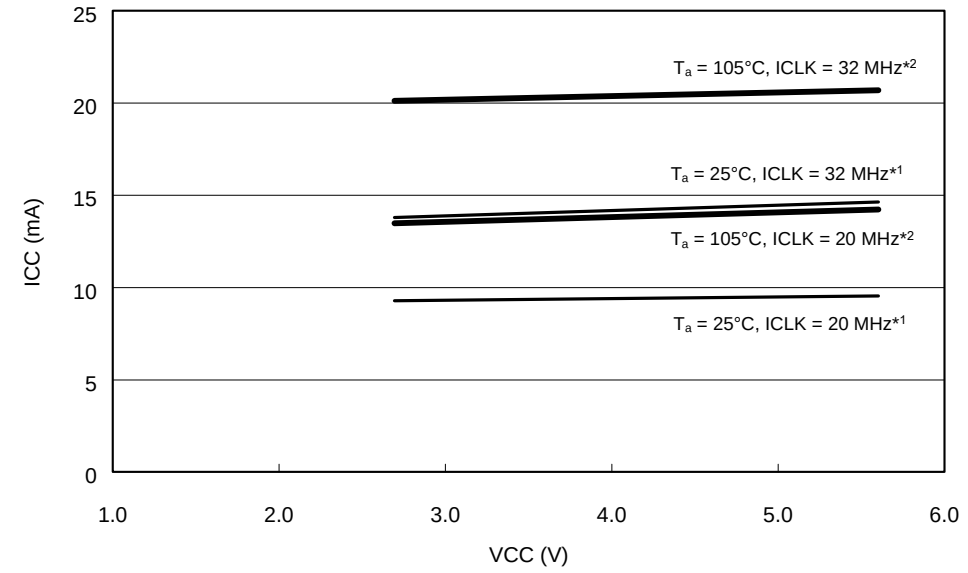
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.

Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is HOCO. BCLK, FCLK, and PCLK are set to divided by 64.

Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is HOCO. BCLK, FCLK, and PCLK are ICLK divided by 1.

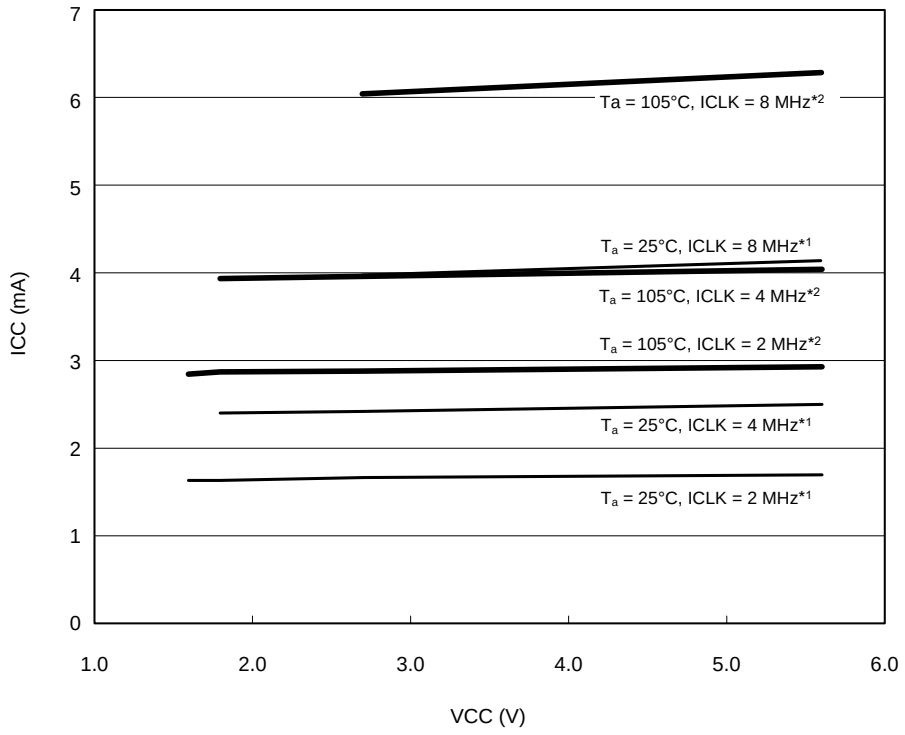
Note 4. This is the increase if data is programmed to or erasing from the ROM or E2 DataFlash during program execution.

Note 5. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is HOCO. BCLK, FCLK, and PCLK are set to divided by 64.
Note 6. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is HOCO. BCLK, FCLK, and PCLK are ICLK divided by 1.
Note 7. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is the sub oscillation circuit. BCLK, FCLK, and PCLK are set to divided by 64.
Note 8. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is the sub oscillation circuit. BCLK, FCLK, and PCLK are ICLK divided by 1.
Note 9. VCC = 3.3 V.



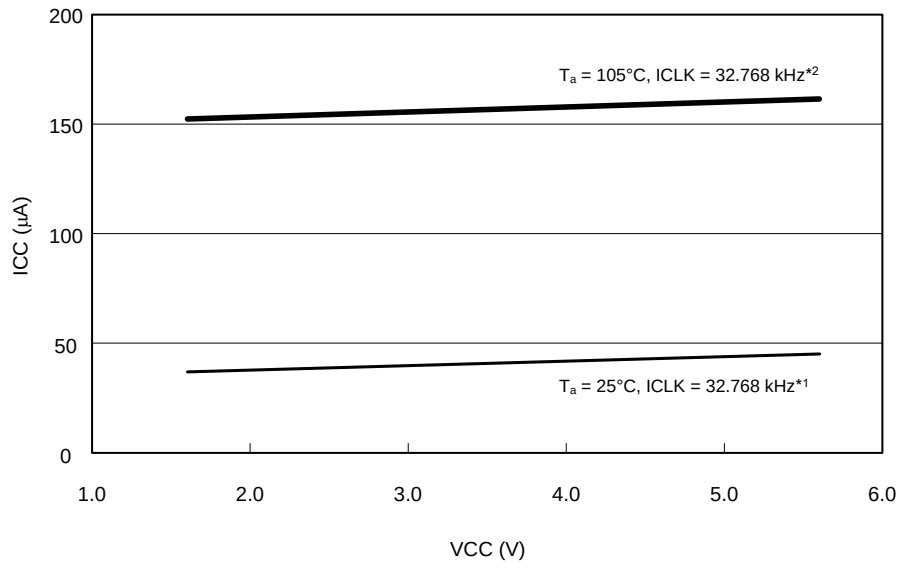
Note 1. All peripheral operation is normal. This does not include BGO operation.
Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation.
Average value of the tested upper-limit samples during product evaluation.

Figure 38.1 Voltage Dependency in Medium-Speed Operating Modes 1A and 1B (Reference Data)



Note 1. All peripheral operation is normal. This does not include BGO operation.
Average value of the tested middle samples during product evaluation
Note 2. All peripheral operation is maximum. This does not include BGO operation.
Average value of the tested upper-limit samples during product evaluation.

Figure 38.2 Voltage Dependency in Low-Speed Operating Mode 1 (Reference Data)



Note 1. All peripheral operation is normal. This does not include BGO operation.
Average value of the tested middle samples during product evaluation
Note 2. All peripheral operation is maximum. This does not include BGO operation.
Average value of the tested upper-limit samples during product evaluation.

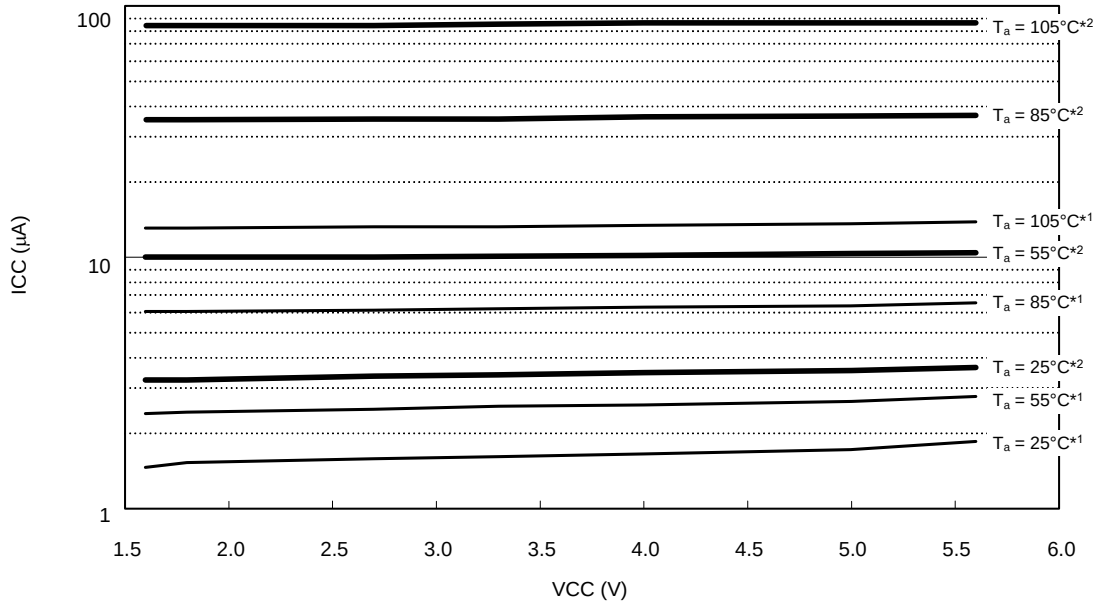
Figure 38.3 Voltage Dependency in Low-Speed Operating Mode 2 (Reference Data)

Table 38.7 DC Characteristics (6)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

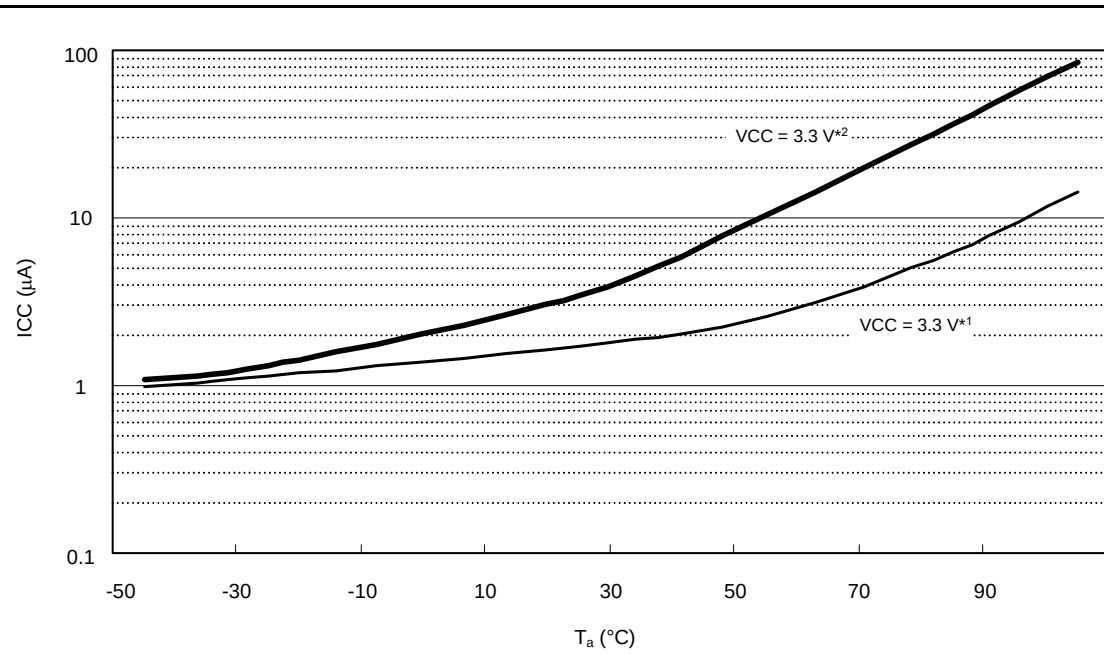
Item				Symbol	Typ.*3	Max.	Unit	Test Conditions
Supply current*1	Software standby mode*2	Flash memory power supplied, HOCO power supplied, POR low power consumption function disabled (SOFTCUT bit = 000b)	T_a = 25°C	I_{CC}	9.3	16.4	µA	
			T_a = 55°C		11.3	25		
			T_a = 85°C		16	62		
			T_a = 105°C		25	127		
		Flash memory power not supplied, HOCO power not supplied, POR low power consumption function enabled (SOFTCUT bit = 11xb)	T_a = 25°C		1.7	7.0		
			T_a = 55°C		2.6	15		
			T_a = 85°C		6.3	51		
			T_a = 105°C		14.2	115		
	Increments produced by running voltage detection circuits and disabling the POR low power consumption function				1.4	—		
	Increment for RTC operation (low CL)				0.6	—		
	Increment for RTC operation (standard CL)				1.4	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. The IWDT and LVD are stopped.
Note 3. VCC = 3.3 V.



Note 1. Average value of the tested middle samples during product evaluation
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 38.4 Voltage Dependency in Software Standby Mode (SOFTCUT Bit = 11xb) (Reference Data)



Note 1. Average value of the tested middle samples during product evaluation
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 38.5 Temperature Dependency in Software Standby Mode (SOFTCUT Bit = 11xb) (Reference Data)

Table 38.8 DC Characteristics (7)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Typ.	Max.	Unit	Test Conditions
Permissible total consumption power*1	Pd	—	350	mW	T_a = –40 to 85°C
		—	150		85°C < T_a ≤ 105°C

Note: • Please contact Renesas Electronics sales office for derating of operation under T_a = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

Note 1. Total power dissipated by the entire chip (including output currents)

Table 38.9 DC Characteristics (8)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VREFH0 = 1.62 to AVCC0, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion	Conversion time = 1.56 µs	AI_{CC}	—	1.0	3.0	mA	
	Waiting for A/D conversion (all units)			—	0.2	3.0	µA	
Reference power supply current	During A/D conversion	Conversion time = 1.56 µs	I_{REFH0}	—	0.1	0.2	mA	
	Waiting for A/D conversion (all units)			—	0.2	0.4	µA	

Table 38.10 DC Characteristics (9)

Conditions: VCC = AVCC0, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	V_{RAM}	1.62	—	—	V	

Table 38.11 DC Characteristics (10)

Conditions: VCC = AVCC0 = 0 to 5.5 V, VREFH0 = 0 to AVCC0, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
VCC rising gradient	SrVCC	0.02	—	20	ms/V	At cold start

Table 38.12 DC Characteristics (11)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (5.5 V) and lower limit (1.62 V).

When VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10	kHz	VCC × 0.1 < $V_{r(VCC)}$ ≤ VCC × 0.2
		—	—	1	MHz	VCC × 0.05 < $V_{r(VCC)}$ ≤ VCC × 0.1
		—	—	10	MHz	$V_{r(VCC)}$ ≤ VCC × 0.05
Allowable voltage change rising/ falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

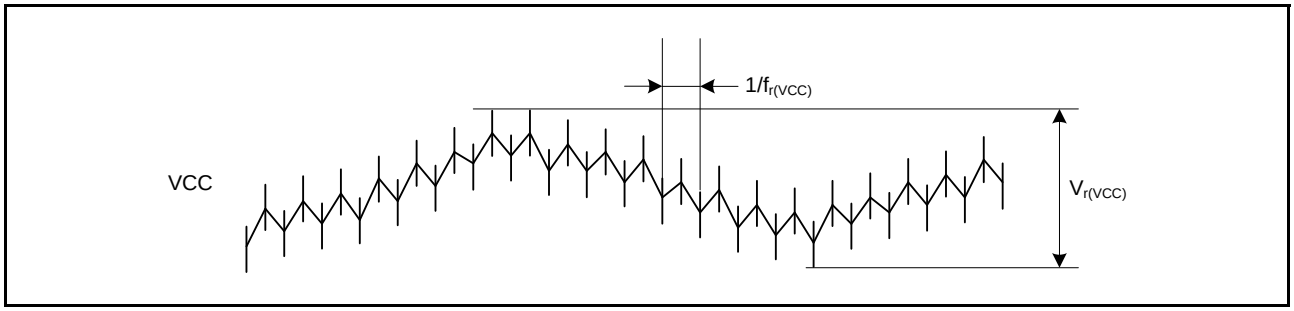


Figure 38.6 Ripple Waveform

Table 38.13 Permissible Output Currents (1)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, when total power (mW) < 1000 – 10 × T_a

Item			Symbol	Max.	Unit
Permissible output low current (average value per 1 pin)		Normal output mode	I_{OL}	4.0	mA
		High-drive output mode		16.0	
Permissible output low current (maximum value per 1 pin)		Normal output mode		4.0	mA
		High-drive output mode		16.0	
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}	–4.0	mA
		High-drive output mode		–8.0	
Permissible output high current (maximum value per 1 pin)		Normal output mode		–4.0	mA
		High-drive output mode		–8.0	
Permissible output high current (total)	Total of all output pins		ΣI_{OH}	–60	mA

Table 38.14 Permissible Output Currents (2)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, when total power (mW) ≥ 1000 – 10 × T_a

Item			Symbol	Max.	Unit
Permissible output low current (average value per 1 pin)		Normal output mode	I_{OL}	2.0	mA
		High-drive output mode		8.0	
Permissible output low current (maximum value per 1 pin)		Normal output mode		2.0	mA
		High-drive output mode		8.0	
Permissible output low current (total)	Total of all output pins		ΣI_{OL}	40	mA
Permissible output high current (average value per 1 pin)		Normal output mode	I_{OH}	–2.0	mA
		High-drive output mode		–4.0	
Permissible output high current (maximum value per 1 pin)		Normal output mode		–2.0	mA
		High-drive output mode		–4.0	
Permissible output high current (total)	Total of all output pins		ΣI_{OH}	–30	mA

Table 38.15 Output Values of Voltage (1)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, when total power (mW) < 1000 – 10 × T_a

Item			Symbol	Min.	Max.	Unit	Test Conditions	
							VCC = 2.7 to 4.0 V	VCC = 4.0 to 5.5 V
Output low	All output pins (other than RIIC)	Normal output mode	V_{OL}	—	1.0	V	I_{OL} = 3.0 mA	I_{OL} = 4.0 mA
		High-drive output mode		—	1.0		I_{OL} = 8.0 mA	I_{OL} = 16.0 mA
	RIIC pins			—	0.4		I_{OL} = 3.0 mA	
				—	0.6		I_{OL} = 6.0 mA	
Output high	All output pins	Normal output mode	V_{OH}	VCC – 1.0	—	V	I_{OH} = – 3.0 mA	I_{OH} = – 4.0 mA
		High-drive output mode		VCC – 1.0	—		I_{OH} = – 5.0 mA	I_{OH} = – 8.0 mA

Table 38.16 Output Values of Voltage (2)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, when total power (mW) ≥ 1000 – 10 × T_a

Item			Symbol	Min.	Max.	Unit	Test Conditions	
							VCC = 2.7 to 4.0 V	VCC = 4.0 to 5.5 V
Output low	All output pins (other than RIIC)	Normal output mode	V_{OL}	—	1.0	V	I_{OL} = 2.0 mA	I_{OL} = 2.0 mA
		High-drive output mode		—	1.0		I_{OL} = 8.0 mA	I_{OL} = 8.0 mA
	RIIC pins			—	0.4		I_{OL} = 3.0 mA	
				—	0.6		I_{OL} = 6.0 mA	
Output high	All output pins	Normal output mode	V_{OH}	VCC – 1.0	—	V	I_{OH} = – 2.0 mA	I_{OH} = – 2.0 mA
		High-drive output mode		VCC – 1.0	—		I_{OH} = – 4.0 mA	I_{OH} = – 4.0 mA

Table 38.17 Output Values of Voltage (3)

Conditions: VCC = AVCC0 = 1.62 to 2.7 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output pins (other than RIIC)	Normal output mode	V_{OL}	—	0.4	V	I_{OL} = 0.5 mA
		High-drive output mode		—	0.4		I_{OL} = 2.0 mA
Output high	All output pins	Normal output mode	V_{OH}	VCC – 0.4	—	V	I_{OH} = –0.5 mA
		High-drive output mode		VCC – 0.4	—		I_{OH} = –1.0 mA

38.2.1 Standard I/O Pin Output Characteristics (1)

Figure 38.7 to Figure 38.11 show the characteristics when normal output is selected by the drive capacity control register.

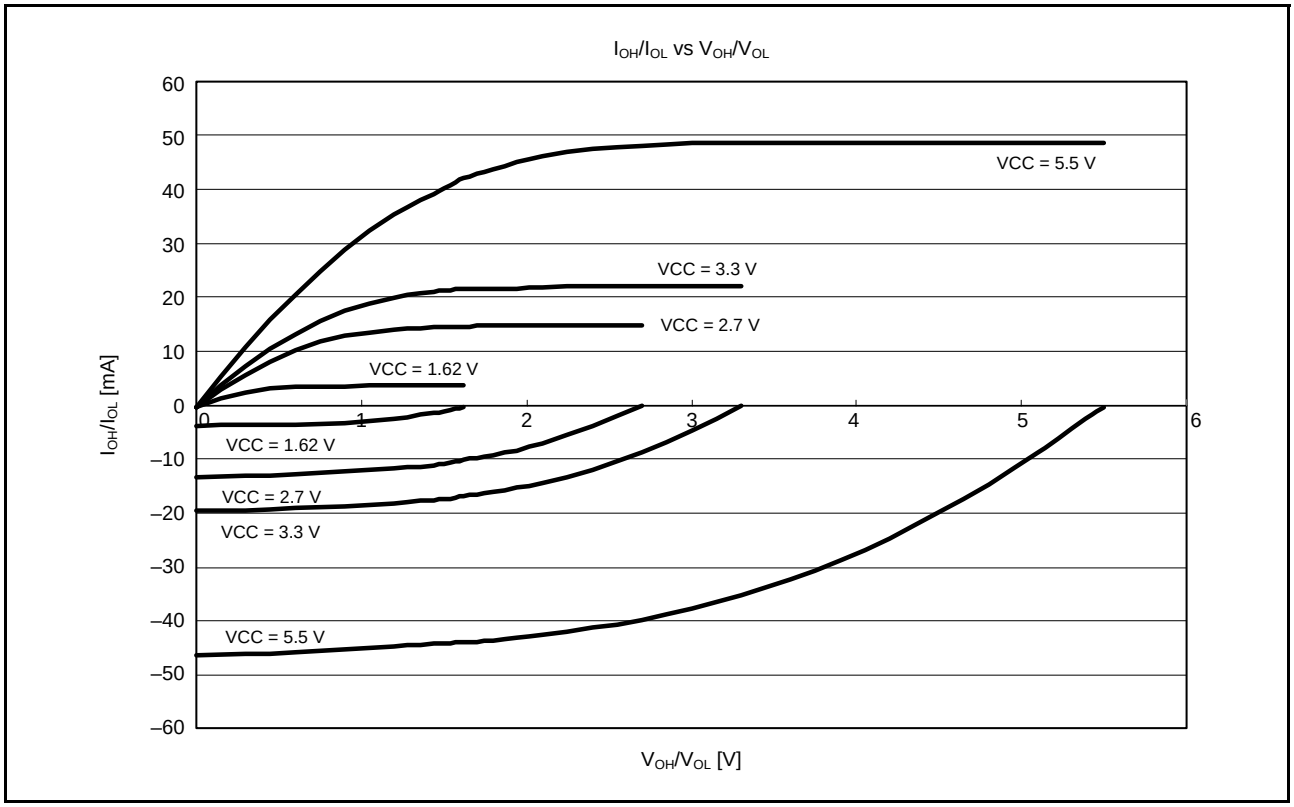


Figure 38.7 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C when Normal Output is Selected (Reference Data)

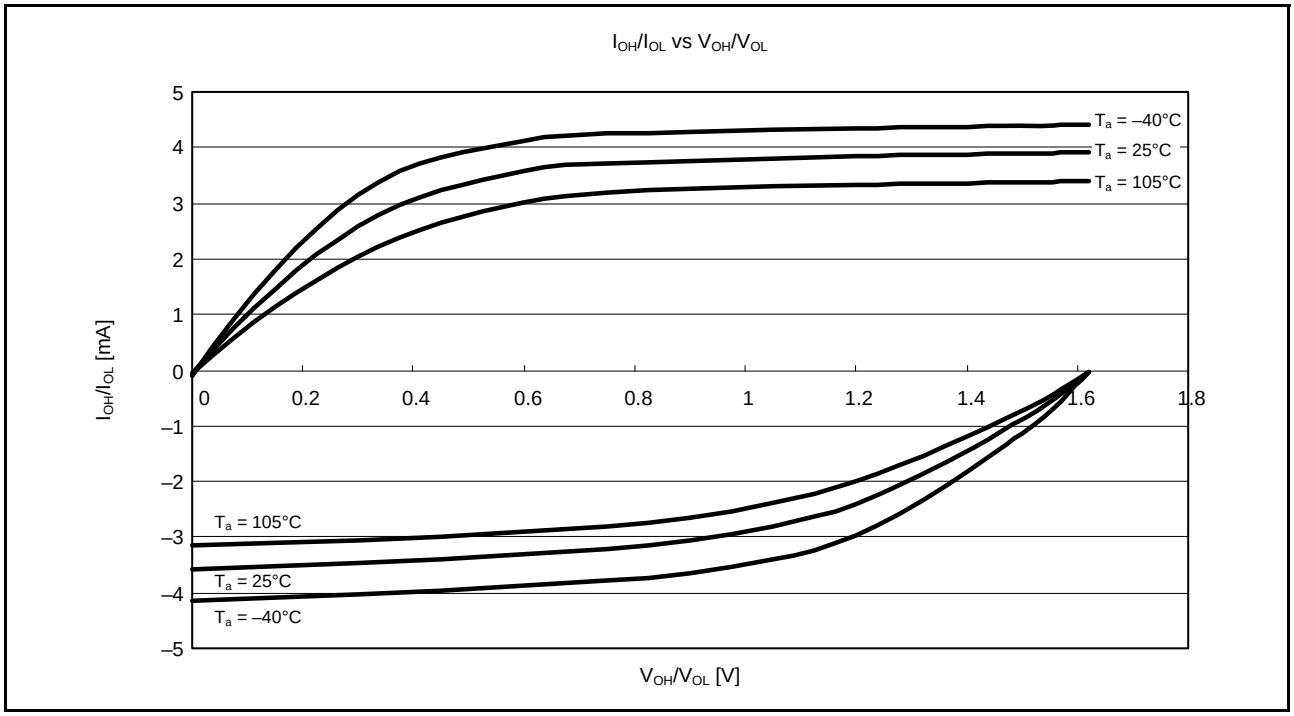


Figure 38.8 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 1.62 V when Normal Output is Selected (Reference Data)

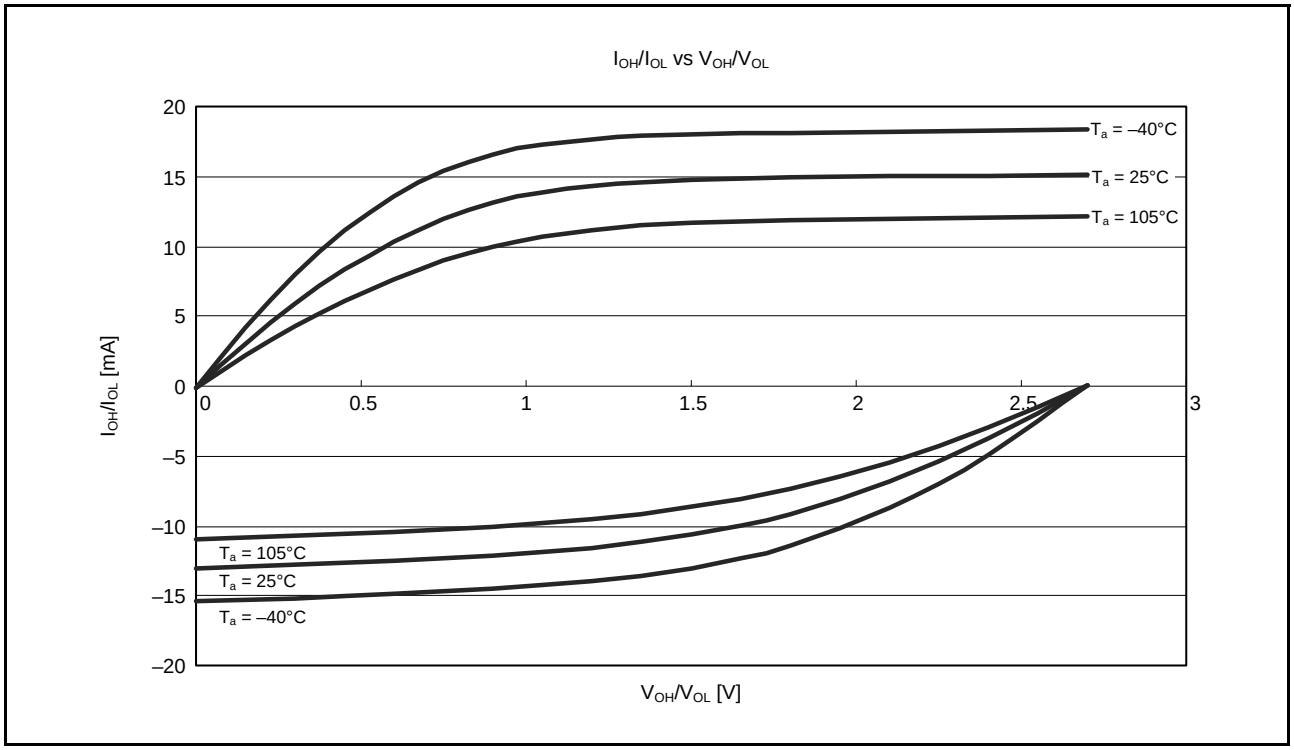


Figure 38.9 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V when Normal Output is Selected (Reference Data)

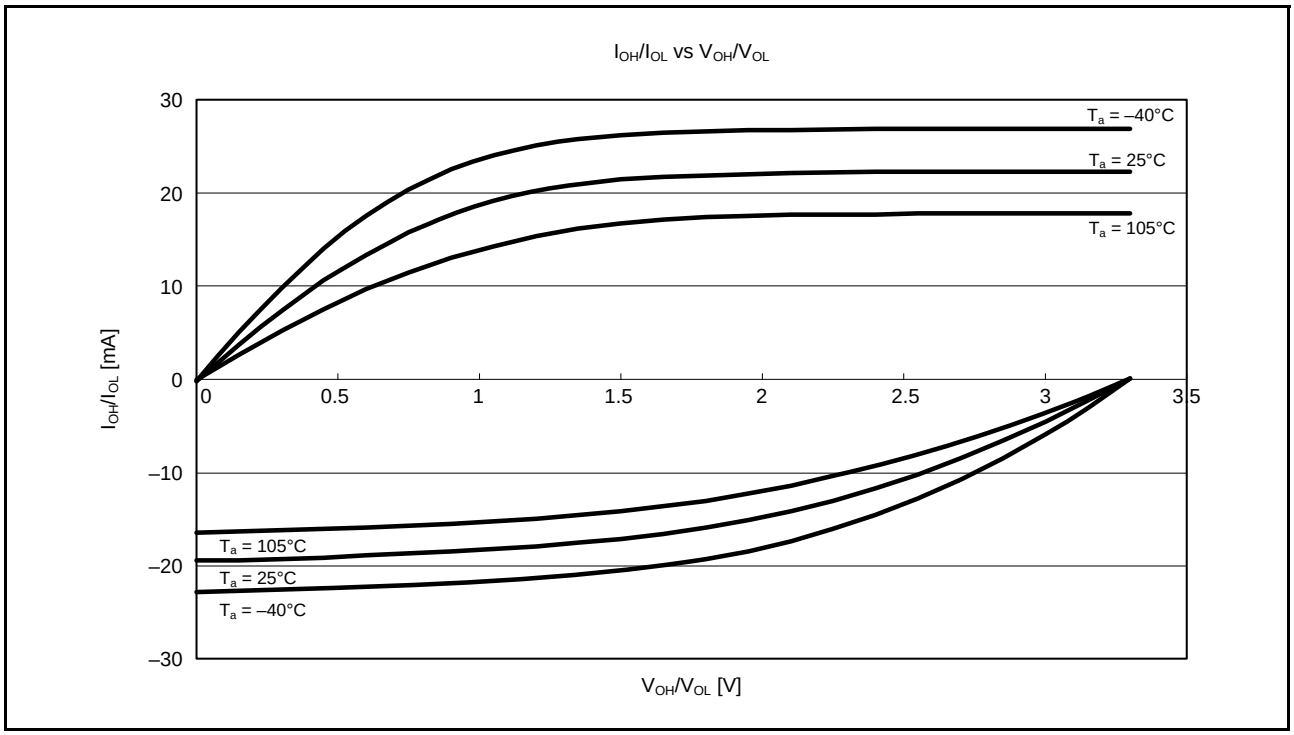


Figure 38.10 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 3.3 V when Normal Output is Selected (Reference Data)

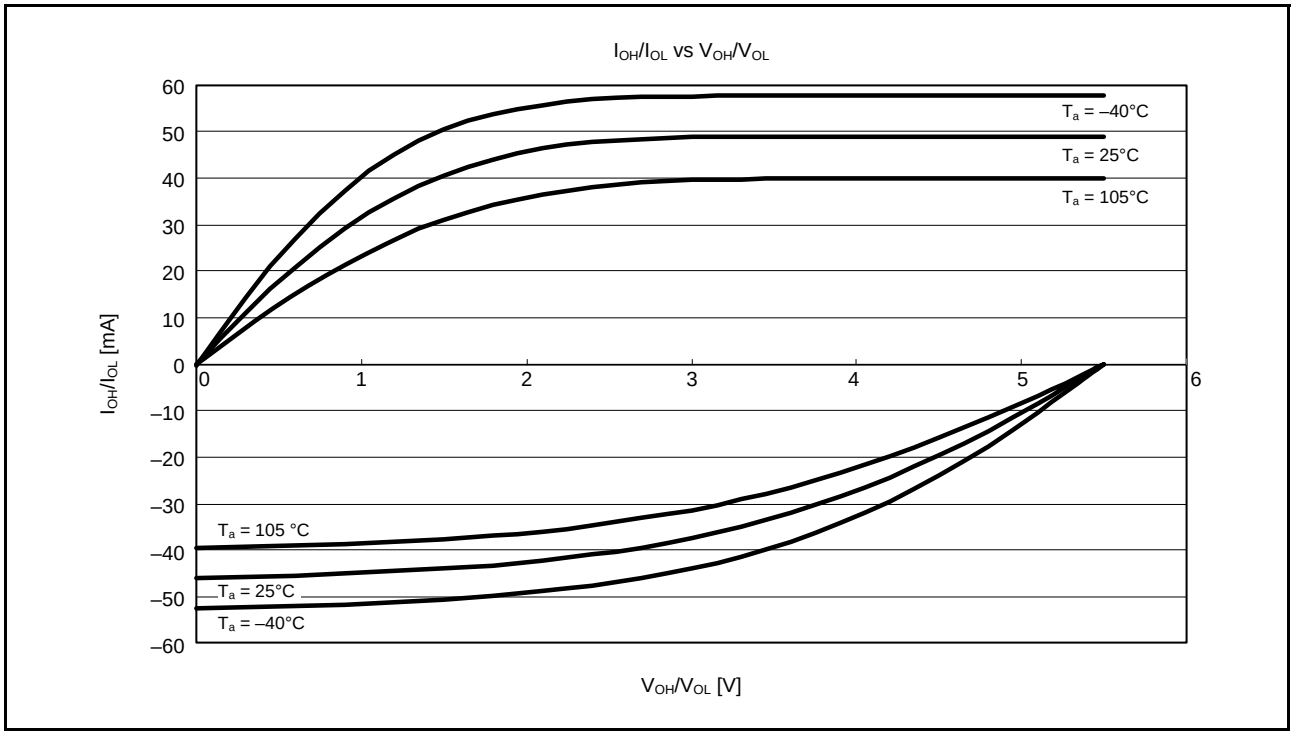


Figure 38.11 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.5 V when Normal Output is Selected (Reference Data)

38.2.2 Standard I/O Pin Output Characteristics (2)

Figure 38.12 to Figure 38.16 show the characteristics when high-drive output is selected by the drive capacity control register.

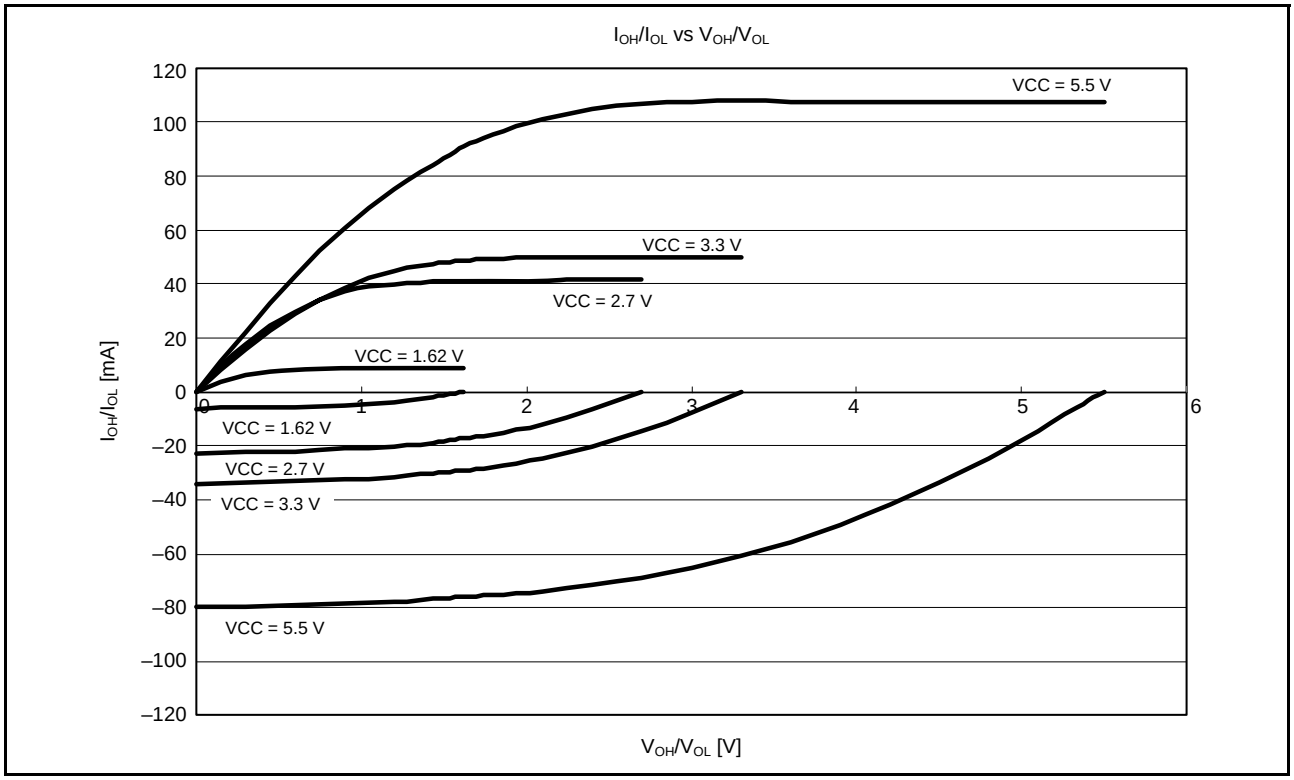


Figure 38.12 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C when High-Drive Output is Selected (Reference Data)

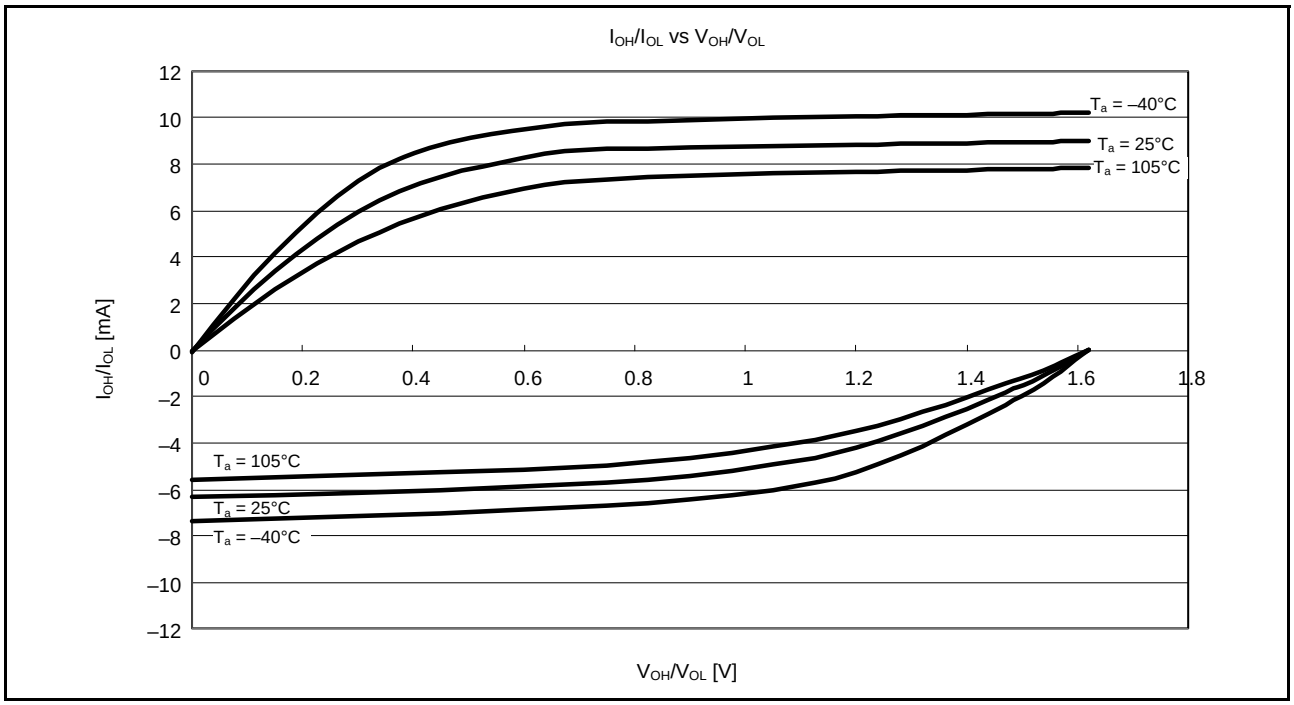


Figure 38.13 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 1.62 V when High-Drive Output is Selected (Reference Data)

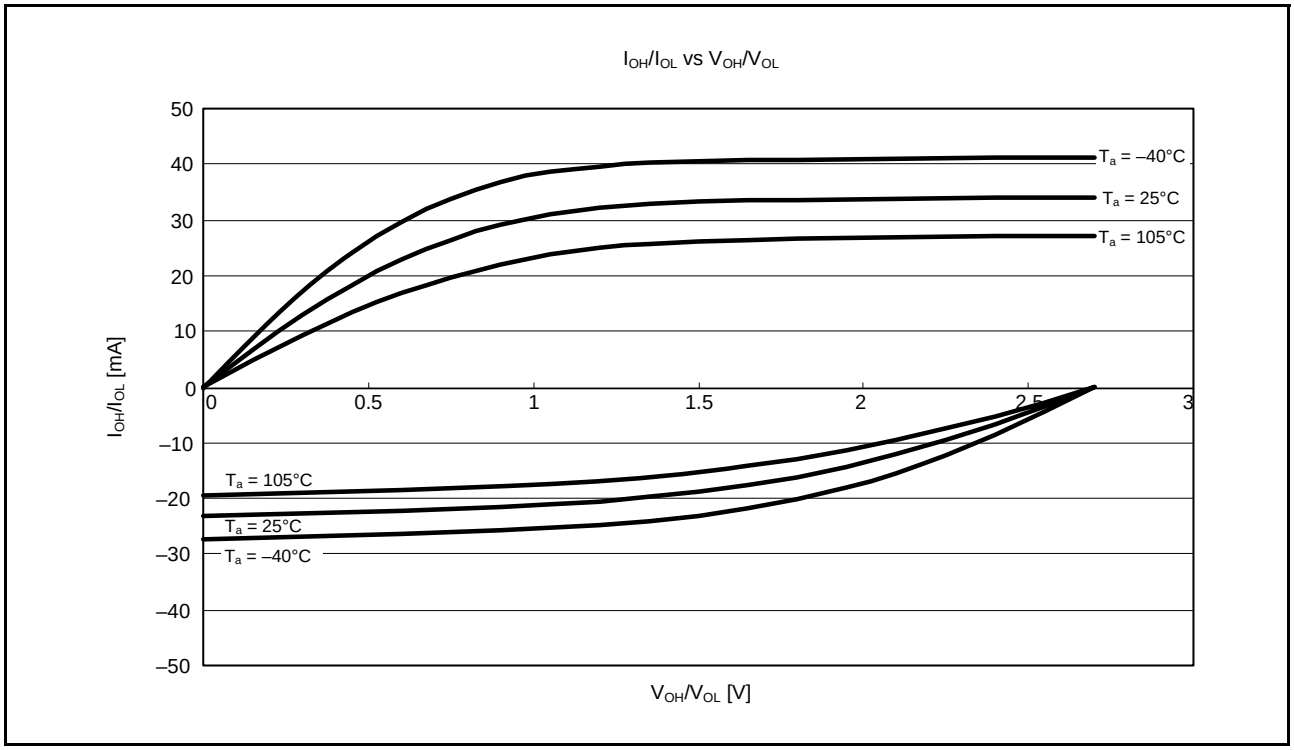


Figure 38.14 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V when High-Drive Output is Selected (Reference Data)

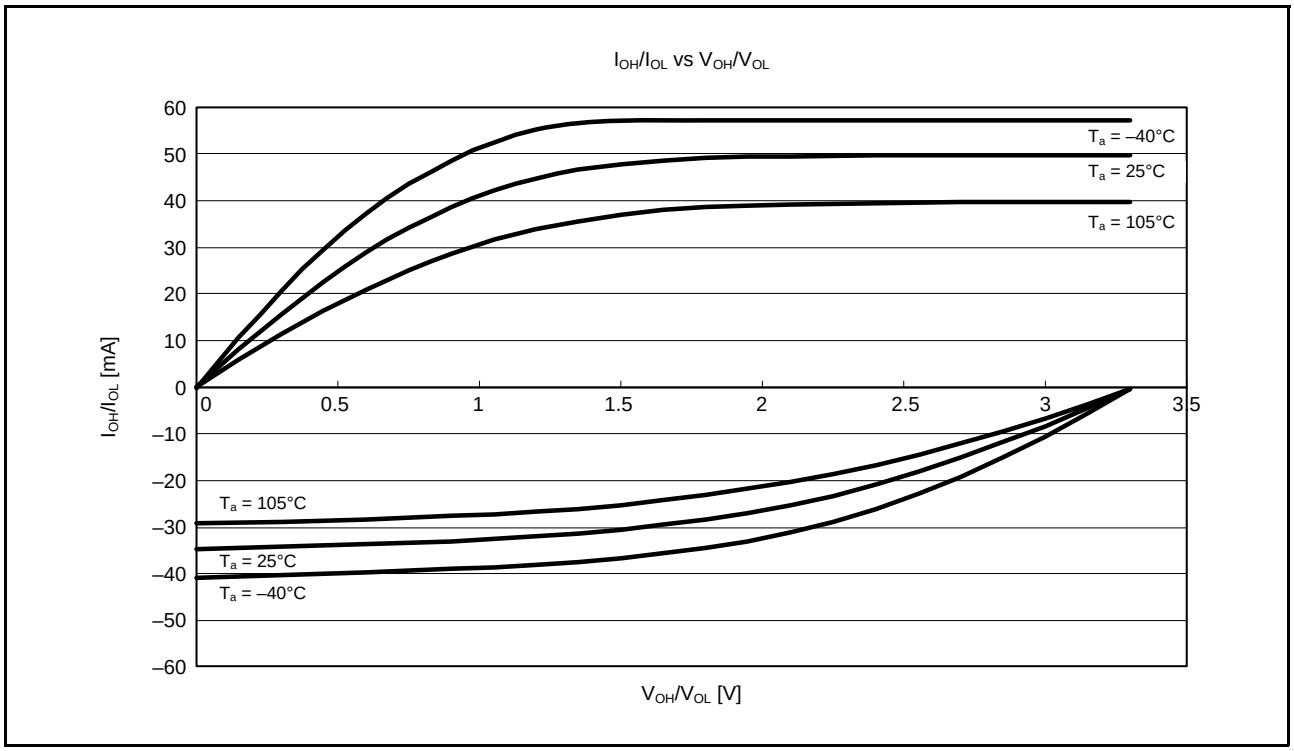


Figure 38.15 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 3.3 V when High-Drive Output is Selected (Reference Data)

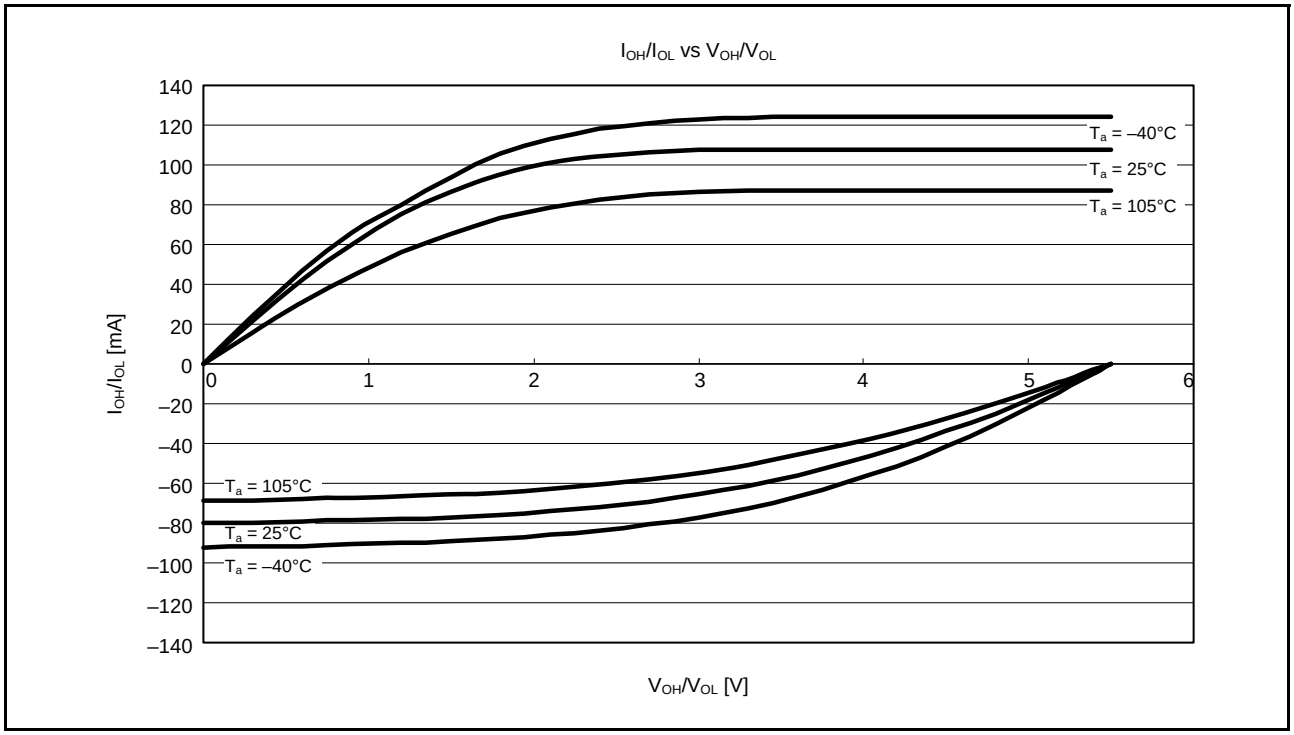


Figure 38.16 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.5 V when High-Drive Output is Selected (Reference Data)

38.2.3 RIIC Pin Output Characteristics

Figure 38.17 to Figure 38.20 show the output characteristics of the RIIC pin.

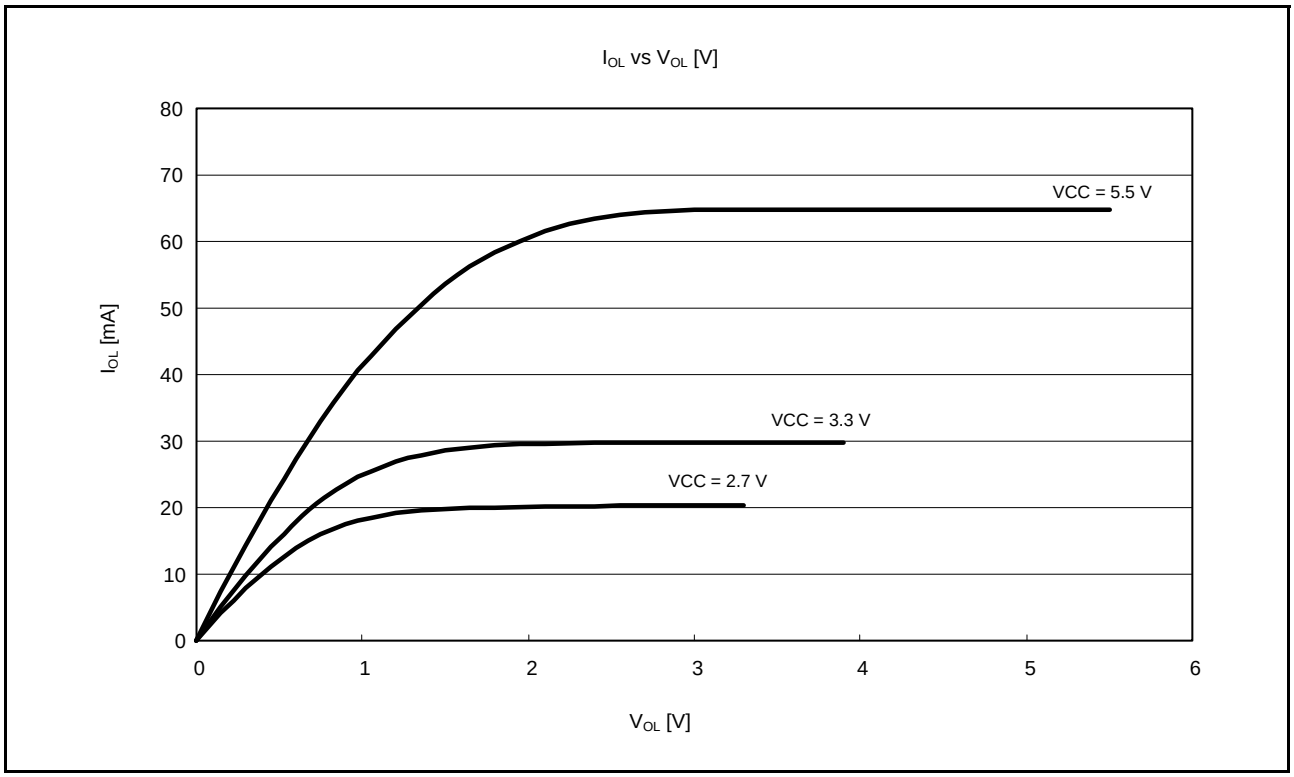


Figure 38.17 V_{OL} and I_{OL} Voltage Characteristics of RIIC Output Pin at T_a = 25°C (Reference Data)

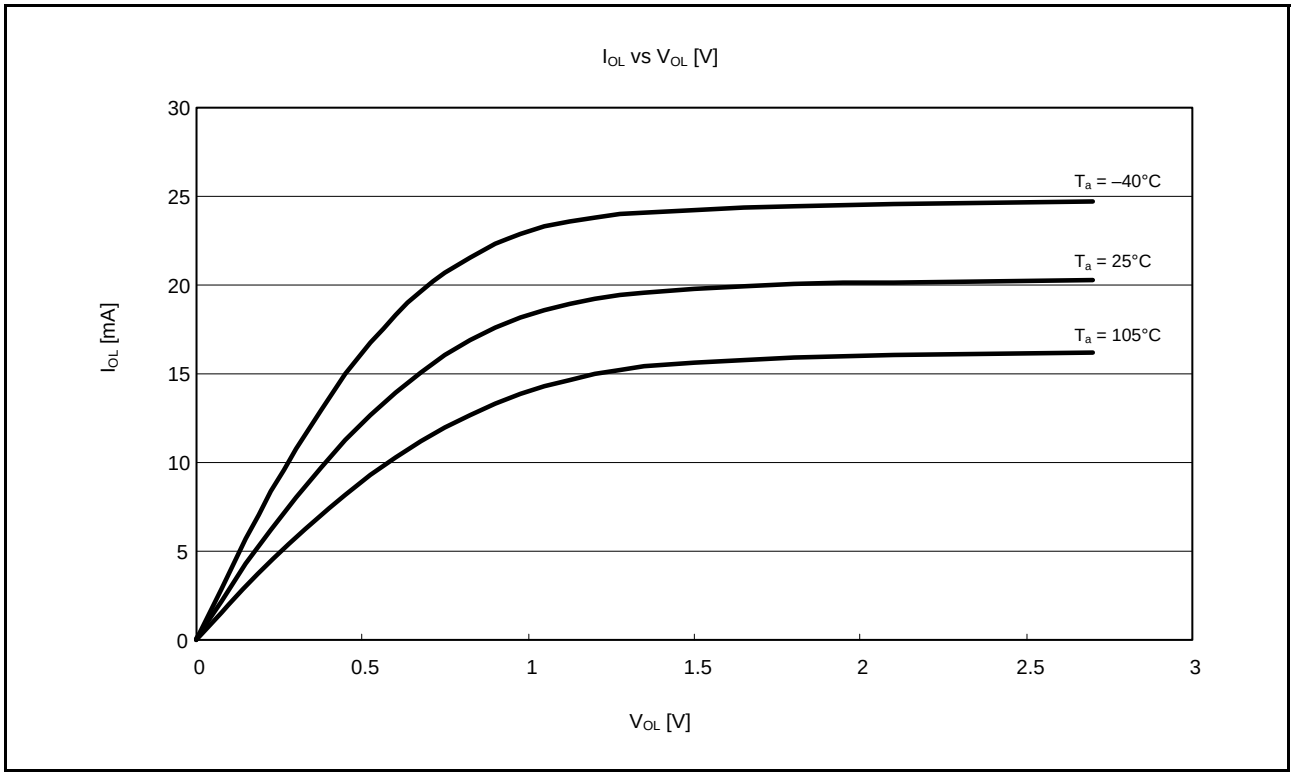


Figure 38.18 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 2.7 V (Reference Data)

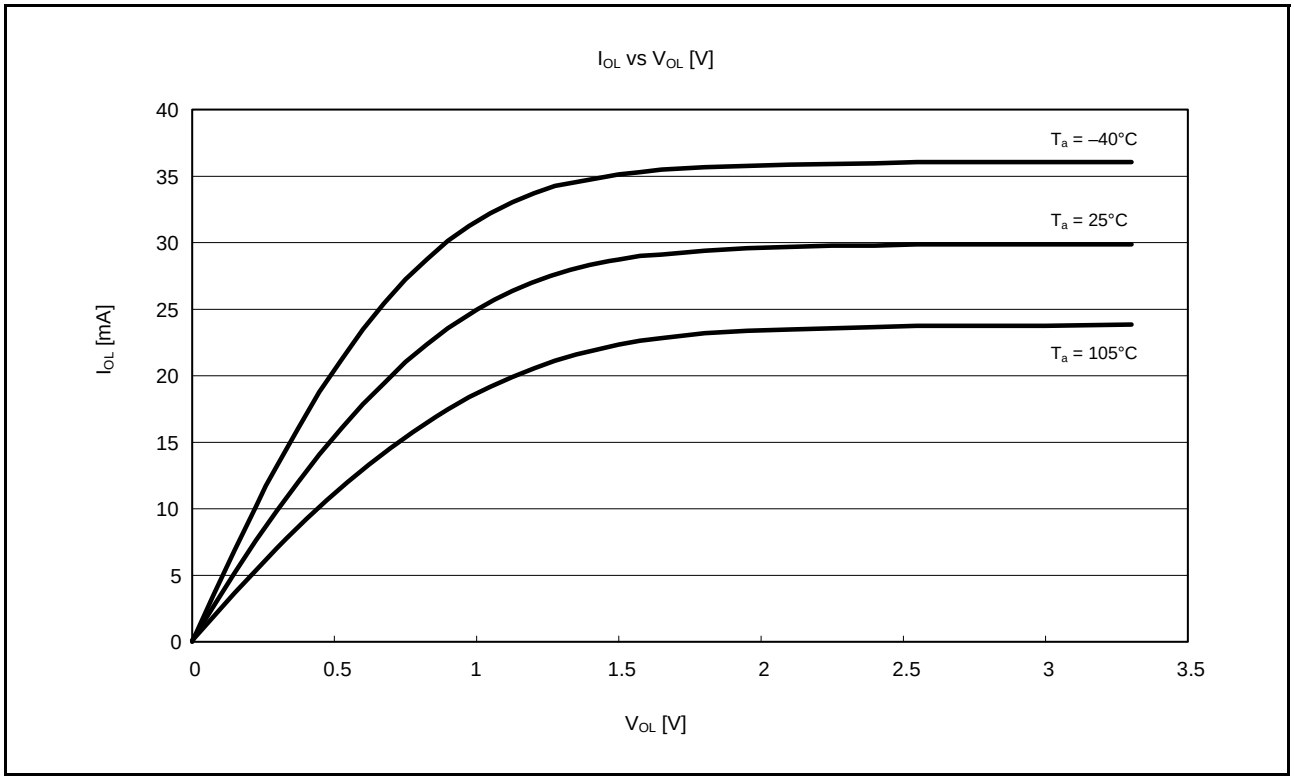


Figure 38.19 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 3.3 V (Reference Data)

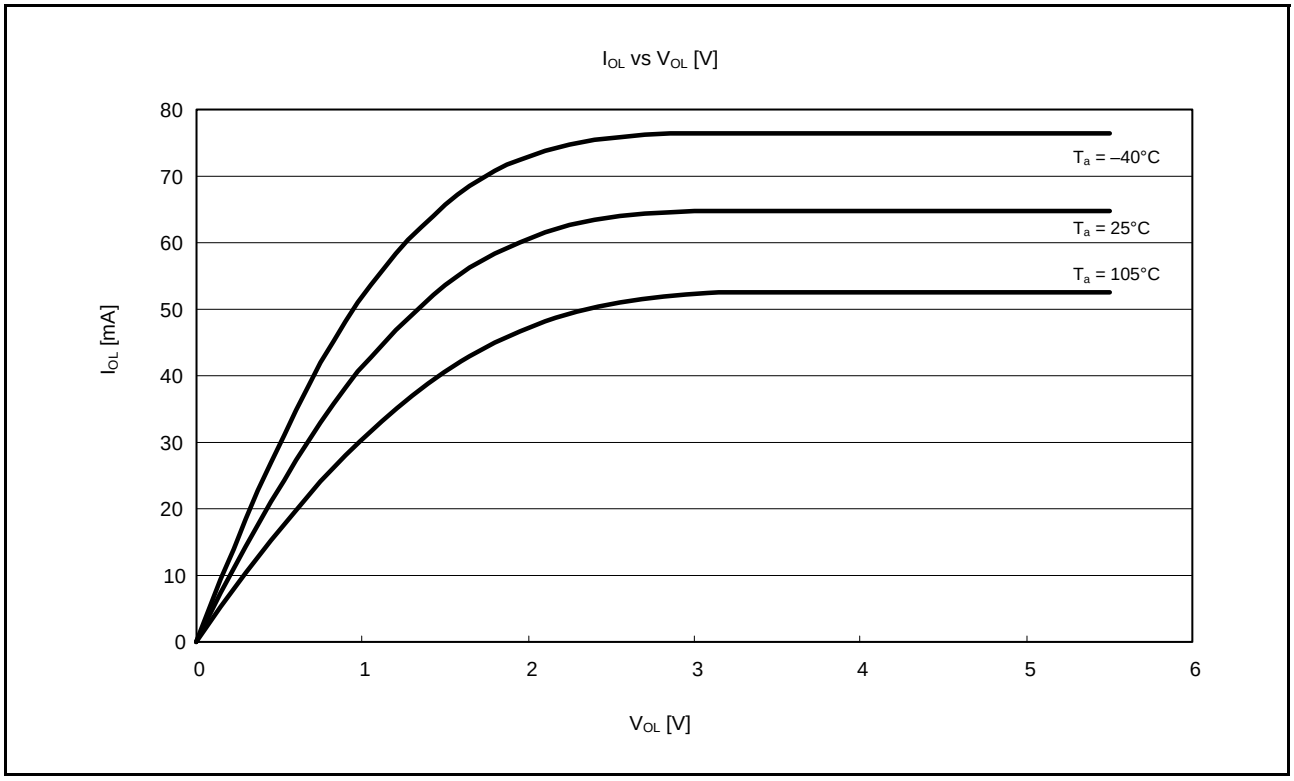


Figure 38.20 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 5.5 V (Reference Data)

38.3 AC Characteristics

Table 38.18 Operation Frequency Value (Medium-Speed Operating Mode 1A)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC			Unit
			1.62 to 1.8 V	1.8 to 2.7 V	2.7 to 5.5 V	
Maximum operating frequency	System clock (ICLK)	f_{max}	8	8	32	MHz
	FlashIF clock (FCLK)*1		8	8	32	
	Peripheral module clock (PCLKB)		8	8	32	
	Peripheral module clock (PCLKD)*2		8	8	32	

Note 1. The VCC is 2.7 to 5.5 V and the FCLK must be running at a frequency of at least 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 38.19 Operation Frequency Value (Medium-Speed Operating Mode 1B)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 =VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC			Unit
			1.62 to 1.8 V	1.8 to 2.7 V	2.7 to 5.5 V	
Maximum operating frequency	System clock (ICLK)	f_{max}	8	8	32	MHz
	FlashIF clock (FCLK)*1		8	8	32	
	Peripheral module clock (PCLKB)		8	8	32	
	Peripheral module clock (PCLKD)*2		8	8	32	

Note 1. The VCC is 1.62 to 3.6 V and the FCLK must be running at a frequency of at least 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 38.20 Operation Frequency Value (Low-Speed Operating Mode 1)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC			Unit
			1.62 to 1.8 V	1.8 to 2.7 V	2.7 to 5.5 V	
Maximum operating frequency	System clock (ICLK)	f_{max}	2	4	8	MHz
	FlashIF clock (FCLK)*1		2	4	8	
	Peripheral module clock (PCLKB)		2	4	8	
	Peripheral module clock (PCLKD)*2		2	4	8	

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 38.21 Operation Frequency Value (Low-Speed Operating Mode 2)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	VCC			Unit
			1.62 to 1.8 V	1.8 to 2.7 V	2.7 to 5.5 V	
Maximum operating frequency	System clock (ICLK)	f_{max}	32.768	32.768	32.768	kHz
	FlashIF clock (FCLK)*1		32.768	32.768	32.768	
	Peripheral module clock (PCLKB)		32.768	32.768	32.768	
	Peripheral module clock (PCLKD)*2		32.768	32.768	32.768	

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The A/D converter cannot be used.

38.3.1 Clock Timing

Table 38.22 Clock Timing

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{EXcyc}	50	—	—	ns	Figure 38.21
EXTAL external clock input high pulse width	t_{EXH}	20	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	20	—	—	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*1	t_{EXWT}	1	—	—	ms	
Main clock oscillator oscillation frequency*2	f_{MAIN}	1	—	20	MHz	
Main clock oscillation stabilization time (crystal)*2	$t_{MAINOSC}$	—	3	—	ms	Figure 38.22
Main clock oscillation stabilization time (ceramic resonator)*2	$t_{MAINOSC}$	—	50	—	µs	
Main clock oscillation stabilization wait time (crystal)*2	$t_{MAINOSCWT}$	—	6	—	ms	
Main clock oscillation stabilization wait time (ceramic resonator)*2	$t_{MAINOSCWT}$	—	100	—	µs	
LOCO clock cycle time	t_{cyc}	7.27	8	8.89	µs	
LOCO clock oscillation frequency	f_{LOCO}	112.5	125	137.5	kHz	
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	20	µs	Figure 38.23
HOCO clock oscillation frequency	f_{HOCO}	31.680	32	32.320	MHz	T_a = 0 to 50°C
		36.495	36.864	37.233		
		39.600	40	40.400		
		49.500	50	50.500		
		31.520	32	32.480		T_a = -40 to 105°C
		36.311	36.864	37.417		
		39.400	40	40.600		
		49.250	50	50.750		
HOCO clock oscillation stabilization time 1	t_{HOCO1}	—	—	50	µs	Figure 38.24
HOCO clock oscillation stabilization time 2	t_{HOCO2}	—	—	10	µs	Figure 38.25
HOCO clock oscillation stabilization wait time	t_{HOCOWT}	—	—	20	µs	Figure 38.25
HOCO clock power supply stabilization time	t_{HOCOP}	—	—	350	µs	Figure 38.26
Sub-clock oscillator oscillation frequency	f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization time*3	t_{SUBOSC}	2	—	—	s	Figure 38.27
Sub-clock oscillation stabilization wait time*3	$t_{SUBOSCWT}$	4	—	—	s	

Note 1. The time interval from the time P36 and P37 are configured for input and the main clock oscillator stopping bit (MOSCCR.MOSTP) is set to 0 (operating) until the clock becomes available.

Note 2. When specifying the main clock oscillator stabilization time, load MOSCWTCR register with a stabilization time value that is greater than the resonator-vendor-recommended value. When determining the main lock oscillation stabilization wait time, allow an adequate margin (2 times is recommended) for the main clock oscillation stabilization time.
Start using the main clock in the main clock oscillation stabilization wait time (tMAINOSCWT) after setting up the main clock oscillator for operation with the MOSCCR.MOSTP bit.
The indicated value is a reference value that is measured for an 8 MHz resonator.

Note 3. When specifying the sub-clock oscillation stabilization time, load SOSCWTCR register with the resonator-vendor-recommended stabilization time value minus 2 seconds. When determining the sub-clock oscillation stabilization wait time, allow an adequate margin (2 times is recommended) for the sub-clock oscillation stabilization time. Start using the sub-clock in the sub-clock oscillation stabilization wait time (tSUBOSCWT) after setting up the sub-clock oscillator for operation with the SOSCCR.SOSTP or RCR3.RTCEN bit.

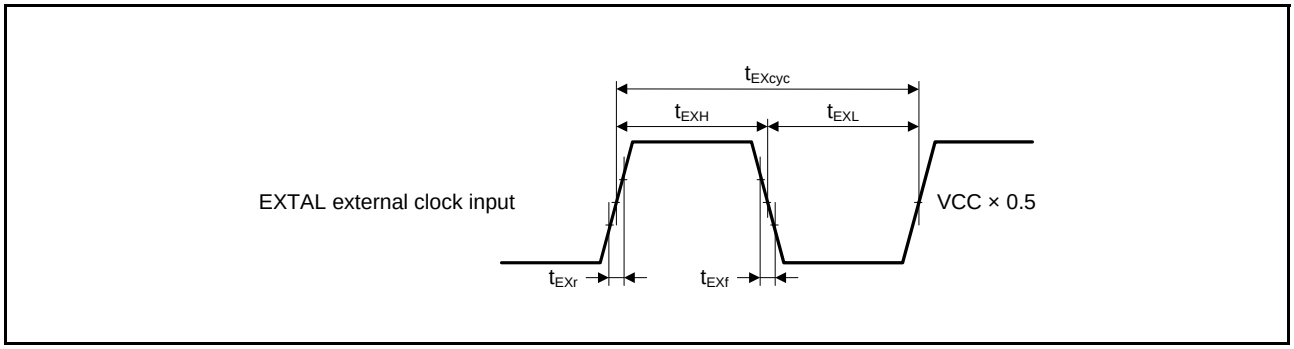


Figure 38.21 EXTAL External Clock Input Timing

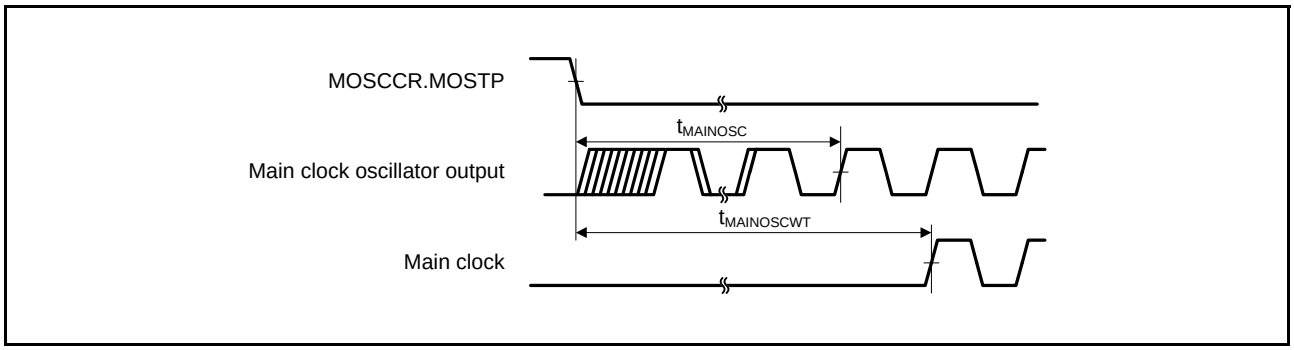


Figure 38.22 Main Clock Oscillation Start Timing

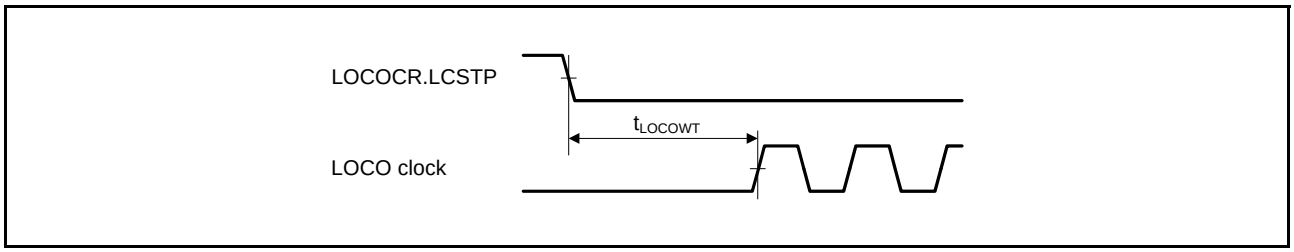


Figure 38.23 LOCO Clock Oscillation Start Timing

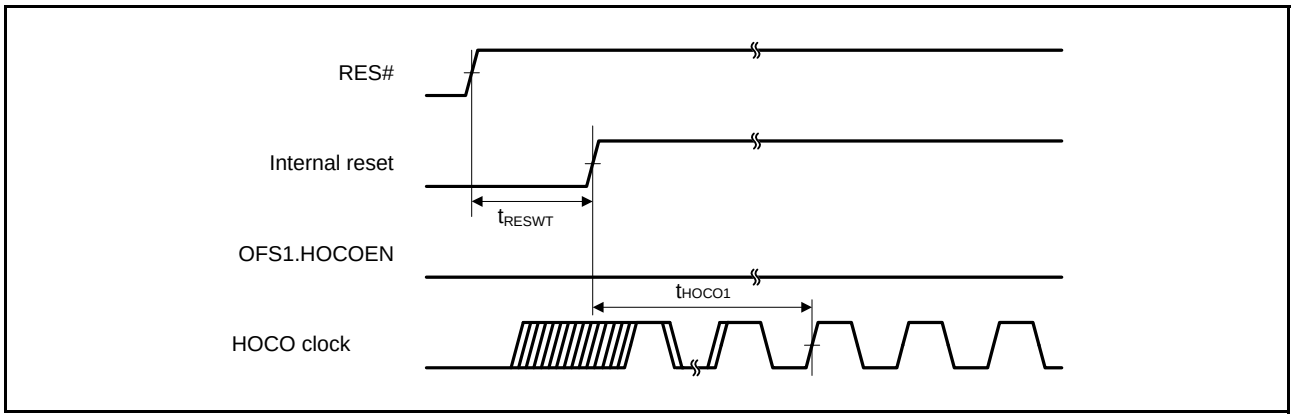


Figure 38.24 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting the OFS1.HOCOEN Bit to 0)

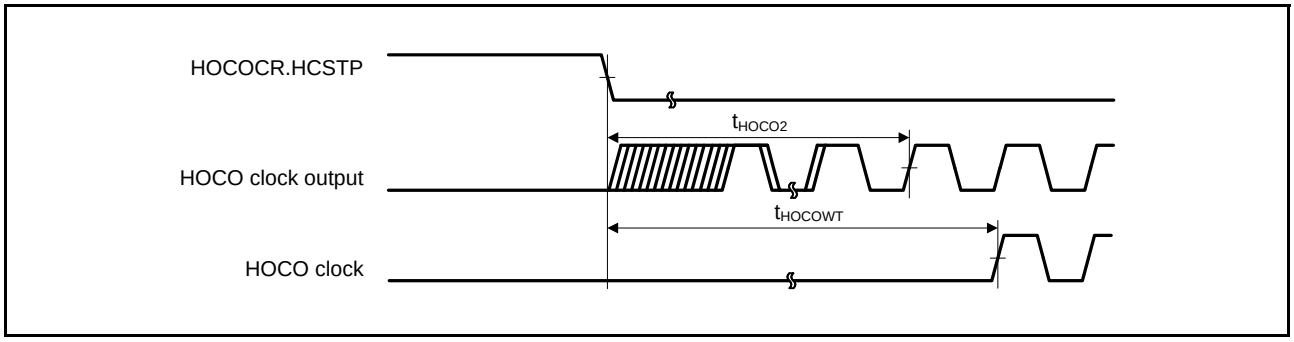


Figure 38.25 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

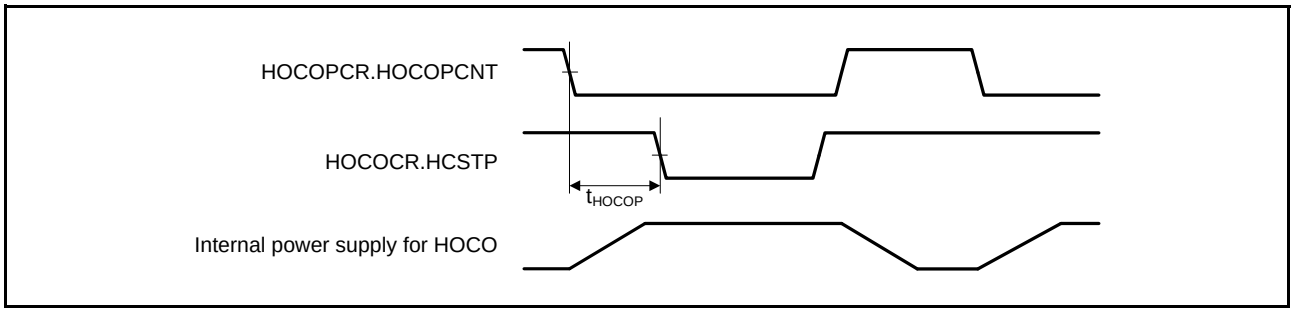


Figure 38.26 HOCO Power Control Timing

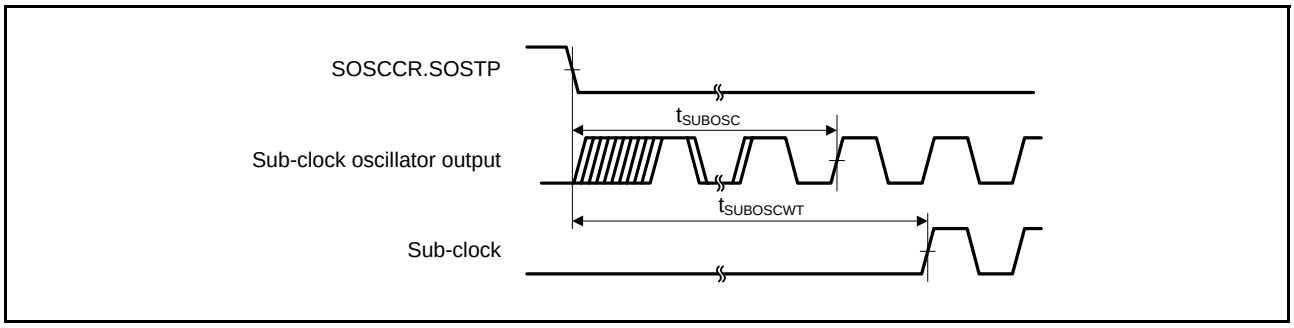


Figure 38.27 Sub-clock Oscillation Start Timing

38.3.2 Reset Timing

Table 38.23 Reset Timing

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	8	—	—	ms	Figure 38.28
	Software standby mode, low-speed operating modes 1 and 2	t_{RESWS}	1	—	—	ms	Figure 38.29
	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}	—	—	912	µs	Figure 38.28
Internal reset time (independent watchdog timer reset, software reset)		t_{RESW2}	—	—	1.4	ms	

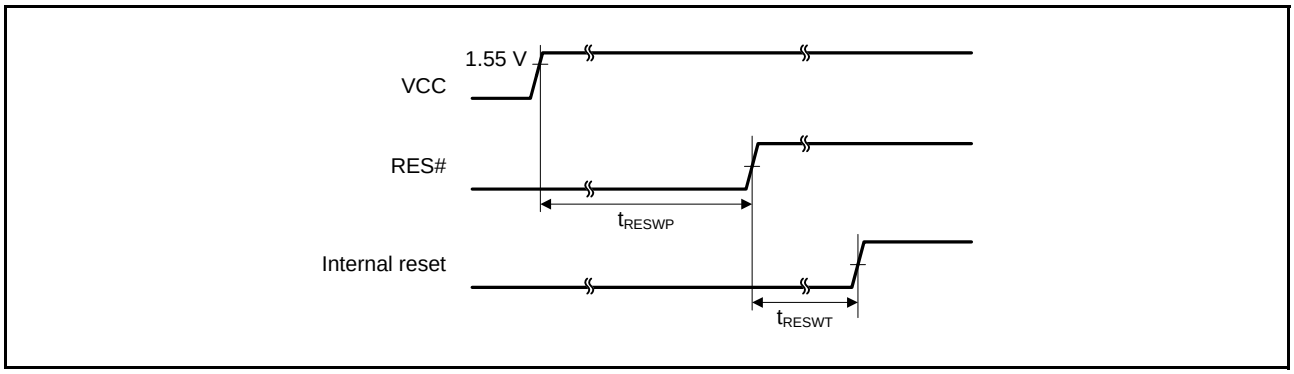


Figure 38.28 Reset Input Timing at Power-On

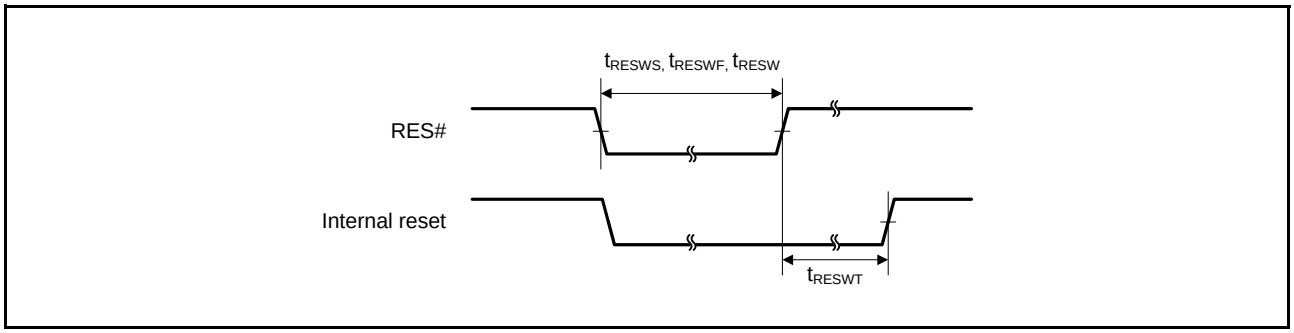


Figure 38.29 Reset Input Timing

38.3.3 Timing of Recovery from Low Power Consumption Modes

Table 38.24 Timing of Recovery from Low Power Consumption Modes

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time after cancellation of software standby mode (HOCO power supplied) (SOFTCUT[2:0] bits = 000b)*1	Crystal resonator connected to main clock oscillator*2	Main clock oscillator operating	t_{SBYMC}	—	3	—	ms	Figure 38.30
	External clock input to main clock oscillator*4	Main clock oscillator operating	t_{SBYEX}	7	—	—	µs	
	Sub-clock oscillator operating*5		t_{SBYSC}	2*3	—	—	s	
	HOCO clock oscillator operating*6		t_{SBYHO}	—	—	50	µs	
	LOCO clock oscillator operating*5		t_{SBYLO}	—	—	90	µs	
Recovery time after cancellation of software standby mode (HOCO power not supplied) (SOFTCUT[2:0] bits = 11xb)*1	Crystal resonator connected to main clock oscillator*2	Main clock oscillator operating	t_{SBYMC}	—	3	—	ms	Figure 38.30
	External clock input to main clock oscillator*4	Main clock oscillator operating	t_{SBYEX}	40	—	—	µs	
	Sub-clock oscillator operating*5		t_{SBYSC}	2*3	—	—	s	
	HOCO clock oscillator operating*6		t_{SBYHO}	—	—	0.8	ms	
	LOCO clock oscillator operating*5		t_{SBYLO}	—	—	90	µs	

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source, and depends on the time set in the wait control registers corresponding to the oscillators.
Note 2. The indicated value is measured for an 8 MHz crystal resonator. ICLK is set to divided by 1.
Note 3. When RCR3.RTCEN = 1, the time will be the time set in the SOSCWTCR register minus 2 s.
Note 4. When the external clock frequency is 20 MHz. ICLK is set to divided by 1.
Note 5. ICLK is set to divided by 1.
Note 6. When the frequency is 50 MHz, HOCOWTCR2.HSTS2[4:0] = 10101b and ICLK is set to divided by 2.
When the frequency is 32 MHz, HOCOWTCR2.HSTS2[4:0] = 10100b and ICLK is set to divided by 1.

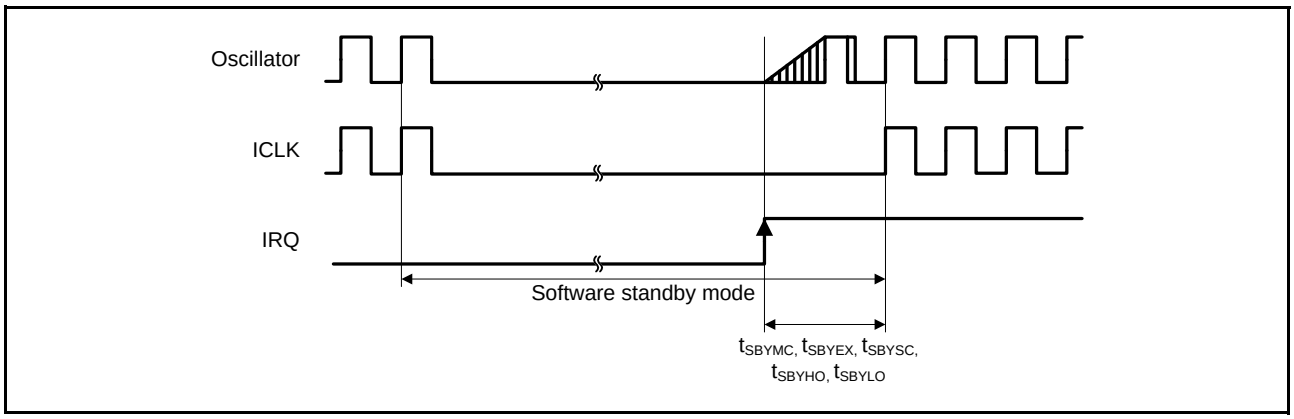


Figure 38.30 Software Standby Mode Cancellation Timing

38.3.4 Control Signal Timing

Table 38.25 Control Signal Timing

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
NMI pulse width	t_{NMIW}	200	—	—	ns	$t_{c(PCLKB)} \times 2 \leq 200$ ns, Figure 38.31
		$t_{c(PCLKB)} \times 2$	—	—	ns	$t_{c(PCLKB)} \times 2 > 200$ ns, Figure 38.31
IRQ pulse width	t_{IRQW}	200	—	—	ns	$t_{c(PCLKB)} \times 2 \leq 200$ ns, Figure 38.32
		$t_{c(PCLKB)} \times 2$	—	—	ns	$t_{c(PCLKB)} \times 2 > 200$ ns, Figure 38.32

Note: • 200 ns minimum in software standby mode.

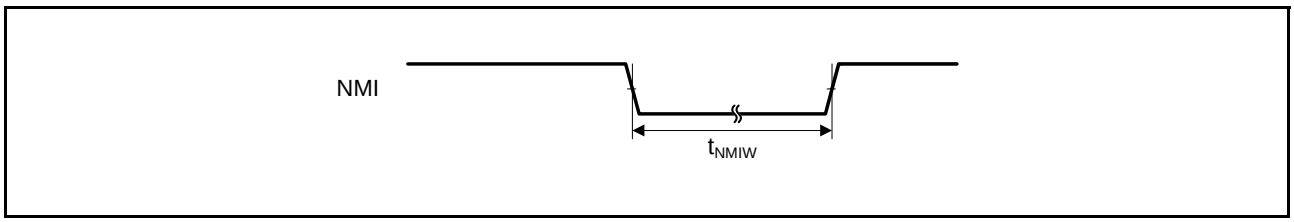


Figure 38.31 NMI Interrupt Input Timing

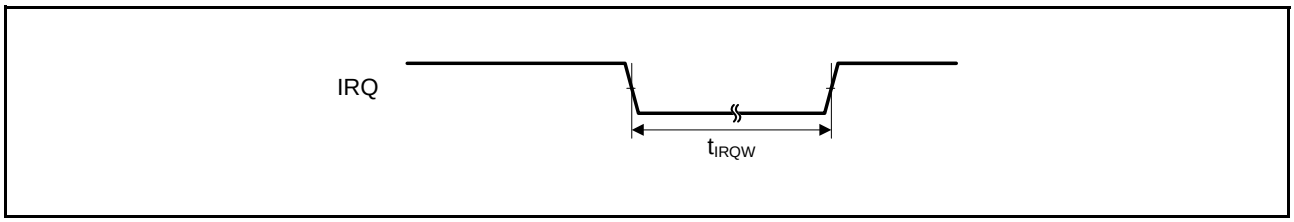


Figure 38.32 IRQ Interrupt Input Timing

38.3.5 Timing of On-Chip Peripheral Modules

Table 38.26 Timing of On-Chip Peripheral Modules (1)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width			t_{PRW}	1.5	—	t_{Pcyc}	Figure 38.33
MTU	Input capture input pulse width		Single-edge setting	t_{TICW}	1.5	—	t_{Pcyc}	Figure 38.34
			Both-edge setting		2.5	—		
	Timer clock pulse width		Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—	t_{Pcyc}	Figure 38.35
			Both-edge setting		2.5	—		
			Phase counting mode		2.5	—		
POE	POE# input pulse width			t_{POEW}	1.5	—	t_{Pcyc}	Figure 38.36
8-bit timer	Timer clock pulse width		Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{Pcyc}	Figure 38.37
			Both-edge setting		2.5	—		
SCI	Input clock cycle		Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 38.38
			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*2		Asynchronous	t_{Scyc}	16	—	t_{Pcyc}	C = 30 pF Figure 38.39
			Clock synchronous		4	—		
	Output clock pulse width*2			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time*2			t_{SCKr}	—	20	ns	
	Output clock fall time*2			t_{SCKf}	—	20	ns	
	Transmit data delay time*3	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{TXD}	—	40	ns	
			1.62 V ≤ VCC < 2.7 V		—	80		
	Receive data setup time	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{RXS}	40	—	ns	
			1.62 V ≤ VCC < 2.7 V		80	—		
	Receive data hold time		Clock synchronous	t_{RXH}	40	—	ns	
A/D converter	Trigger input pulse width			t_{TRGW}	1.5	—	t_{Pcyc}	Figure 38.40
CAC	CACREF input pulse width		$t_{Pcyc} \le t_{cac}$*4	t_{CACREF}	4.5 t_{cac} + 3 t_{Pcyc}	—	ns	
			$t_{Pcyc} > t_{cac}$*4		5 t_{cac} + 6.5 t_{Pcyc}			

Note 1. t_{Pcyc}: PCLKB cycle
Note 2. Value when the drive capacity of clock output ports is set to normal output.
Note 3. Value when the drive capacity of data output ports is set to normal output.
Note 4. t_{cac}: CAC count clock source cycle

Table 38.27 Timing of On-Chip Peripheral Modules (2)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
RSPI	RSPCK clock cycle*2	Master		t_{SPcyc}	2	4096	t_{Pcyc}	C = 30 pF Figure 38.41
		Slave			8	4096		
	RSPCK clock high pulse width*2	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*2	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*2	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	—	10	ns	
			1.62 V ≤ VCC < 2.7 V		—	20		
		Input			—	1	µs	
	Data input setup time	Master		t_{SU}	4	—	ns	C = 30 pF Figure 38.42 to Figure 38.47
		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	t_{HF}	0	—		
		Slave		t_H	$20 + 2 \times t_{Pcyc}$	—		
	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	2.7 V ≤ VCC ≤ 5.5 V	t_{OD}	—	14	ns	
			1.62 V ≤ VCC < 2.7 V		—	28		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		—	$3 \times t_{Pcyc} + 40$		
			1.62 V ≤ VCC < 2.7 V		—	$3 \times t_{Pcyc} + 80$		
	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	2.7 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	—	10	ns	
			1.62 V ≤ VCC < 2.7 V		—	20		
		Input			—	1	µs	
	SSL rise/fall time	Output		t_{SSLr}, t_{SSLf}	—	20	ns	
		Input			—	1	µs	
	Slave access time			t_{SA}	—	4	t_{Pcyc}	C = 30 pF Figure 38.45 and Figure 38.47
	Slave output release time			t_{REL}	—	3	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLKB cycle
Note 2. Value when the drive capacity of clock output ports is set to normal output.

Table 38.28 Timing of On-Chip Peripheral Modules (3)

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)*2		t_{SPcyc}	4	65536	t_{Pcyc}	C = 30 pF Figure 38.41
	SCK clock cycle input (slave)			6	65536		
	SCK clock high pulse width*2		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width*2		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time		t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time	2.7 V ≤ VCC ≤ 5.5 V	t_{SU}	40	—	ns	C = 30 pF Figure 38.42 to Figure 38.47
		1.62 V ≤ VCC < 2.7 V		80	—		
	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	2.7 V ≤ VCC ≤ 5.5 V	t_{OD}	—	40	ns	
		1.62 V ≤ VCC < 2.7 V		—	80		
	Data output hold time		t_{OH}	0	—	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}	C = 30 pF Figure 38.45 and Figure 38.47
	Slave output release time		t_{REL}	—	5	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLKB cycle
Note 2. Value when the drive capacity of clock output ports is set to normal output.

Table 38.29 Timing of On-Chip Peripheral Modules (4)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, fPCLKB = up to 32 MHz, T_a = –40 to +105°C

Item		Symbol	Min.*1,*2	Max.	Unit	Test Conditions
RIIC (Standard mode, SMBus)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 38.48
	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 (5) × 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	Figure 38.48
	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 ICMR3.NF[1:0] are set to 11b while a digital filter is enabled with ICFER.NFE = 1.
Note 2. C_b indicates the total capacity of the bus line.

Table 38.30 Timing of On-Chip Peripheral Modules (5)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, fPCLKB = up to 32 MHz, T_a = –40 to +105°C

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
Simple IIC (Standard mode)	SDA input rise time	t_{Sr}	—	1000	ns	Figure 38.48
	SDA input fall time	t_{Sf}	—	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	$4 \times t_{Pcyc}$*2	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	Figure 38.48
	SCL, SDA input fall time	t_{Sf}	$20 + 0.1C_b$	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	$4 \times t_{Pcyc}$*2	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: • t_{Pcyc}: PCLKB cycle

Note 1. C_b indicates the total capacity of the bus line.

Note 2. This applies when the SMR.CKS[1:0] bits = 00b and the SNFR.NFCS[2:0] bits = 010b while the SNFR.NFE bit = 1 and the digital filter is enabled.

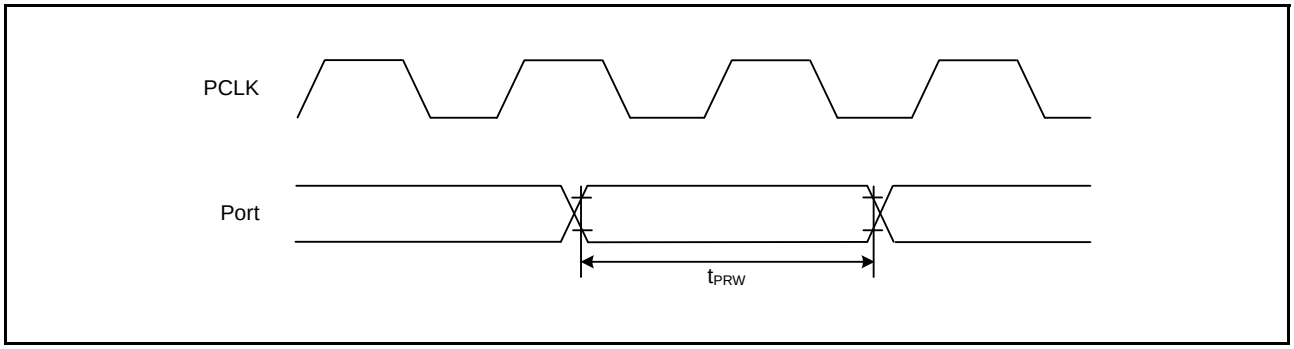


Figure 38.33 I/O Port Input Timing

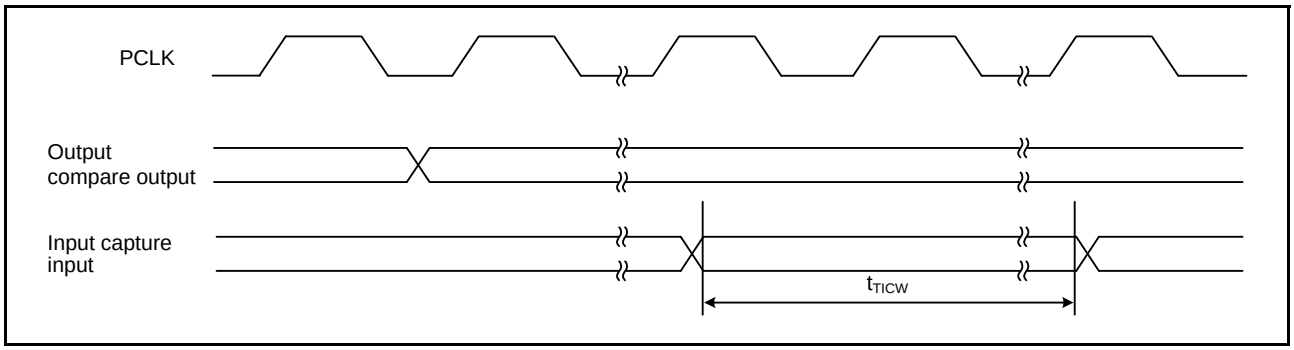


Figure 38.34 MTU Input/Output Timing

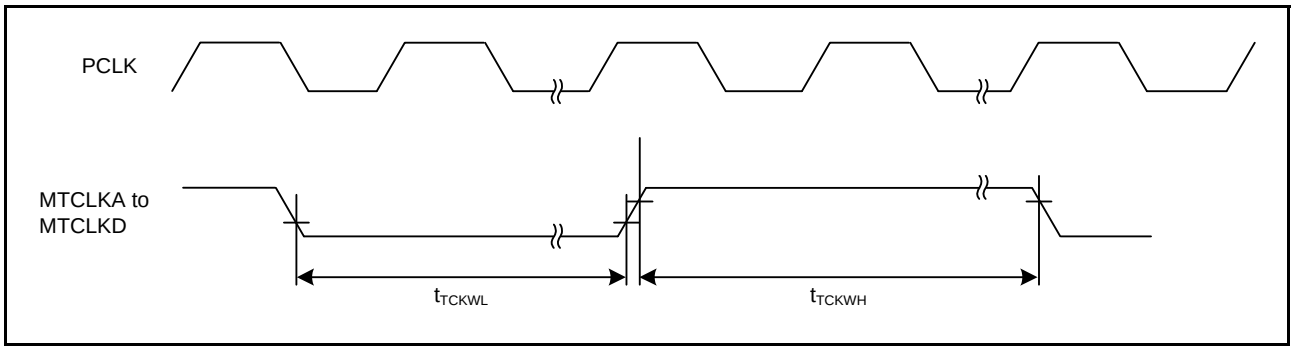


Figure 38.35 MTU Clock Input Timing

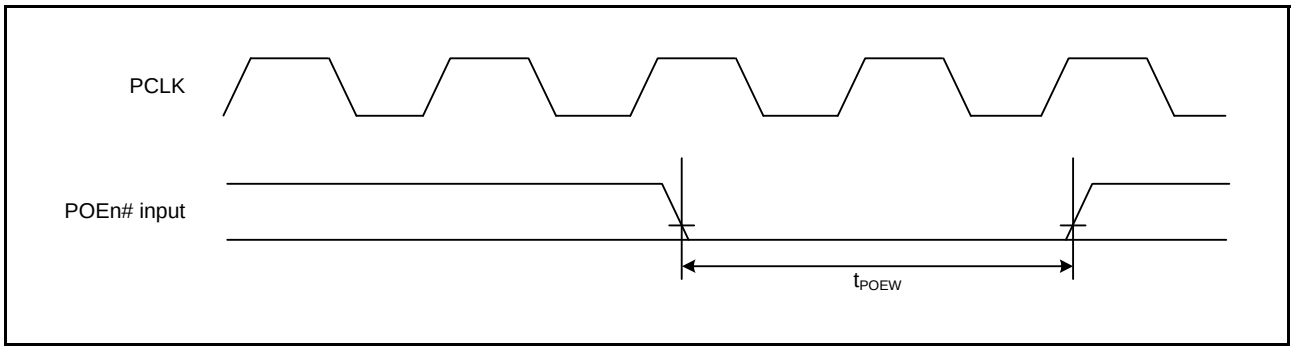


Figure 38.36 POE# Input Timing

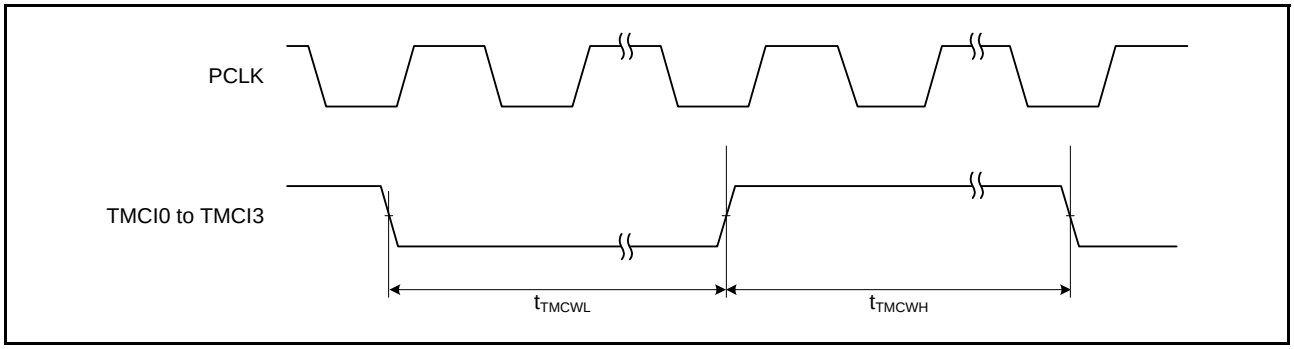


Figure 38.37 8-Bit Timer Clock Input Timing

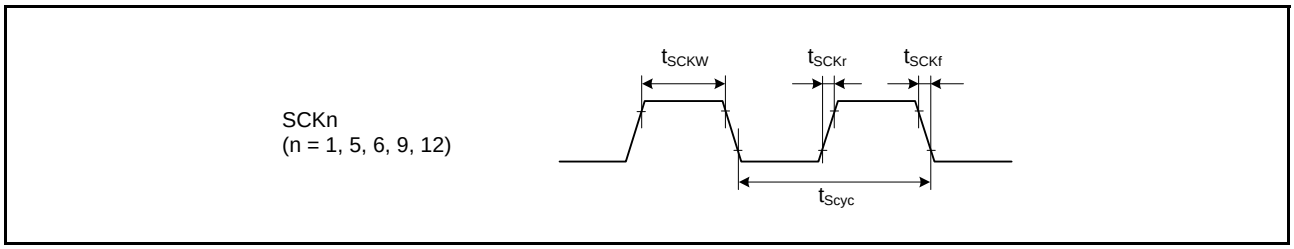


Figure 38.38 SCK Clock Input Timing

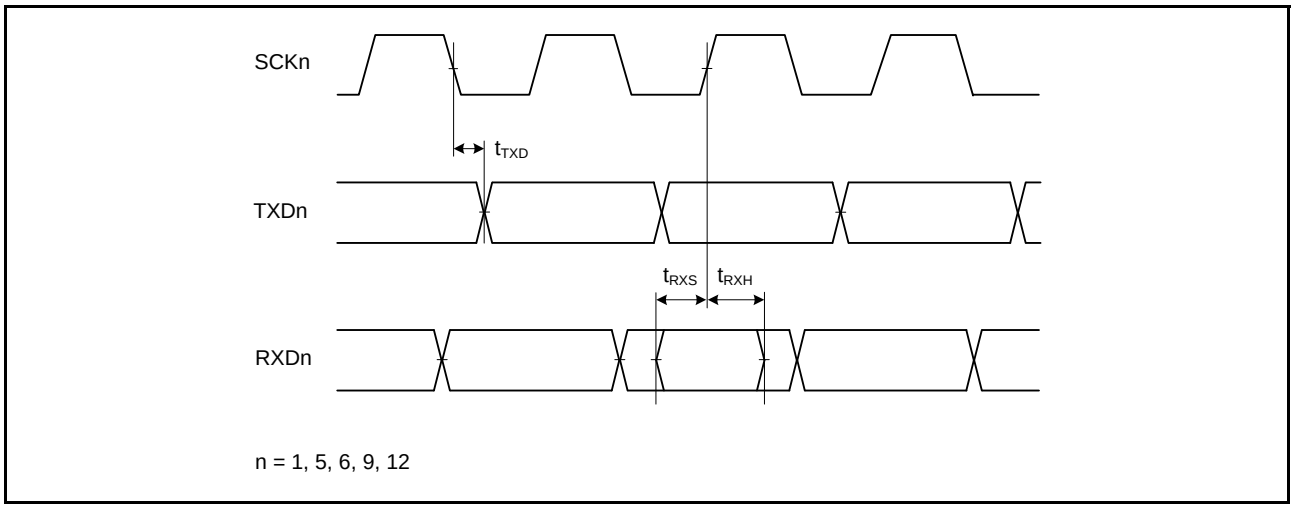


Figure 38.39 SCI Input/Output Timing: Clock Synchronous Mode

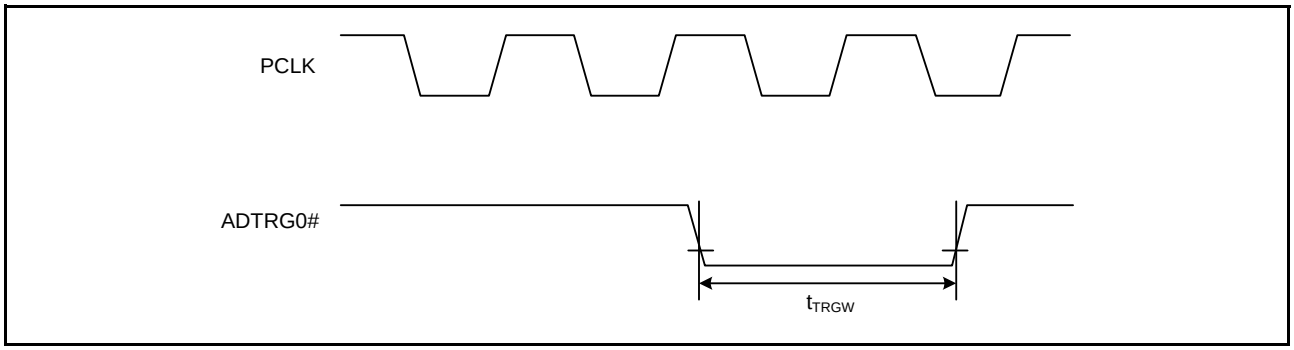


Figure 38.40 A/D Converter External Trigger Input Timing

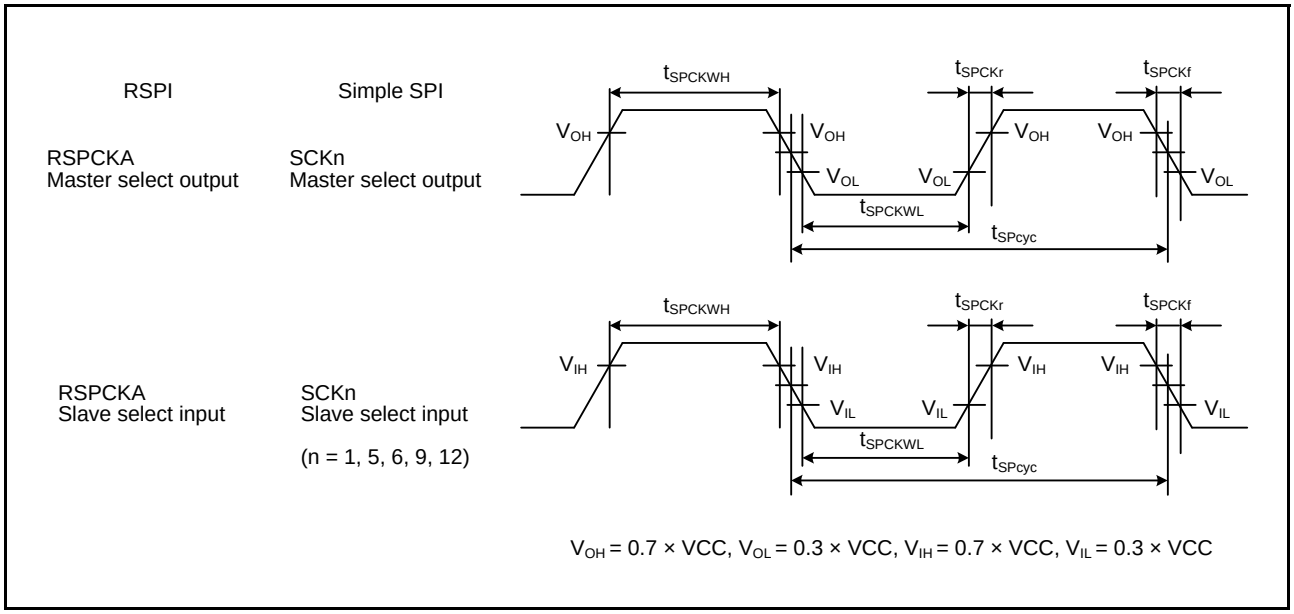


Figure 38.41 RSPI Clock Timing and Simple SPI Clock Timing

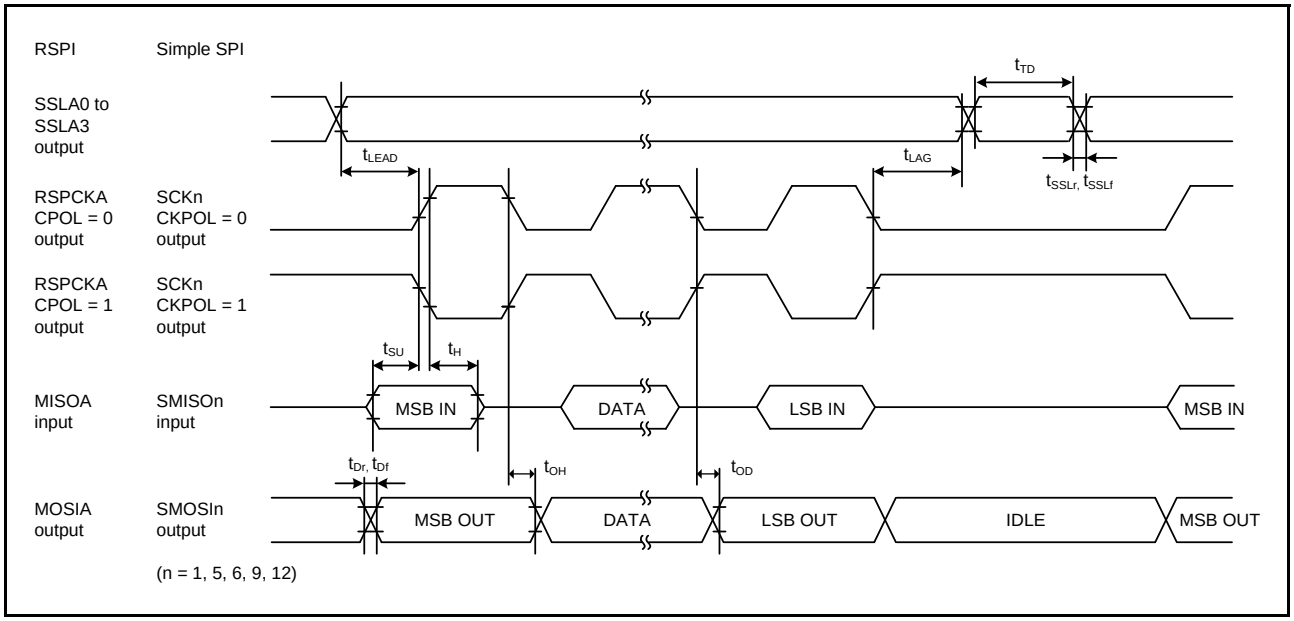


Figure 38.42 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 1)

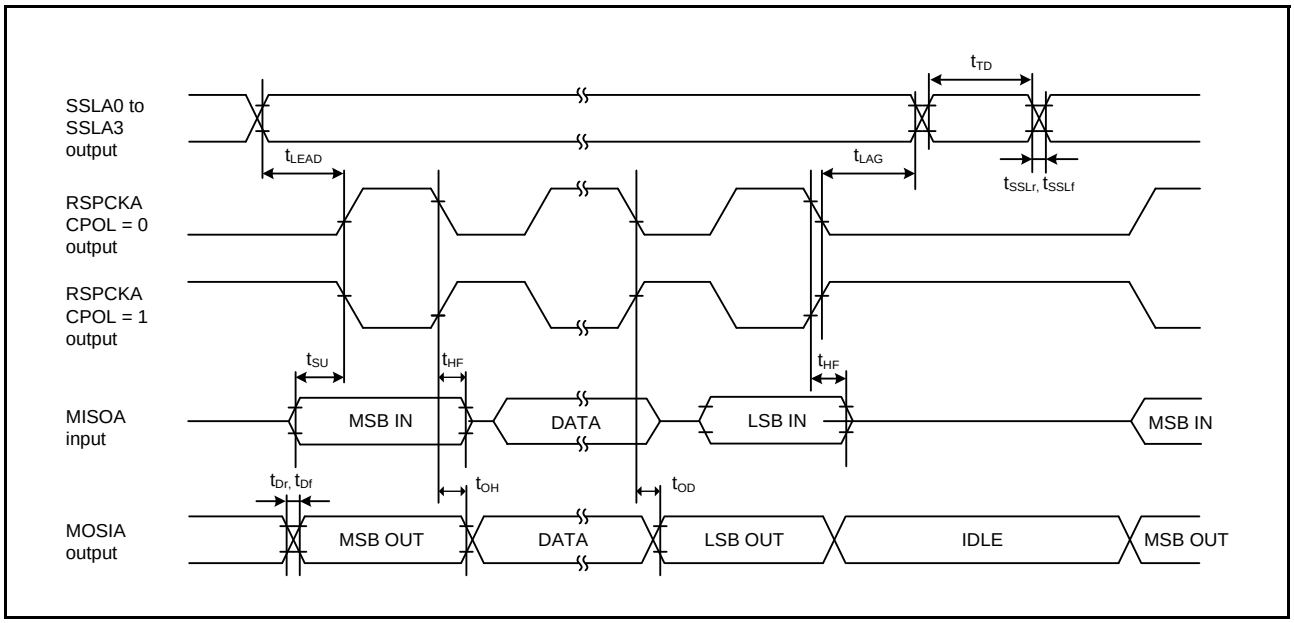


Figure 38.43 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Divided by 2)

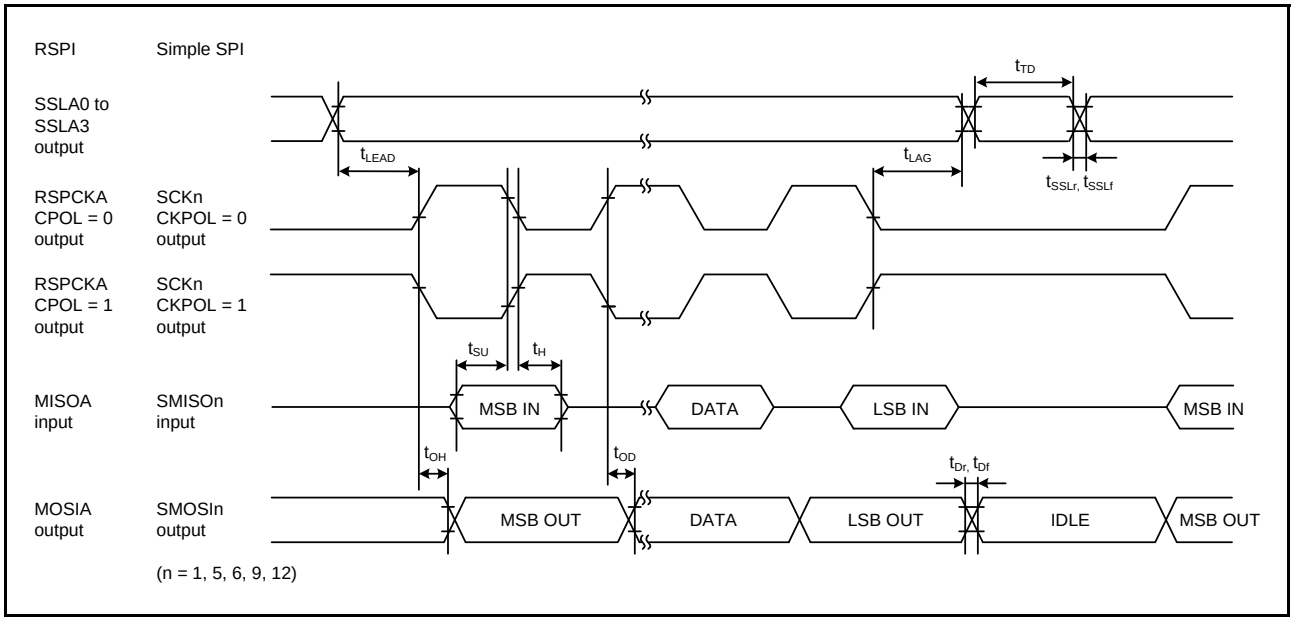


Figure 38.44 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 0)

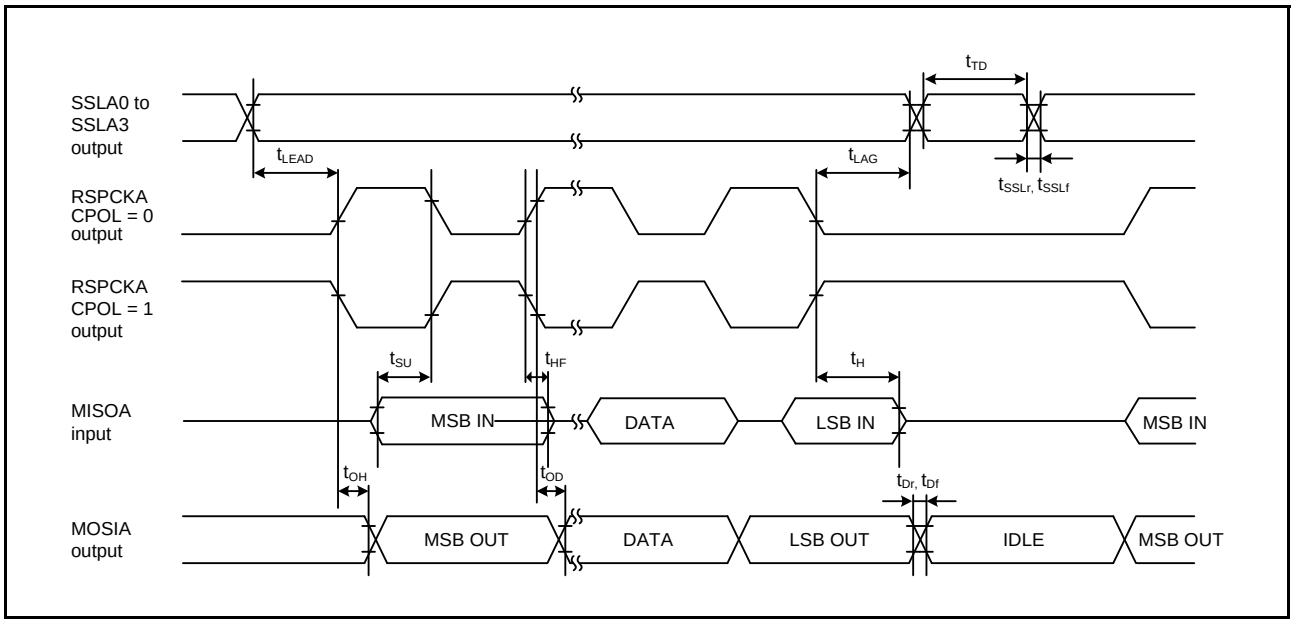


Figure 38.45 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Divided by 2)

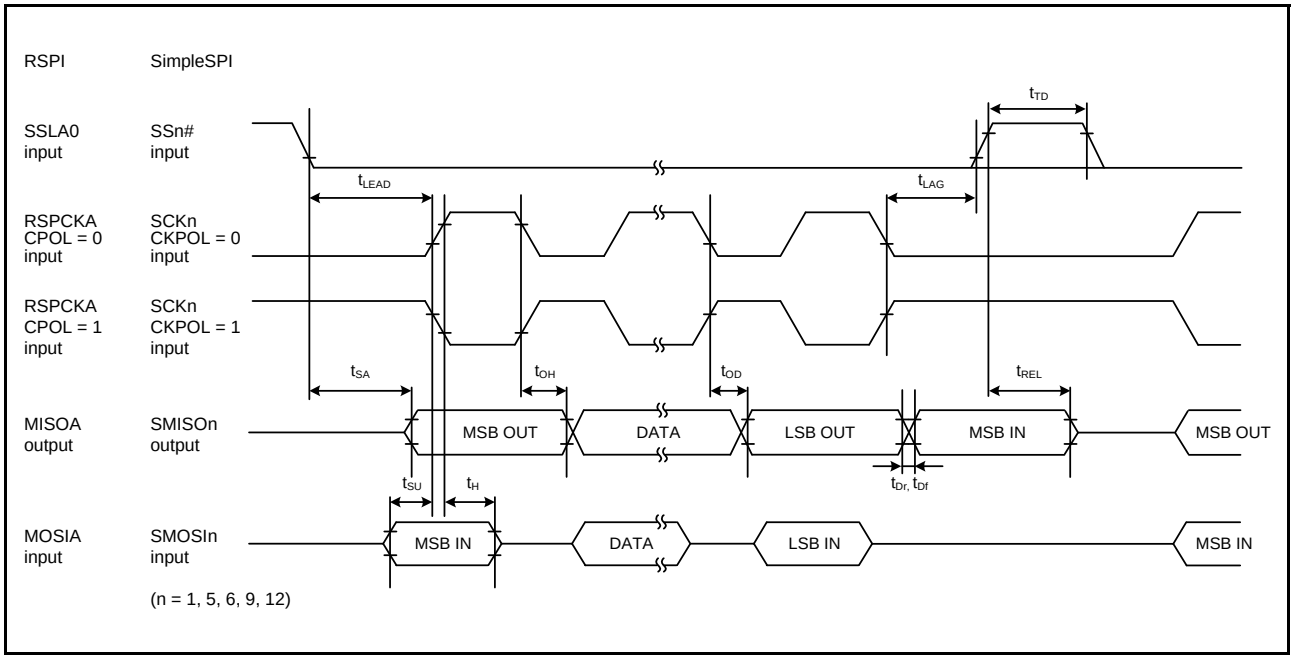


Figure 38.46 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)

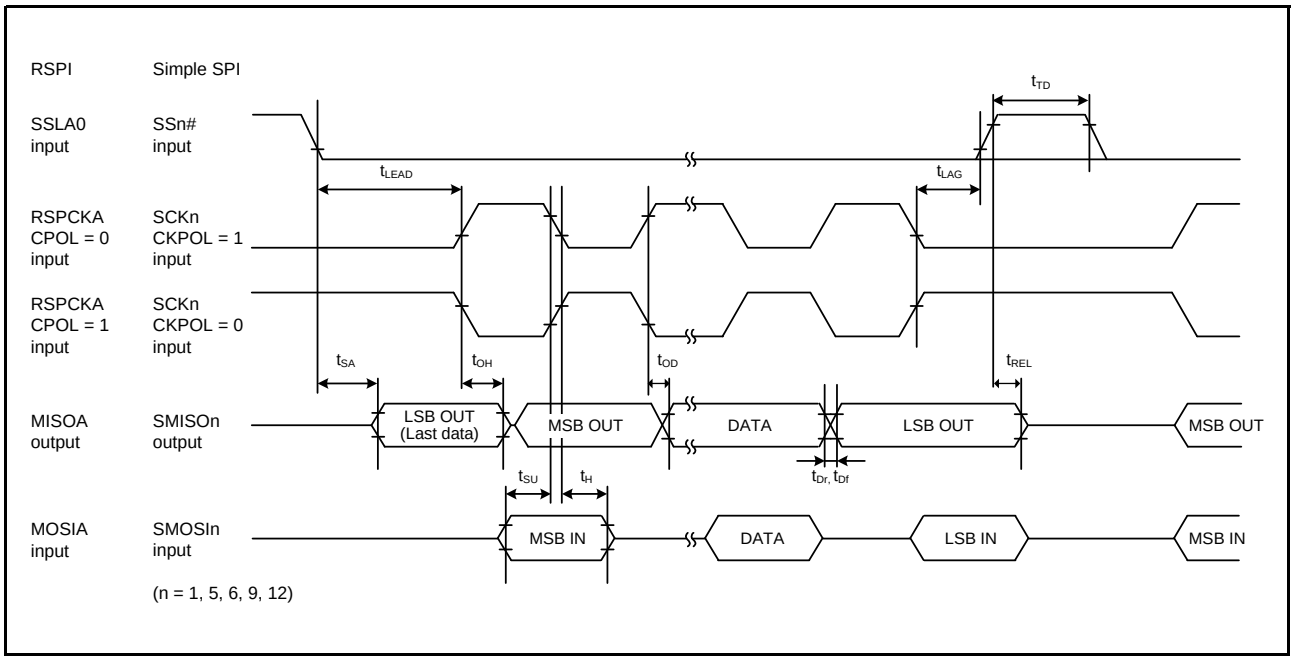


Figure 38.47 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)

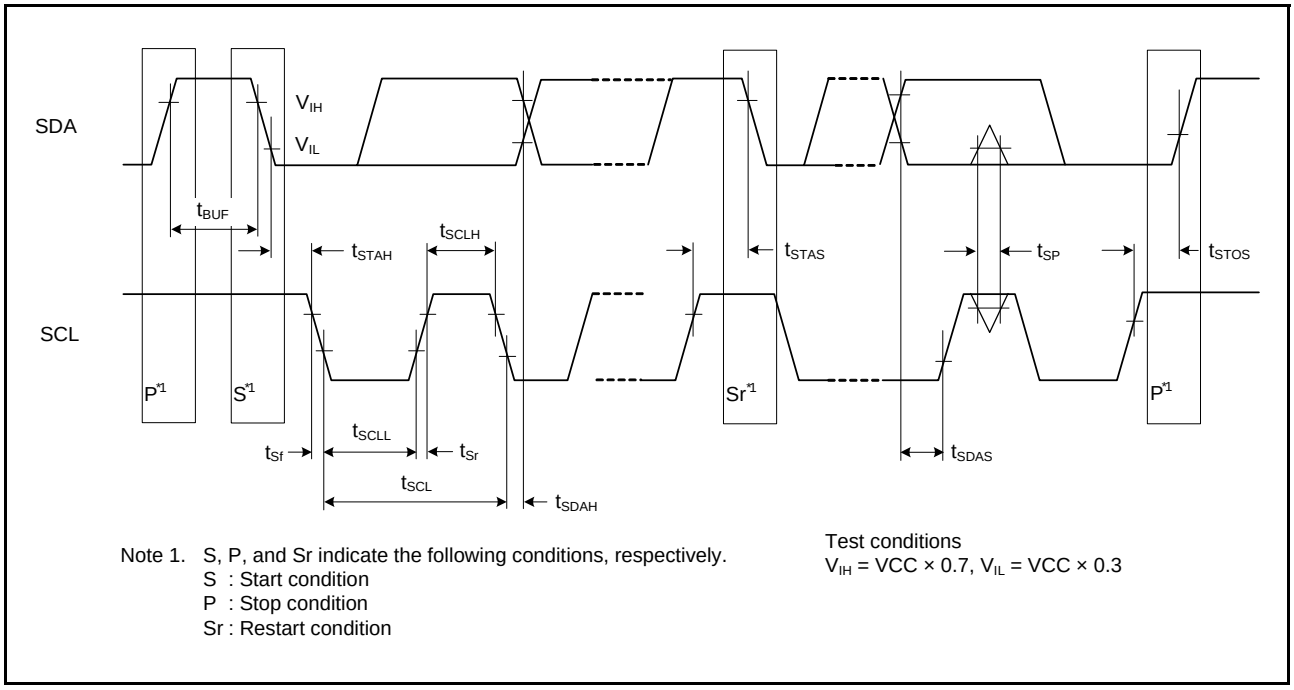


Figure 38.48 RIIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing

38.4 A/D Conversion Characteristics

Table 38.31 A/D Conversion Characteristics (1)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, 2.7 ≤ VREFH0 ≤ 5.5 V, AVCC0 – 0.9 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
A/D conversion clock frequency (fPCLKD)		1	—	32	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at fPCLKD = 32 MHz)	Permissible signal source impedance (Max.) = 1 kΩ	1.56 (0.652)*2	—	—	µs	Sampling in 20 states
	Permissible signal source impedance (Max.) = 5 kΩ	3.29 (2.35)*2	—	—		Sampling in 75 states
Analog input capacitance		—	—	30	pF	
Offset error		—	±0.5	±4.5	LSB	High-precision channel
				±7.5		Normal-precision channel
Full-scale error		—	±0.75	±4.5	LSB	High-precision channel
				±7.5		Normal-precision channel
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±1.25	±5.0	LSB	High-precision channel
		—	±1.25	±8.0	LSB	Normal-precision channel
DNL differential nonlinearity error		—	±1.0	—	LSB	
INL integral nonlinearity error		—	±1.0	±3.0	LSB	High-precision channel
		—	±1.0	±5.0	LSB	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 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 38.32 Channel Classification for A/D Converter

Classification	Channel	
High-precision channel	AN000 to AN007	It is disallowed to use pins AN000 to AN007 as digital outputs when the A/D converter is used.
Normal-precision channel	AN008 to AN015	

Table 38.33 A/D Internal Reference Voltage Characteristics

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	1.35	1.50	1.65	V	

Table 38.34 A/D Conversion Characteristics (2)

Conditions: VCC = AVCC0 = 1.62 to 3.6 V, 1.62 ≤ VREFH0 ≤ 2.7 V, AVCC0 – 0.9 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
A/D conversion clock frequency (fPCLKD)		1	—	8	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at fPCLKD = 8 MHz)	Permissible signal source impedance (Max.) = 1 kΩ	5.25 (1.5)*2	—	—	µs	Sampling in 12 states
	Permissible signal source impedance (Max.) = 5 kΩ	6.25 (2.5)*2	—	—		Sampling in 20 states
Analog input capacitance		—	—	30	pF	
Offset error		—	±0.5	±7.5	LSB	
Full-scale error		—	±1.25	±7.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±3.0	±8.0	LSB	
DNL differential nonlinearity error		—	±1.25	—	LSB	
INL integral nonlinearity error		—	±1.5	±5.0	LSB	

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 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 38.35 Sampling Time

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Typ.	Unit	Test Conditions
Sampling time	High-precision channel	Ts	0.208 + 0.417 × R0 (kΩ)	µs	Figure 38.49
	Normal-precision channel				

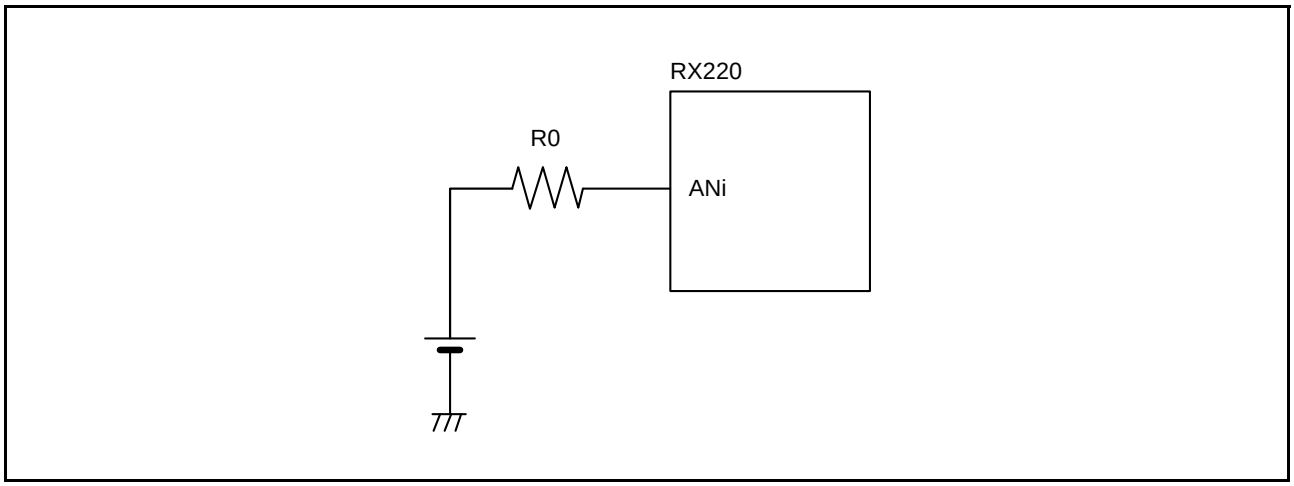


Figure 38.49 Internal Equivalent Circuit of Analog Input Pin

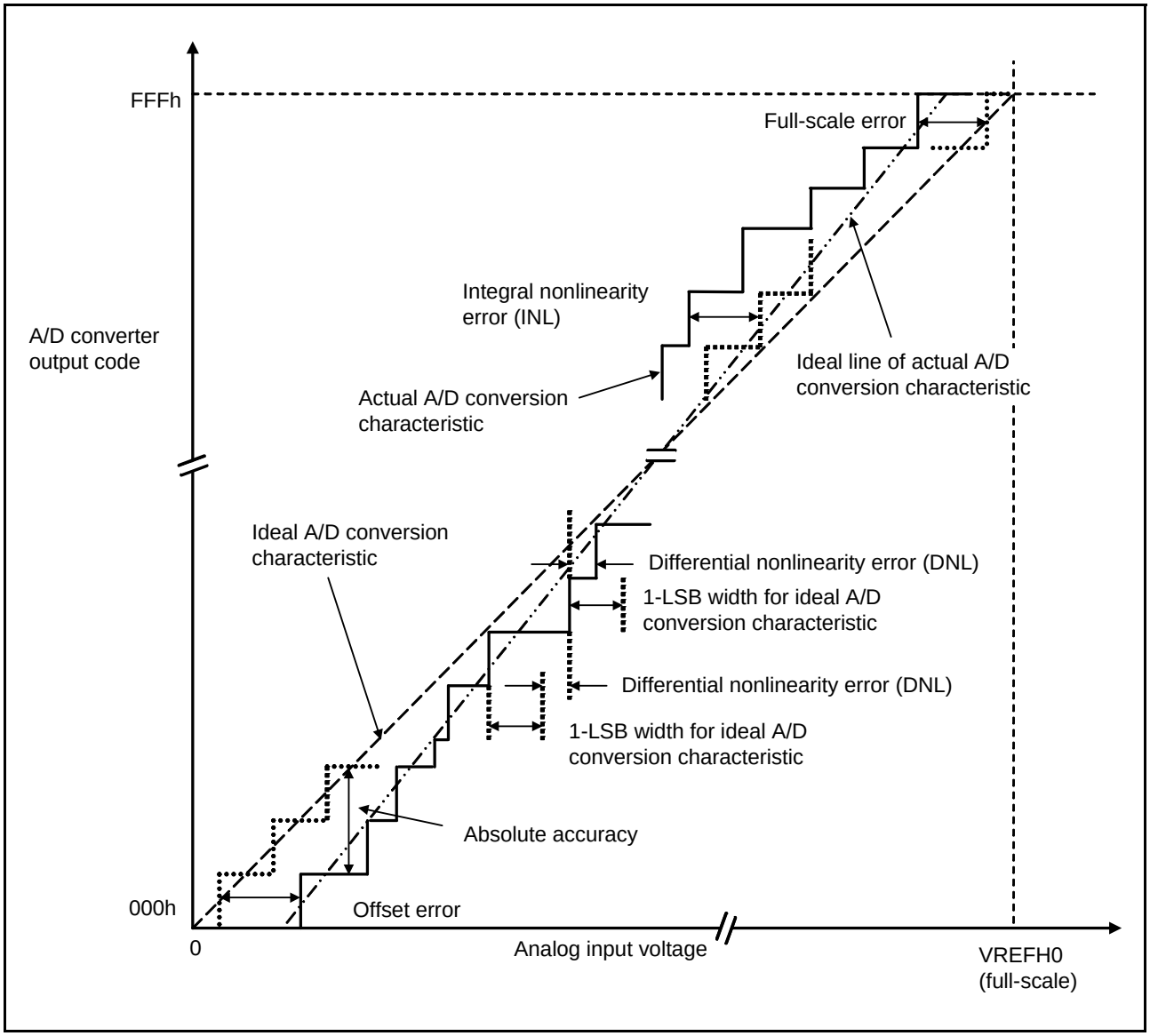


Figure 38.50 Illustration of A/D Converter Characteristic Terms

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.

38.5 Comparator Characteristics

Table 38.36 Comparator Characteristics

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Comparator A	External standard voltage input range	LVREF	1.4	—	VCC	V	
	External comparison voltage*1 (CMPA1, CMPA2) input range	VI	–0.3	—	VCC + 0.3	V	
	Offset	—	—	±50	±150	mV	
	Comparator output delay time*2	—	—	3	—	µs	At falling edge VI = LVREF – 110 mV
			—	2	—	µs	At falling edge VI < LVREF – 1 V
			—	3	—	µs	At rising edge VI = LVREF + 160 mV
			—	1.5	—	µs	At rising edge VI > LVREF + 1 V
	Comparator operating current	ICMPA	—	0.5	—	µA	VCC = 5.0 V

Note 1. VCC does not include ripple.
Note 2. When the digital filter is disabled.

38.6 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 38.37 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (1)

Conditions: VCC = AVCC, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	Low power consumption function disabled*1	V_{POR}	1.30	1.40	1.55	V	Figure 38.51 and Figure 38.52
		Low power consumption function enabled*2		1.00	1.20	1.45		
	Voltage detection circuit (LVD0)*3		V_{det0_0}	3.65	3.80	3.95	V	Figure 38.53
			V_{det0_1}	2.70	2.80	2.90		
			V_{det0_2}	1.80	1.90	2.00		
			V_{det0_3}	1.62	1.72	1.82		
	Voltage detection circuit (LVD1)*4		V_{det1_0}	4.00	4.15	4.30	V	Figure 38.54
			V_{det1_1}	3.85	4.00	4.15		
			V_{det1_2}	3.70	3.85	4.00		At falling edge VCC
			V_{det1_3}	3.55	3.70	3.85		
			V_{det1_4}	3.40	3.55	3.70		
			V_{det1_5}	3.25	3.40	3.55		
			V_{det1_6}	3.10	3.25	3.40		
			V_{det1_7}	2.95	3.10	3.25		
			V_{det1_8}	2.85	2.95	3.05		
			V_{det1_9}	2.70	2.80	2.90		
			V_{det1_A}	2.55	2.65	2.75		
			V_{det1_B}	2.40	2.50	2.60		
			V_{det1_C}	2.25	2.35	2.45		
			V_{det1_D}	2.10	2.20	2.30		
			V_{det1_E}	1.95	2.05	2.15		
			V_{det1_F}	1.80	1.90	2.00		

Note: • These characteristics apply when noise is not superimposed on the power supply.
Note 1. When the CPU is in a mode other than software standby mode, when the CPU transits to software standby mode with the FHSSBYCR.SOFTCUT[2] bit set to 0.
Note 2. When the CPU transits to software standby mode with the FHSSBYCR.SOFTCUT[2] bit set to 1.
Note 3. # in the symbol Vdet0_# denotes the value of the OFS1.VDSEL[1:0] bits.
Note 4. # in the symbol Vdet1_# denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

Table 38.38 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (2)

Conditions: VCC = AVCC0, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Voltage detection circuit (LVD2)*1	V_{det2_0}	4.00	4.15	4.30	V	Figure 38.55
		V_{det2_1}	3.85	4.00	4.15		At falling edge VCC
		V_{det2_2}	3.70	3.85	4.00		
		V_{det2_3}	3.55	3.70	3.85		
		V_{det2_4}	3.40	3.55	3.70		
		V_{det2_5}	3.25	3.40	3.55		
		V_{det2_6}	3.10	3.25	3.40		
		V_{det2_7}	2.95	3.10	3.25		
		V_{det2_8}	2.85	2.95	3.05		
		V_{det2_9}	2.70	2.80	2.90		
		V_{det2_A}	2.55	2.65	2.75		
		V_{det2_B}	2.40	2.50	2.60		
		V_{det2_C}	2.25	2.35	2.45		
		V_{det2_D}	2.10	2.20	2.30		
		V_{det2_E}	1.95	2.05	2.15		
		V_{det2_F}	1.80	1.90	2.00		
		V_{CMPA2}	1.18	1.33	1.48		EXVCCINP2 = 1
Internal reset time	Power-on reset time	t_{POR}	—	9	—	ms	Figure 38.52
	Voltage monitoring 0 reset time	t_{LVD0}	—	9	—		Figure 38.53
	Voltage monitoring 1 reset time	t_{LVD1}	—	1.4	—		Figure 38.54
	Voltage monitoring 2 reset time	t_{LVD2}	—	1.4	—		Figure 38.55
Minimum VCC down time*2		t_{VOFF}	200	—	—	µs	Figure 38.51
Response delay time		t_{det}	—	—	200	µs	Figure 38.52
LVD operation stabilization time (after LVD is enabled)		$Td_{(E-A)}$	—	—	15	µs	Figure 38.54 and Figure 38.55
Power-on reset enable time		$t_{W(POR)}$	1	—	—	ms	Figure 38.52 VCC = 0.9 V or lower
Hysteresis width (LVD1 and LVD2)		V_{LVH}	—	100	—	mV	When selection is from among VdetX_0 to 7.
			—	50	—		When selection is from among VdetX_8 to F.

Note: • These characteristics apply when noise is not superimposed on the power supply.

Note 1. # in the symbol Vdet2_# denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.

Note 2. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for the POR/ LVD.

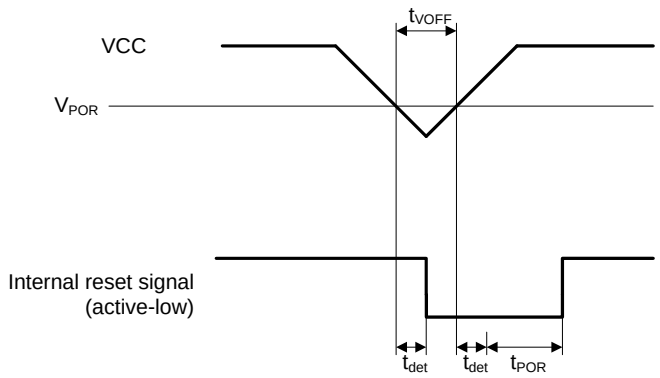
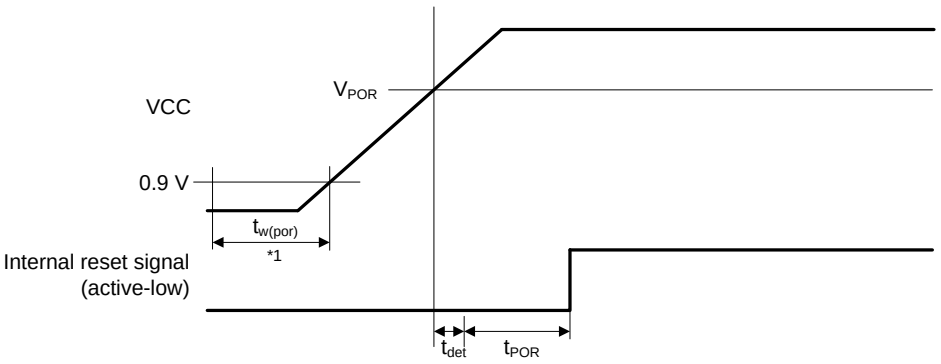


Figure 38.51 Voltage Detection Reset Timing



Note 1. $t_{w(por)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (0.9 V).
When VCC turns on, maintain $t_{w(por)}$ for 1 ms or more.

Figure 38.52 Power-on Reset Timing

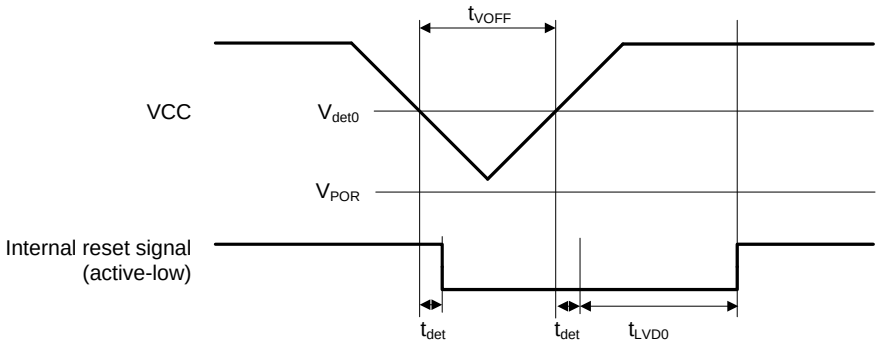


Figure 38.53 Voltage Detection Circuit Timing (V_{det0})

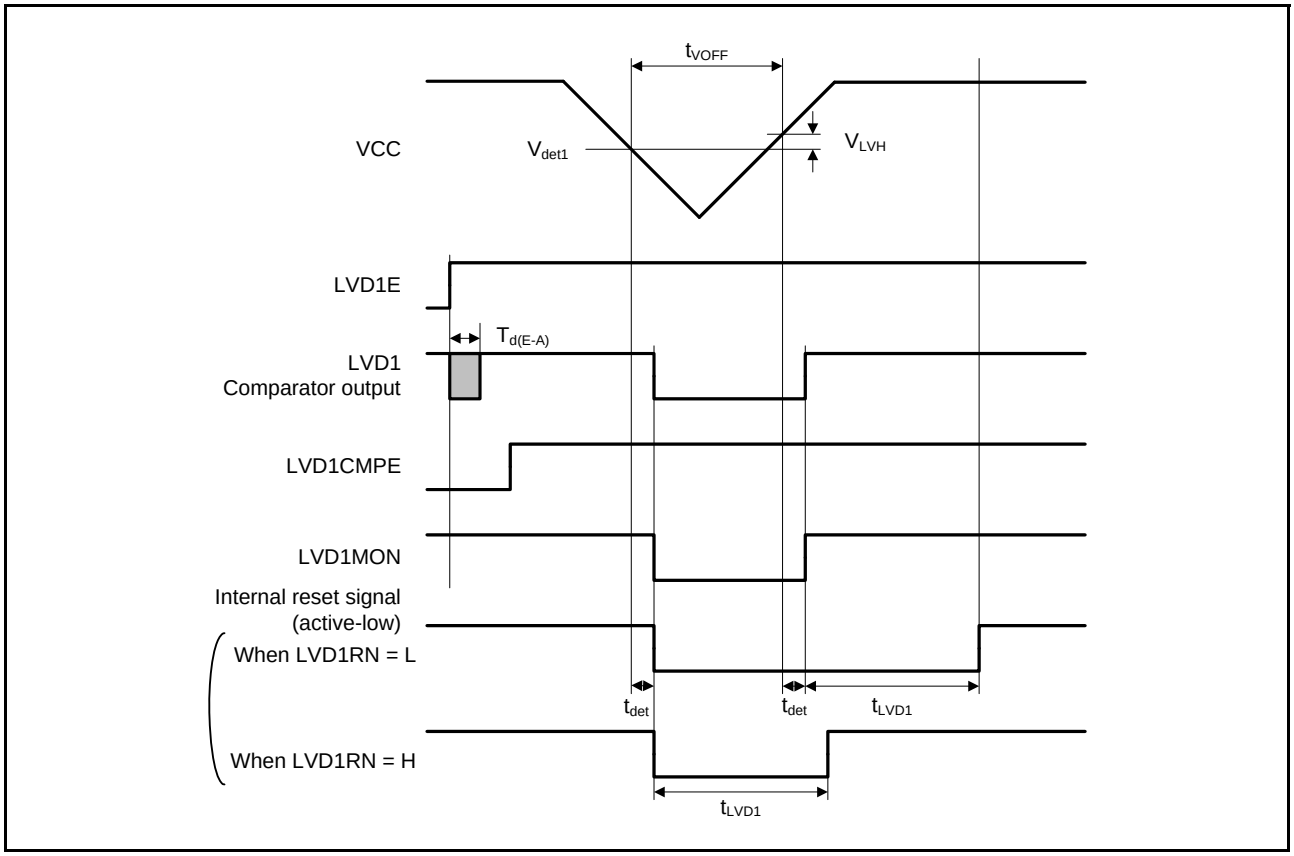


Figure 38.54 Voltage Detection Circuit Timing (V_{det1})

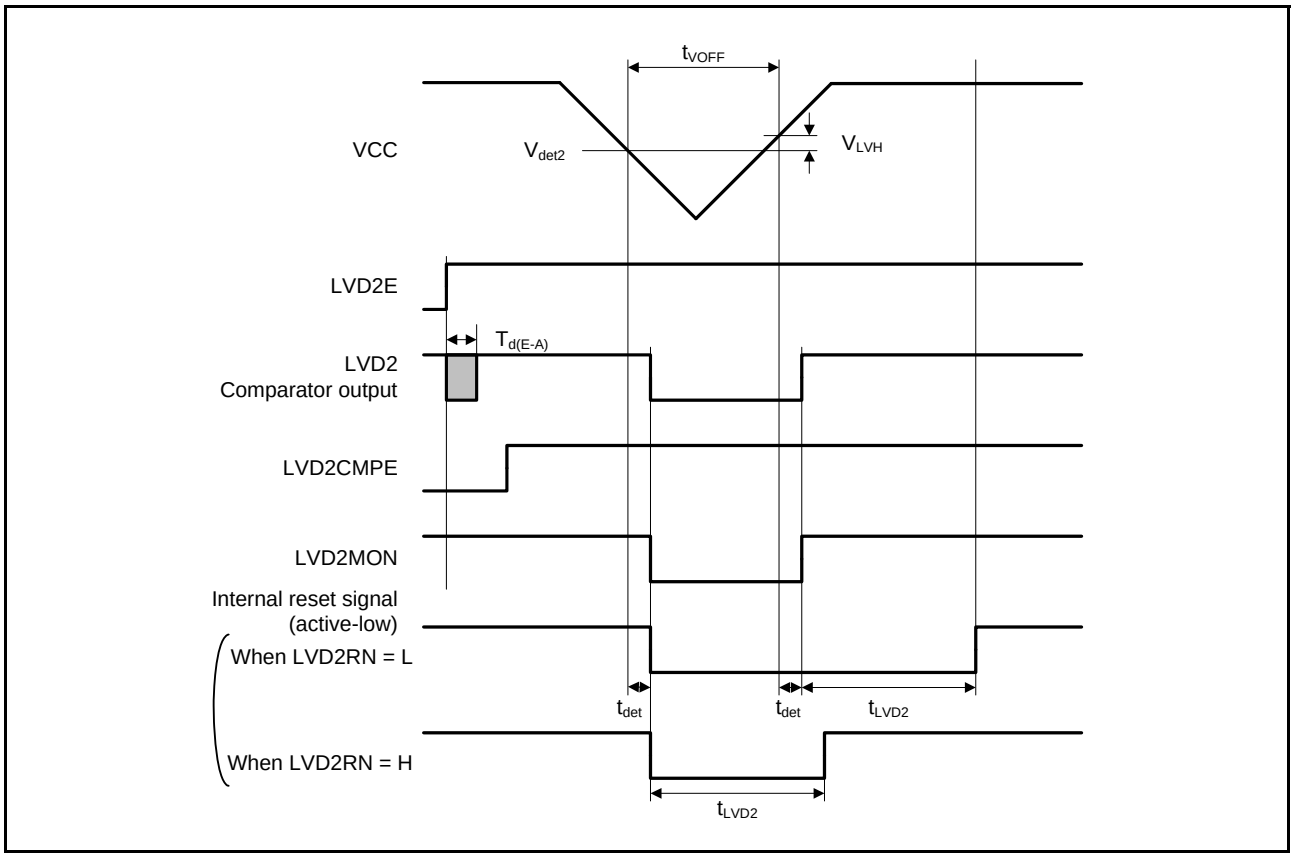


Figure 38.55 Voltage Detection Circuit Timing (V_{det2})

38.7 Oscillation Stop Detection Timing

Table 38.39 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = AVCC0 = 1.62 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 38.56

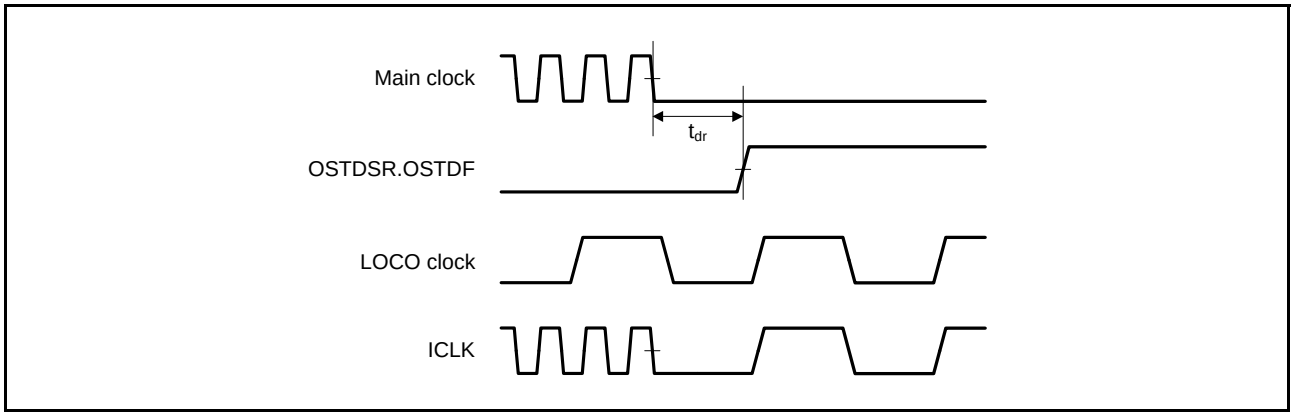


Figure 38.56 Oscillation Stop Detection Timing

38.8 ROM (Flash Memory for Code Storage) Characteristics

Table 38.40 ROM (Flash Memory for Code Storage) Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	10000	—	—	Times	
Data hold time	After 1000 times of N_{PEC}	t_{DRP}	30*2	—	—	Year	T_a = +85°C
	After 10000 times of N_{PEC}		1*2	—	—	Year	

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 = 10000), 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 38.41 ROM (Flash Memory for Code Storage) Characteristics (2)

Item	Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
Peripheral clock notification command wait time	t_{PCKA}	—	—	960	—	—	120	µs

Table 38.42 ROM (Flash Memory for Code Storage) Characteristics (3) medium-speed operating mode 1A

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{PEC} ≤ 100 times	2 bytes	t_{P2}	—	0.19	4.3	—	0.12	2.0	ms
	8 bytes	t_{P8}	—	0.19	4.4	—	0.12	2.0	
	128 bytes	t_{P128}	—	0.67	10.7	—	0.41	4.8	
Programming time when N_{PEC} > 100 times	2 bytes	t_{P2}	—	0.23	5.3	—	0.15	2.5	ms
	8 bytes	t_{P8}	—	0.23	5.4	—	0.15	2.5	
	128 bytes	t_{P128}	—	0.80	13.2	—	0.48	6.0	
Erasure time when N_{PEC} ≤ 100 times	2 Kbytes	t_{E2K}	—	13.0	92.8	—	10.5	29	ms
Erasure time when N_{PEC} > 100 times	2 Kbytes	t_{E2K}	—	15.9	176.9	—	12.8	60	ms
Suspend delay time during programming (in programming/erasure priority mode)		t_{SPD}	—	—	0.9	—	—	0.8	ms
First suspend delay time during programming (in suspend priority mode)		t_{SPSD1}	—	—	220	—	—	120	µs
Second suspend delay time during programming (in suspend priority mode)		t_{SPSD2}	—	—	0.9	—	—	0.8	ms
Suspend delay time during erasing (in programming/erasure priority mode)		t_{SED}	—	—	0.9	—	—	0.8	ms
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	220	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{SESD2}	—	—	0.9	—	—	0.8	ms
FCU reset time		t_{FCUR}	20 µs or longer and FCLK × 6 or greater	—	—	20 µs or longer and FCLK × 6 or greater	—	—	µs

Table 38.43 ROM (Flash Memory for Code Storage) Characteristics (4) medium-speed operating mode 1B

Conditions: VCC = AVCC0 = 1.62 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 4 MHz			FCLK = 32 MHz*1			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{PEC} ≤ 100 times	2 bytes	t_{P2}	—	0.25	5.0	—	0.21	2.8	ms
	8 bytes	t_{P8}	—	0.25	5.3	—	0.21	3.0	
	128 bytes	t_{P128}	—	0.92	14.0	—	0.65	8.3	
Programming time when N_{PEC} > 100 times	2 bytes	t_{P2}	—	0.31	6.2	—	0.26	3.5	ms
	8 bytes	t_{P8}	—	0.31	6.6	—	0.26	3.7	
	128 bytes	t_{P128}	—	1.09	17.5	—	0.77	10.0	
Erasure time when N_{PEC} ≤ 100 times	2 Kbytes	t_{E2K}	—	21.0	113.6	—	18.5	46	ms
Erasure time when N_{PEC} > 100 times	2 Kbytes	t_{E2K}	—	25.6	220.6	—	22.5	90 (1000 times ≥ N_{PEC} > 100 times), 98 (10000 times ≥ N_{PEC} > 1000 times)	ms
Suspend delay time during programming (in programming/erasure priority mode)		t_{SPD}	—	—	1.7	—	—	1.6	ms
First suspend delay time during programming (in suspend priority mode)		t_{SPSD1}	—	—	220	—	—	120	µs
Second suspend delay time during programming (in suspend priority mode)		t_{SPSD2}	—	—	1.7	—	—	1.6	ms
Suspend delay time during erasing (in programming/erasure priority mode)		t_{SED}	—	—	1.7	—	—	1.6	ms
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	220	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{SESD2}	—	—	1.7	—	—	1.6	ms
FCU reset time		t_{FCUR}	20 µs or longer and FCLK × 6 or greater	—	—	20 µs or longer and FCLK × 6 or greater	—	—	µs

Note 1. The operating frequency is 8 MHz (max.) when the voltage is in the range from 1.62 V to less than 2.7 V.

38.9 E2 DataFlash (Flash Memory for Data Storage) Characteristics

Table 38.44 E2 DataFlash Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100000	—	—	Times	
Data hold time	After 100000 times of N_{DPEC}	t_{DRP}	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 38.45 E2 DataFlash Characteristics (2)

Item	Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
Peripheral clock notification command wait time	t_{PCKA}	—	—	960	—	—	120	µs

Table 38.46 E2 DataFlash Characteristics (3) medium-speed operating mode 1A

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VREFH0 = AVCC0, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{DPEC} ≤ 100 times	2 bytes	t_{DP2}	—	0.19	4.4	—	0.13	2.0	ms
	8 bytes	t_{DP8}	—	0.24	5.1	—	0.13	2.2	
Programming time when N_{DPEC} > 100 times	2 bytes	t_{DP2}	—	0.25	6.4	—	0.17	3.0	ms
	8 bytes	t_{DP8}	—	0.32	7.5	—	0.18	3.2	
Erasure time when N_{DPEC} ≤ 100 times	128 bytes	t_{DE128}	—	3.3	27.1	—	2.5	8	ms
Erasure time when N_{DPEC} > 100 times	128 bytes	t_{DE128}	—	4.0	45.1	—	3.0	12	ms
Blank check time	2 bytes	t_{DBC2}	—	—	98	—	—	35	µs
	2 Kbytes	t_{DBC2K}	—	—	16	—	—	2.5	ms
Suspend delay time during programming (in programming/erasure priority mode)		t_{DSPD}	—	—	0.9	—	—	0.8	ms
First suspend delay time during programming (in suspend priority mode)		t_{DSPSD1}	—	—	220	—	—	120	µs
Second suspend delay time during programming (in suspend priority mode)		t_{DSPSD2}	—	—	0.9	—	—	0.8	ms
Suspend delay time during erasing (in programming/erasure priority mode)		t_{DSED}	—	—	0.9	—	—	0.8	ms
First suspend delay time during erasing (in suspend priority mode)		t_{DSESD1}	—	—	220	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{DSESD2}	—	—	0.9	—	—	0.8	ms

**Table 38.47 E2 DataFlash Characteristics (4)
medium-speed operating mode 1B**

Conditions: VCC = AVCC0 = 1.62 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: T_a = –40 to +105°C

Item		Symbol	FCLK = 4 MHz			FCLK = 32 MHz*1			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time when N_{DPEC} ≤ 100 times	2 bytes	t_{DP2}	—	0.28	5.1	—	0.20	2.8	ms
	8 bytes	t_{DP8}	—	0.32	6.0	—	0.22	3.2	
Programming time when N_{DPEC} > 100 times	2 bytes	t_{DP2}	—	0.36	7.6	—	0.25	4.2	ms
	8 bytes	t_{DP8}	—	0.40	8.8	—	0.28	4.5	
Erasure time when N_{DPEC} ≤ 100 times	128 bytes	t_{DE128}	—	4.8	32.3	—	4.1	12	ms
Erasure time when N_{DPEC} > 100 times	128 bytes	t_{DE128}	—	5.8	51.4	—	4.9	17	ms
Blank check time	2 bytes	t_{DBC2}	—	—	110	—	—	40	µs
	2 Kbytes	t_{DBC2K}	—	—	16.3	—	—	2.6	ms
Suspend delay time during programming (in programming/erasure priority mode)		t_{DSPD}	—	—	1.7	—	—	1.6	ms
First suspend delay time during programming (in suspend priority mode)		t_{DSPSD1}	—	—	220	—	—	120	µs
Second suspend delay time during programming (in suspend priority mode)		t_{DSPSD2}	—	—	1.7	—	—	1.6	ms
Suspend delay time during erasing (in programming/erasure priority mode)		t_{DSED}	—	—	1.7	—	—	1.6	ms
First suspend delay time during erasing (in suspend priority mode)		t_{DSESD1}	—	—	220	—	—	120	µs
Second suspend delay time during erasing (in suspend priority mode)		t_{DSESD2}	—	—	1.7	—	—	1.6	ms

Note 1. The operating frequency is 8 MHz (max.) when the voltage is in the range from 1.62 V to less than 2.7 V.

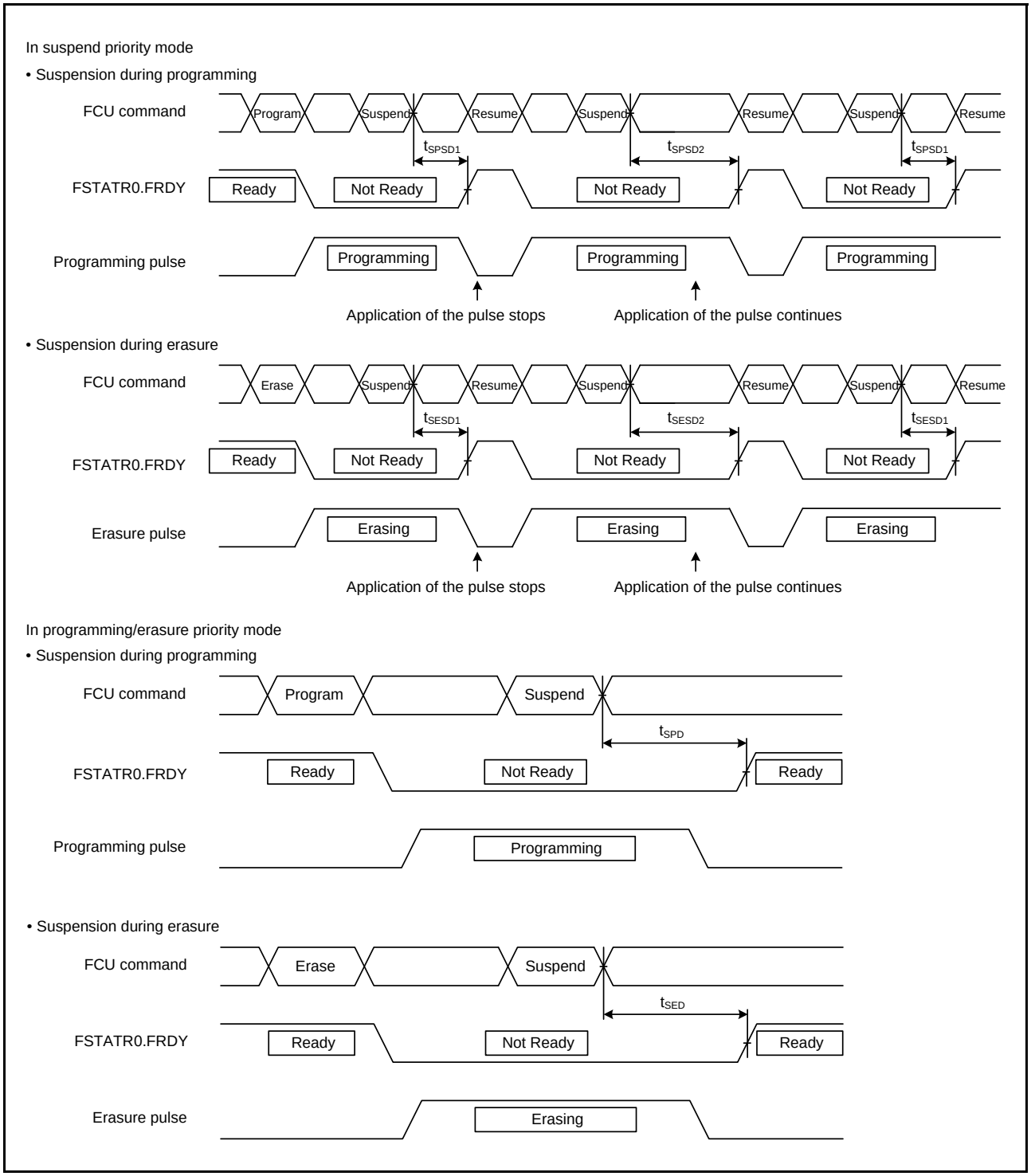


Figure 38.57 Flash Memory Program/Erase Suspend Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State

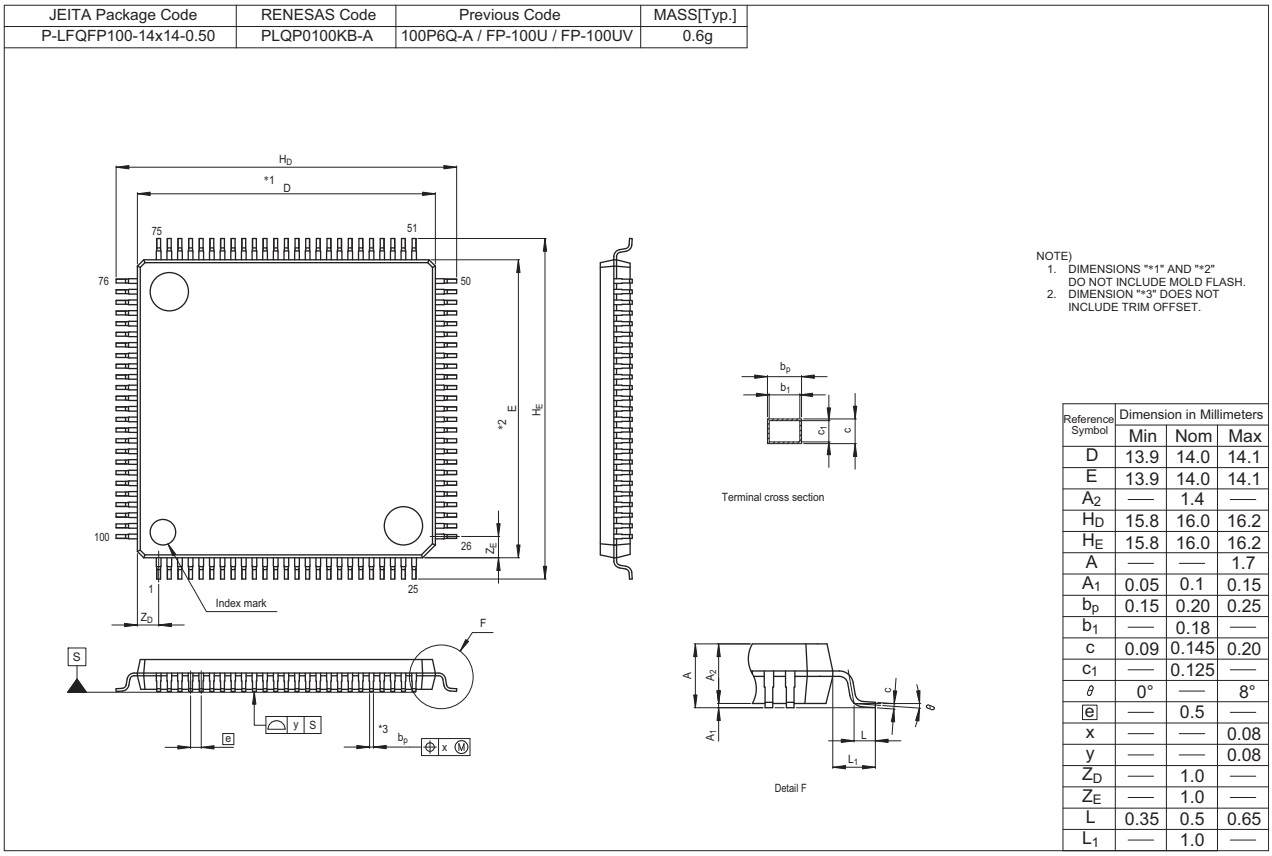
Port Name Pin Name	Reset	Software Standby Mode
P03, P05, P07	Hi-Z	Keep-O
P12 to P17 (IRQ2 to IRQ7)	Hi-Z	Keep-O*1
P20 to P27	Hi-Z	Keep-O
P30 to P34 (IRQ0 to IRQ4)	Hi-Z	Keep-O*1
P35 (NMI)	Hi-Z	Keep-O*1
P36, P37	Hi-Z	Keep-O
P40 to P47	Hi-Z	Keep-O
P50 to P55	Hi-Z	Keep-O
PA0 to PA2	Hi-Z	Keep-O
PA3, PA4 (IRQ6, IRQ5)	Hi-Z	Keep-O*1
PA5 to PA7	Hi-Z	Keep-O
PB0	Hi-Z	Keep-O
PB1 (IRQ4)	Hi-Z	Keep-O*1
PB2 to PB7	Hi-Z	Keep-O
PC0 to PC7	Hi-Z	Keep-O
PD0 to PD7 (IRQ0 to IRQ7)	Hi-Z	Keep-O*1
PE0, PE1	Hi-Z	Keep-O
PE2 (IRQ7)	Hi-Z	Keep-O*1
PE3, PE4	Hi-Z	Keep-O
PE5 to PE7 (IRQ5 to IRQ7)	Hi-Z	Keep-O*1
PH0	Hi-Z	Keep-O
PH1, PH2 (IRQ0, IRQ1)	Hi-Z	Keep-O*1
PH3	Hi-Z	Keep-O
PJ1, PJ3	Hi-Z	Keep-O

Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Hi-Z: High-impedance
Note 1. Input is enabled if the pin is specified as the software standby canceling source while it is used as an external interrupt pin.

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.

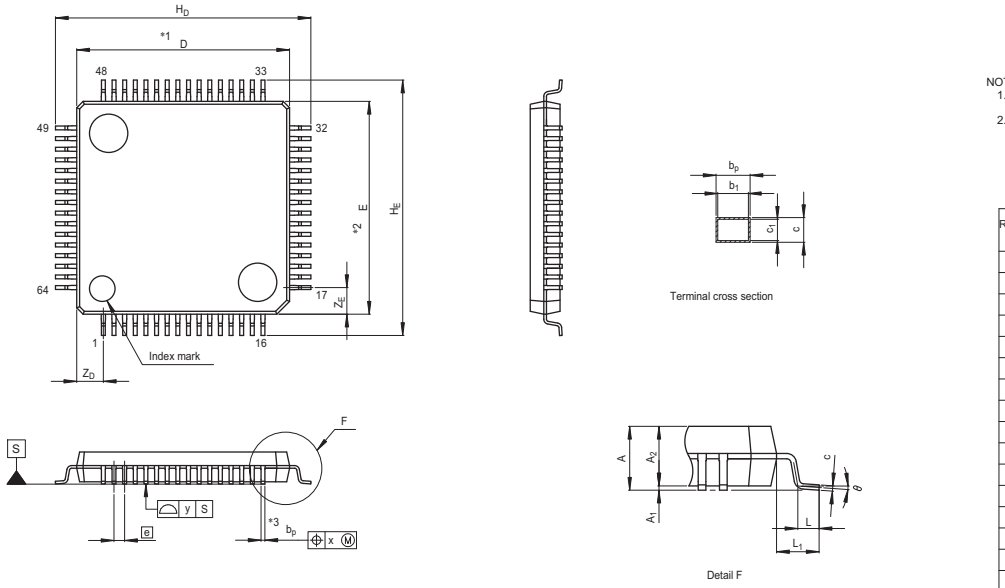
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure A 100-Pin LQFP (PLQP0100KB-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP64-10x10-0.50	PLQP0064KB-A	64P6Q-A / FP-64K / FP-64KV	0.3g

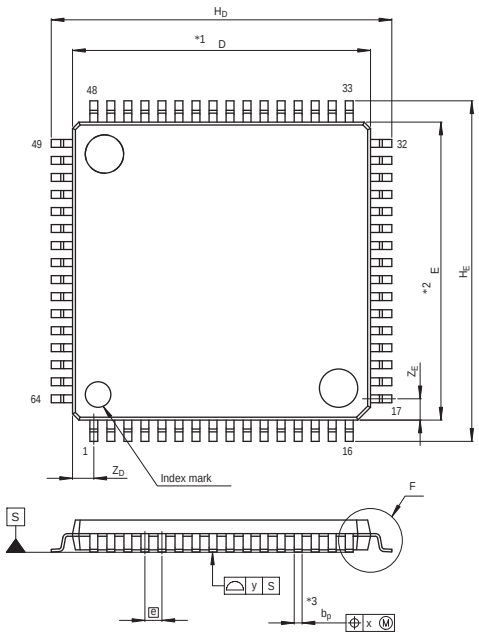


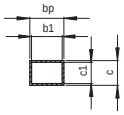
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

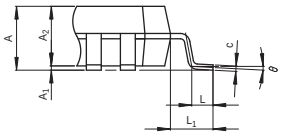
Figure B 64-Pin LQFP (PLQP0064KB-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-14x14-0.80	PLQP0064GA-A	64P6U-A/ —	0.7g





Terminal cross section



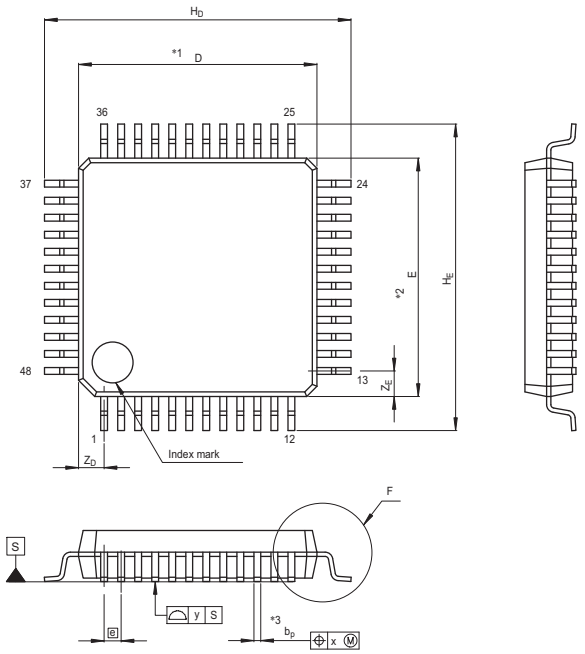
Detail F

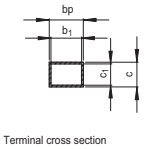
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.3	0.5	0.7
L_1	—	1.0	—

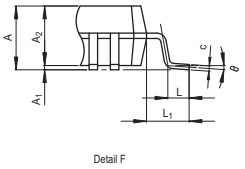
Figure C 64-Pin LQFP (PLQP0064GA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-A	48P6Q-A	0.2g





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	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
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	—	0.75	—
Z_E	—	0.75	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure D 48-Pin LQFP (PLQP0048KB-A)

REVISION HISTORY	RX220 Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.51	May 24, 2012	—	First edition issued
1.00	Dec 21, 2012	All	FINEC pin deleted, RTCb → RTCc, SCIc → SCIe, SCId → SCIf IrDA Interface added Terms, changed Reset generated by the pin → RES# pin reset
		Feature	
		35	IrDA, added Low-power design and architecture, Real-time clock, Up to seven communications channels, Operating temp. range, changed
		1. Overview	
		37, 38	Table 1.1 Outline of Specifications: General I/O ports, Event link controller (ELC), Realtime clock (RTCc), Serial communications interfaces (SCIe, SCIf), IrDA, Power supply voltage/ Operating frequency, Supply current, Operating temperature, changed
		39	Table 1.2 Comparison of Functions for Different Packages, changed
		40	Table 1.3 List of Products, changed Note 1, added
		41	Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type, changed
		42	Figure 1.2 Block Diagram, changed
		43, 44	Table 1.4 Pin Functions: Power supply, On-chip emulator, Serial communications interface (SCIe), changed
		47	Figure 1.4 Pin Assignments of the 64-Pin LQFP, Figure 1.5 Pin Assignments of the 48-Pin LQFP, changed
		48, 49	Table 1.5 List of Pins and Pin Functions (100-Pin LQFP), chaned
		51	Table 1.6 List of Pins and Pin Functions (64-Pin LQFP), chaned
		53	Table 1.7 List of Pins and Pin Functions (48-Pin LQFP), chaned
		2. CPU	
		63	2.4 Data Types: description, chanegd
		—	2.4.1 Integer, deleted
		—	2.4.2 Bits, deleted
		—	2.4.3 Strings, deleted
		3. Operating Modes	
		85	3.2.3 System Control Register 1 (SYSCR1): Bit description, changed
		87	Figure 3.1 Mode-Setting Pin Levels and Operating Modes, changed
		5. I/O Registers	
		97 to 111	Table 5.1 List of I/O Registers, changed Notes 1 and 2, added
		8. Voltage Detection Circuit (LVDAa)	
		141	Table 8.3 Procedures for Setting up Monitoring against Vdet1, changed
		141	Table 8.4 Procedures for Setting up Monitoring against Vdet1, changed
		9. Clock Generation Circuit	
		155	9.2.3 Main Clock Oscillator Control Register (MOSCCR): Bit description, changed
		163	9.2.11 Main Clock Oscillator Forced Oscillation Control Register (MOFCR): Description, changed
		168	9.4.2 Handling of Pins when Sub-Clock is not Used: Description, changed
		174, 175	9.7.5 Notes on Sub-Clock, changed
		11. Low Power Consumption	
		194	11.2.4 Module Stop Control Register C (MSTPCRC), changed
		196	Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption, changed
		197	• Middle-Speed Operating Mode 1A and • Middle-Speed Operating Mode 1B: Description, changed
		197	Figure 11.2 Relationship between the Operating Voltages and Operating Frequencies in Middle-Speed Operating Modes 1A and 1B, changed
		198	• Low-Speed Operating Mode 1: Description, changed
		200	11.2.6 Sleep Mode Return Clock Source Switching Register (RSTCKCR): Description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Dec 21, 2012	203	11.2.9 HOCO Wait Control Register 2 (HOCOWTCR2), changed
		204	11.2.10 Flash HOCO Software Standby Control Register (FHSSBYCR): Description, changed
		206	11.5.1 Setting Operating Power Consumption Control Mode, changed
		209	11.6.2.1 Transition to All-Module Clock Stop Mode: Note 5, added
		14. Interrupt Controller (ICUb)	
		247	14.3.1 Interrupt Vector Table: Description, changed
		248	Table 14.3 Interrupt Vector Table: Name, changed
		15. Buses	
		—	15.4 Limitations, deleted
		17. Data Transfer Controller (DTCa)	
		319	Figure 17.1 Block Diagram of DTC, changed
		332	Figure 17.4 Operation Flowchart of the DTC, changed
		347	17.8 Event Link Function, changed
		18. Event Link Controller (ELC)	
		351	Table 18.1 ELC Specifications: Note 1, changed
		351	Figure 18.1 Block Diagram of Event Link Controller, changed
		353	18.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 10, 12, 15, 18, 20, 22, 24, 25): Description, changed
		353	Table 18.2 Correspondence between ELSRn Registers and Peripheral Functions: Note 1, added
		354	Table 18.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers: 01010110 (56h), changed
		355	Table 18.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers: 01101010 (6Ah), added
		359	Table 18.4 Registers Related to Port Groups and Corresponding Port Numbers: Note 1, added
		367	(7) Restrictions on Writing to PODR or PDBF by CPU, changed
		369	18.3.6 Procedure for Linking Events: 7., changed
		370	18.4.1 Setting the ELSRn Register, added
		19. I/O Ports	
		371	19.1 Overview, chagned
		381	19.3.8 Drive Capacity Control Register (DSCR): Description, changed
		382	19.3.9 Port Switching Register A (PSRA), added
		383	19.3.10 Port Switching Register B (PSRB), added
		384	Table 19.3 Treatment of Unused Pins, changed Notes 1 and 2, added
		20. Multi-Function Pin Controller (MPC)	
		387	Table 20.1 Allocation of Pin Functions to Multiple Pins: Multi-function timer unit 2 MTU4, changed
		389	Table 20.1 Allocation of Pin Functions to Multiple Pins: Serial communications interface, changed
		393 to 411	20.2.2 P07 Pin Function Control Register (P07PFS) to 20.2.14 PJn Pin Function Control Registers (PJnPFS) (n = 1, 3): Register names, chagned
		401	Table 20.12 Register Settings for Input/Output Pin Function in 100-Pin: 1010b, changed
		402	Table 20.13 Register Settings for Input/Output Pin Function in 64-Pin: 1010b, changed
		402	Table 20.14 Register Settings for Input/Output Pin Function in 48-Pin: 1010b, changed
		405	Table 20.18 Register Settings for Input/Output Pin Function in 100-Pin: 1010b, changed
		405	Table 20.19 Register Settings for Input/Output Pin Function in 64-Pin: 1010b, changed
		412	20.3.1 Procedure for Specifying Input/Output Pin Function, changed
		21. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		418	Figure 21.1 Block Diagram of MTU, changed
		440	• TIER (MTU5): Address, changed
		501	Figure 21.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1's OLSN = 1 and OLSP = 1), chagned
		507	(b) Register Operation: Description, changed
		515	(j) Method for Generating PWM Output in Complementary PWM Mode: Description, changed
		515	(k) 0% and 100% Duty Cycle Output in Complementary PWM Mode, page changed
		539	Figure 21.79 Delay in Dead Time in Complementary PWM Mode Operation, changed
		540	Figure 21.81 Example of Motor Control Circuit Configuration, changed
		541	Figure 21.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation, changed
		545	(4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or MTU0.TGRB, changed

Rev.	Date	Description	
		Page	Summary
1.00	Dec 21, 2012	559	21.6.9 Contention between TGR Read Operation and Input Capture: Description, changed
		568	21.6.24 Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers, added
		574 to 599	(1) Operation when Error Occurs in Normal Mode and Operation is Restarted in Normal Mode to (29) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode: Description changed
			Figure 21.126 Error Occurrence in Normal Mode, Recovery in Normal Mode to Figure 21.154 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode, changed
		601	(3) Counter restart operation: Description, changed
		—	Figure 21.113 Contention between TGR Read Operation and Input Capture (MTU5), deleted
		—	Figure 21.124 Example of Synchronous Clearing (when Condition 2 Applies), deleted
		25. Realtime Clock (RTCc)	
		656	25.1 Overview, Table 25.1 Specifications of RTC, Note 1, changed
		657	Figure 25.1 Block Diagram of RTC, changed
		6547	Table 25.2 Pin Configuration of RTC: RTCOUT, changed
		658 to 679	25.2.1 64-Hz Counter (R64CNT) to 25.2.20 Time Error Adjustment Register (RADJ), changed
		680	Figure 25.2 Outline of Initial Settings after Power-On, changed
		681	Figure 25.3 Clock and the count mode Setting Procedure, changed
		682	Figure 25.4 Setting the Time, changed
		682	25.3.4 30-Second Adjustment: Description, changed
		684	25.3.6 Alarm Function: Description, changed
		685, 686	25.3.8.1 Automatic Adjustment, changed
		689	(3) Carry interrupt (CUP): Description, changed
		689	Figure 25.10 Carry Interrupt (CUP) Timing Chart, changed
		690	25.5.1 Register Writing during Counting: Description, changed
		690	25.5.2 Use of Periodic Interrupts and 25.5.3 RTCOUT (1-Hz/64-Hz) Clock Output, changed
		691	25.5.5 Points for Caution when Writing to and Reading from Registers, changed
		691	25.5.6 Changing Count Mode, added
		—	25.5.6 Initialization Procedure when the Realtime Clock is not to be Used, deleted
		26. Independent Watchdog Timer (IWDTa)	
		693	Table 26.1 Specifications of IWDT: Note 1, changed
		708	26.3.3 Refresh Operation, changed
		709	26.3.7 Reading the Down-Counter Value, changed
		—	Figure 26.7 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b), deleted
		—	Figure 26.9 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 11b), deleted
		27. Serial Communications Interface (SCIe, SCIf)	
		711	Table 27.1 Specifications of SCIe: Asynchronous mode, changed
		712	Table 27.2 Specifications of SCIf: Asynchronous mode, changed
		724	27.2.6 Serial Control Register (SCR): Note 2, changed
		729	27.2.7 Serial Status Register (SSR), changed
		738	Table 27.15 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Bus Mode): Note 1, changed
		741, 742	27.2.10 Serial Extended Mode Register (SEMR), changed
		742	Figure 27.4 Example of Average Transfer Rate Setting when TMR Clock is Input, changed
		753	27.2.20 Control Register 2 (CR2): Description, changed
		764	27.3 Operation in Asynchronous Mode, changed
		770	Figure 27.10 Example of Serial Transmission Flowchart in Asynchronous Mode, changed
		774	Figure 27.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		776	Figure 27.16 Example of Multi-Processor Serial Transmission Flowchart, changed
		779	Figure 27.19 Example of Multi-Processor Serial Reception Flowchart (2), changed
		787	Figure 27.26 Example of Serial Reception Flowchart (Clock Synchronous Mode), changed
		793	27.6.5 Initialization of the SCI (Smart Card Interface Mode): 6., changed
		826	Figure 27.63 Example of Operations with the Digital Filter, changed
		837	27.13.2 Break Detection and Processing, changed
		837	27.13.6 Restrictions on Clock Synchronous Transmission, changed
		838	27.13.9 SCI Operations during Low Power Consumption State: (1) Transmission, changed

Rev.	Date	Description	
		Page	Summary
1.00	Dec 21, 2012	839	Figure 27.69 Example of Flowchart for Transition to Software Standby Mode during Transmission, changed
		842	27.13.11 Limitation on Using Simple SPI Bus Mode, changed
		28. IrDA Interface	
		844 to 848	Added
		29. I2C Bus Interface (RIIC)	
		852	Table 29.3 Register Allocation for 16-Bit Access, added
		881	29.2.18 Timeout internal counter (TMOCNT), changed
		883	Figure 29.5 Example of RIIC Initialization Flowchart, changed
		885	Figure 29.6 Example of Master Transmission Flowchart, changed
		890	Figure 29.10 Example of Master Reception Flowchart (7-Bit Address Format), changed
		894	Figure 29.14 Example of Slave Transmission Flowchart, changed
		897	Figure 29.17 Example of Slave Reception Flowchart, changed
		30. Serial Peripheral Interface (RSPI)	
		928	30.1 Overview, changed
		937	30.2.4 RSPI Status Register (SPSR): Bit description, changed
		1004	Table 30.13 Interrupt Sources of RSPI: RSPI idle, changed
		1006	30.4.5 Transmit End Event Output, changed
		32. 12-Bit A/D Converter (S12ADb)	
		1054	32.7.7 Allowable Impedance of Signal Source, changed
		1054	Table 32.9 Specifications of Analog Input Pins: Max., changed Notes 1 and 2, changed
		34. Data Operation Circuit (DOC)	
		1073	Table 34.1 Specifications of Data Operation Circuit, changed
		1073	Figure 34.1 Block Diagram of Data Operation Circuit, changed
		1079	34.5 Event Link Output, 34.5.1 Interrupt Handling and Event Linking, added
		36. ROM (Flash Memory for Code Storage)	
		1093	36.2.8 Flash P/E Mode Entry Register (FENTRYR): Bit description, changed
		1114	Figure 36.11 Using the Peripheral Clock Notification Command, changed
		1145	(13) Boot Program Status Inquiry, changed
		37. E2 DataFlash Memory (Flash Memory for Data Storage)	
		1158	Table 37.1 Specifications of E2 DataFlash Memory: Protection, changed
		38. Electrical Characteristics	
		1184 to 1236	Added
1.10	Dec 20, 2013	All	PLQP0064GA-A 14×14 mm, 0.8-mm pitch added
		Features	
		35	■ Operating temp. range changed
		1. Overview	
		38	Table 1.1 Outline of Specifications changed, Note 1 added
		40	Table 1.3 List of Products changed, Note added
		41	Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type changed
		3. Operating Modes	
		85	3.2.3 System Control Register 1 (SYSCR1) Note added
		6. Resets	
		113	Table 6.2 Targets Initialized by Each Reset Source Note1 added
		116	6.2.2 Reset Status Register 1 (RSTSR1) changed
		117	6.2.4 Software Reset Register (SWRR) Note added
		8. Voltage Detection Circuit (LVDAa)	
		133 to 137	8.2.1 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 1 (LVD1CR1) to 8.2.5 Voltage Monitoring Circuit/Comparator A Control Register (LVCMPCR) Note added
		138	8.2.6 Voltage Detection Level Select Register (LVDLVLR) changed, Note added
		139, 140	8.2.7 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 0 (LVD1CR0) and 8.2.8 Voltage Monitoring 2 Circuit/Comparator A2 Control Register 0 (LVD2CR0) Note added
		9. Clock Generation Circuit	
		152, 153	9.2.1 System Clock Control Register (SCKCR) changed, Note added
		154	9.2.2 System Clock Control Register 3 (SCKCR3) changed, Note added
		155	9.2.3 Main Clock Oscillator Control Register (MOSCCR) changed, Note added

Rev.	Date	Description	
		Page	Summary
1.10	Dec 20, 2013	156 to 160	9.2.4 Sub-Clock Oscillator Control Register (SOSCCR) to 9.2.8 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2) Note added
		161	9.2.9 Oscillation Stop Detection Control Register (OSTDCR) changed, Note added
		162	9.2.10 Oscillation Stop Detection Status Register (OSTDSR) changed, Note added
		163	9.2.11 Main Clock Oscillator Forced Oscillation Control Register (MOFCR) Note added
		164	9.2.12 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR) changed, Note added
		170	Figure 9.8 Flow of Recovery from Detection of Oscillator Stop changed
		11. Low Power Consumption	
		190 to 194	11.2.1 Standby Control Register (SBYCR) to 11.2.4 Module Stop Control Register C (MSTPCRC) Note added
		195	11.2.5 Operating Power Control Register (OPCCR) changed, Note added
		200 to 204	11.2.6 Sleep Mode Return Clock Source Switching Register (RSTCKCR) to 11.2.10 Flash HOCO Software Standby Control Register (FHSSBYCR) Note added
		14. Interrupt Controller (ICUb)	
		260	14.4.5 Multiple Interrupts added, 14.4.6 Fast Interrupt changed
		16. DMA Controller (DMACA)	
		309	Figure 16.12 Register Setting Procedure changed
		17. Data Transfer Controller (DTCa)	
		329	17.3 Sources of Activation changed
		339	Figure 17.8 Chain Transfer Operation changed
		349	17.10.1 Transfer Data Start Address: Title changed
		18. Event Link Controller (ELC)	
		351	Table 18.1 ELC Specifications changed
		19. I/O Ports	
		384	Table 19.3 Treatment of Unused Pins changed
		20. Multi-Function Pin Controller (MPC)	
		385	Table 20.1 Allocation of Pin Functions to Multiple Pins changed
		21. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		451	21.2.17 Timer Output Master Enable Registers (TOER), Note 1 changed
		510	(g) PWM Cycle Setting changed
		25. Realtime Clock (RTCc)	
		673	25.2.17 RTC Control Register 1 (RCR1) changed
		674 to 677	25.2.18 RTC Control Register 2 (RCR2) changed
		26. Independent Watchdog Timer (IWDTa)	
		708	26.3.4 Status Flags changed
		27. Serial Communications Interface (SCIe, SCIf)	
		782	Figure 27.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode) changed
		799	(a) At transition from smart card interface mode to software standby mode changed
		843	27.13.14 Note on Transmit Enable Bit (TE bit) added
		29. I²C Bus Interface (RIIC)	
		849 to 928	ICRXI → RXI, ICTEI → TEI, ICTXI → TXI, ICEEI → EEI changed
		852, 853	29.2.1 I²C Bus Control Register 1 (ICCR1) changed
		854	29.2.2 I²C Bus Control Register 2 (ICCR2) changed
		885	Figure 29.6 Example of Master Transmission Flowchart changed
		890	Figure 29.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes) added
		893	Figure 29.14 Master Receive Operation Timing (3) (when RDRFS = 0) changed
		918	29.11.1 Timeout Function changed
		30. Serial Peripheral Interface (RSPI)	
		937	30.2.4 RSPI Status Register (SPSR) changed
		940	(a) Writing changed
		989	Figure 30.34 Example of Initialization Flowchart in Master Mode (SPI Operation) changed
		1000	Figure 30.45 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation) changed
		1006	(1) Mode Fault changed
		32. 12-Bit A/D Converter (S12ADb)	
		1040	32.3.2.2 Channel Selection and Self-Diagnosis changed
		1044	32.3.3.2 Channel Selection and Self-Diagnosis changed

Rev.	Date	Description	
		Page	Summary
1.10	Dec 20, 2013	1048	Figure 32.11 Scan Conversion Timing (Activated by Software, or Triggers from the MTU or ELC), Figure 32.12 Scan Conversion Timing (Activated by ADTRG0#) changed
		33. Comparator A	
		1060 to 1066	33.2.1 Voltage Monitoring 1 Circuit/Comparator A1 Control Register 1 (LVD1CR1) to 33.2.7 Voltage Monitoring 2 Circuit/Comparator A2 Control Register 0 (LVD2CR0) Note added
		34. Data Operation Circuit (DOC)	
		1077	34.3.1 Data Comparison Mode Note1 added
		36. ROM (Flash Memory for Code Storage)	
		1116	(4) Programming changed: Title changed
		1127	Figure 36.20 Suspension during Erasure (Programming/Erasure Priority Mode): Title changed
		37. E2 DataFlash Memory (Flash Memory for Data Storage)	
		1173	Table 37.4 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory) changed
		1175	(2) Programming changed
		38. Electrical Characteristics	
		1187	Table 38.4 DC Characteristics (3) changed
		1193	Table 38.8 DC Characteristics (7) added
		1194	Table 38.13 Permissible Output Currents (1) changed, Table 38.14 Permissible Output Currents (2) added
		1195	Table 38.15 Output Values of Voltage (1) changed, Table 38.16 Output Values of Voltage (2) added
		1211	Table 38.26 Timing of On-Chip Peripheral Modules (1) changed
		1223	Table 38.31 A/D Conversion Characteristics (1) changed
		1224	Table 38.34 A/D Conversion Characteristics (2) changed
		1229	Table 38.38 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (2) changed

RX220 Group User's Manual: Hardware

Publication Date: Rev.0.51 May 24, 2012
Rev.1.10 Dec 20, 2013

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

RX220 Group

Renesas Electronics Corporation

R01UH0292EJ0110