User's Manual

32

RX23T Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX200 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Jan 2016

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.

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. 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 RX23T Group. Before using any of the documents, please visit our website to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Datasheet	Overview of hardware and electrical characteristics	RX23T Group Datasheet	R01DS0248EJ
User’s Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX23T Group User’s Manual: Hardware	This User’s manual
User’s Manual: Software	Detailed descriptions of the CPU and instruction set	RX Family RXv2 Instruction Set Architecture User’s Manual: Software	R01US0071EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Examples of register initial setting	RX23T Group Initial Setting Examples	R01AN2551EJ
	Examples of applications and sample programs	—	—
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.	—	—

2. Description of Registers

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

X.X.X ... Register

Address(es): xxxx xxxxh

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

x: Undefined

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

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

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

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

3. List of Abbreviations and Acronyms

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

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Contents

RX23T Group

Renesas MCUs

R01UH0520EJ0110
Rev.1.10
Jan 13, 2016

40-MHz 32-bit RX MCUs, built-in FPU, 65.6 DMIPS,
12-bit ADC (equipped with three S/H circuits, double data registers, and comparator)
40MHz PWM (three-phase complementary output × 2ch)

Features

■ 32-bit RX CPU core
- Max. operating frequency: 40 MHz
  Capable of 65.6 DMIPS in operation at 40 MHz
- Enhanced DSP: 32-bit multiply-accumulate and 16-bit multiply-subtract instructions supported
- Built-in FPU: 32-bit single-precision floating point (compliant to IEEE754)
- Divider (fastest instruction execution takes two CPU clock cycles)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- On-chip debugging circuit
- Memory protection unit (MPU) supported

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

■ On-chip code flash memory, no wait states
- 128-/64-Kbyte capacities
- On-board or off-board user programming

■ On-chip SRAM, no wait states
- 12 Kbytes of SRAM

■ DMA
- DTC: Four transfer modes

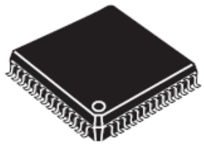
■ Reset and supply management
- Seven types of reset, including the power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions
- Main clock oscillator frequency: 1 to 20 MHz
- External clock input frequency: Up to 20 MHz
- PLL circuit input: 4 MHz to 12.5 MHz
- On-chip low-speed oscillator, on-chip high-speed oscillator, dedicated on-chip oscillator for the IWDT
- Clock frequency accuracy measurement circuit (CAC)

■ Independent watchdog timer
- 15-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, RAM test assistance functions using the DOC, etc.

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



PLQP0064KB-C 10 × 10 mm, 0.5 mm pitch
PLQP0052JA-B 10 × 10 mm, 0.65 mm pitch
PLQP0048KB-B 7 × 7 mm, 0.5 mm pitch

■ Up to 4 communications channels
- SCI with many useful functions (2 channels)
  Asynchronous mode, clock synchronous mode, smart card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- I^2C bus interface: Transfer at up to 400 kbps (one channel)
- RSPI capable of high speed connection (one channel)

■ Up to 12 extended-function timers
- 16-bit MTU3: 40MHz operation, input capture, output compare, three-phase complementary PWM output, CPU-efficient complementary PWM, phase counting mode (six channels)
- 8-bit TMRs (4 channels),
- 16-bit compare-match timers (4 channels)

■ 12-bit A/D converter: 10ch
- On-chip sample-and-hold circuit: 12bit × up to 3 channels
- Sampling time can be set for each channel
- Self-diagnostic function and analog input disconnection detection assistance function (compliant to IEC60730)
- ADC: three sample-and-hold circuits, double data registers, comparator (3 channels)

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

■ Up to 50 pins for general I/O ports
- 5-V tolerant, open drain, input pull-up

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

■ Applications
- General industrial and consumer equipment

1. Overview

1.1 Outline of Specifications

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

Table 1.1 is for products with the greatest number of functions, so the number of peripheral modules and channels will differ in accordance with the package type. 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: 40 MHz • 32-bit RX CPU (RX v2) • Minimum instruction execution time: One instruction per clock cycle • Address space: 4-Gbyte linear • Register set General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • Basic instructions: 75 Variable-length instruction format • Floating-point instructions: 11 • DSP instructions: 23 • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit • On-chip divider: 32-bit ÷ 32-bit → 32 bits • Barrel shifter: 32 bits • Memory protection unit (MPU)
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	ROM	• Capacity: 64 K/128 Kbytes • 32 MHz, no-wait memory access 32 to 40 MHz: wait states • Programming/erasing method: Serial programming (asynchronous serial communication), self-programming
	RAM	• Capacity: 12 Kbytes • 40 MHz, no-wait memory access
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	• Main clock oscillator, low-speed and high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • Oscillation stop detection: Available • Clock frequency accuracy measurement circuit (CAC): Available • 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): 40 MHz (at max.) MTU3c runs in synchronization with the PCLKA: 40 MHz (at max.) Peripheral modules other than MTU3c run in synchronization with the PCLKB: 40 MHz (at max.) ADCLK operated in S12ADE runs in synchronization with the PCLKD: 40 MHz (at max.) The flash peripheral circuit runs in synchronization with the FCLK: 32 MHz (at max.)
Resets		RES# pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, and software reset
Voltage detection	Voltage detection circuit (LVDAb)	• 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 2 levels Voltage detection circuit 1 is capable of selecting the detection voltage from 9 levels Voltage detection circuit 2 is capable of selecting the detection voltage from 4 levels
Low power consumption	Low power consumption functions	• Module stop function • Three low power consumption modes Sleep mode, deep sleep mode, and software standby mode
	Function for lower operating power consumption	• Operating power control modes High-speed operating mode and middle-speed operating mode

Table 1.1 Outline of Specifications (2/3)

Classification	Module/Function	Description
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 83 • External interrupts: 7 (NMI, IRQ0 to IRQ5 pins) • Non-maskable interrupts: 5 (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
DMA	Data transfer controller (DTCa)	• Transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Interrupts • Chain transfer function
I/O ports	General I/O ports	64-/52-/48-pin • I/O: 50/40/37 • Input: 1/1/1 • Pull-up resistors: 50/40/37 • Open-drain outputs: 42/32/29 • 5-V tolerance: 2/2/2
Multi-function pin controller (MPC)		Capable of selecting the input/output function from multiple pins
Timers	Multi-function timer pulse unit 3 (MTU3c)	• 6 units (16bis × 6 channels) • Provides up to 16 pulse-input/output lines and three pulse-input lines • Select from among fourteen counter-input clock signals for each channel (PCLK/1, PCLK/2, PCLK/4, PCLK/8, PCLK/16, PCLK/32, PCLK/64, PCLK/256, PCLK/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD, MTIOC1A) other than channel 1/3/4, for which only eleven signals are available, channel 2 for 12, channel 5 for 10 • 26 output compare/input capture registers • Counter clear operation (with compare match- or input capture-sourced simultaneous counter clear capability) • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous register input/output by synchronous counter operation • Buffer operation • Cascaded operation • 28 interrupt sources • Automatic transfer of register data • Pulse output modes: Toggle/PWM/complementary PWM/reset-synchronized PWM • Complementary PWM output mode 3-phase non-overlapping waveform output for inverter control Automatic dead time setting Adjustable PWM duty cycle: from 0 to 100% A/D conversion request delaying function Interrupt at crest/trough can be skipped Double buffer function • Reset-synchronized PWM mode Outputs three phases each for positive and negative PWM waveforms in user-specified duty cycle • Phase counting modes: 16-bit mode (channel 1 and 2)/32-bit mode (channel 1 and 2) • Dead time compensation counter function • A/D converter start trigger can be generated • A/D converter start triggers can be skipped • Signals from the input capture and external counter clock pins are input via a digital filter
	Port output enable 3 (POE3b)	Controls the high-impedance state of the MTU's waveform output pins
	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 • Count clock: Dedicated low-speed on-chip oscillator for the IWDT Frequency divided by 1, 16, 32, 64, 128, or 256
	8-bit timer (TMR)	• (8 bits × 2 channels) × 2 units • Seven internal clocks (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, and PCLK/8192) and an external clock can be selected • Pulse output and PWM output with any duty cycle are available • Two channels can be cascaded and used as a 16-bit timer • Generates A/D conversion start trigger • Generates baud rate clock for the SCI5

Table 1.1 Outline of Specifications (3/3)

Classification	Module/Function	Description
Communication functions	Serial communications interfaces (SCIg)	• 2 channels (channel 1 and 5: SCIg) • SCIg 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 for SCI5 Simple I^2C Simple SPI 9-bit transfer mode Bit rate modulation
	I^2C bus interface (RIICa)	• 1 channel • Communications formats: I^2C bus format/SMBus format • Master mode or slave mode selectable • Supports fast mode
	Serial peripheral interface (RSPIa)	• 1 channel • Transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPI clock (RSPCK) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) • Capable of handling serial transfer as a master or slave • Data formats • Choice of LSB-first or MSB-first transfer The number of bits in each transfer can be changed to 8, 9, 10, 11, 12, 13, 14, 15, 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 (S12ADE)		• 12 bits (10 channels × 1 unit) • 12-bit resolution • Minimum conversion time: 1.0 µs per channel when the ADCLK is operating at 40 MHz • Operating modes Scan mode (single scan mode, continuous scan mode, and group scan mode) Group A priority control (only for group scan mode) • Sampling variable Sampling time can be set up for each channel • Self-diagnostic function • Double trigger mode (A/D conversion data duplicated) • Detection of analog input disconnection • A/D conversion start conditions A software trigger, a trigger from a timer (MTU, TMR), or an external trigger signal
Comparator C (CMPC)		• 3 channels • Function to compare the reference voltage and the analog input voltage • Reference voltage: Select from among two voltages • Analog input voltage: Select from among four voltages
D/A converter (DA) for generating comparator C reference voltage		• 1 channel • 8-bit resolution • Output voltage: 0 to AVCC0 • Reference voltage generation circuit for comparator C
CRC calculator (CRC)		• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Data operation circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Power supply voltages/Operating frequencies		VCC = 2.7 to 5.5V: 40MHz
Supply current		15 mA at 40 MHz (typ.)
Operating temperature range		D version: –40 to +85°C, G version: –40 to +105°C
Packages		64-pin LFQFP 0.5mm pitch 52-pin LQFP 0.65mm pitch 48-pin LFQFP 0.5mm pitch
On-chip debugging system		E1 emulator (FINE interface)

Table 1.2 Comparison of Functions for Different Packages

Module/Functions		RX23T Group		
		48 Pins	52 Pins	64 Pins
Interrupts	External interrupts	NMI, IRQ0 to IRQ5		
DTC	Data transfer controller	Available		
Timers	Multi-function timer pulse unit 3*1	6 channels		
	Port output enable 3	POE0# to POE8#, POE10#		
	8-bit timer	2 channels × 2 units		
	Compare match timer	2 channels × 2 units		
	Independent watchdog timer	Available		
Communication functions	Serial communications interfaces (SCIg) [including simple IIC and simple SPI]	2 channels (SCI1, 5)		
	I^2C bus interface	1 channel		
	Serial peripheral interface	1 channel		
12-bit A/D converter (including high-precision channels)		10 channels (8 channels)		
CRC calculator		Available		
Packages		48-pin LFQFP	52-pin LQFP	64-pin LFQFP

Note 1. For multi-function timer pulse unit 3, the number of pins differs depending on the package. For details, see the "List of Pins and Pin Functions" table for each pin.

1.2 List of Products

Table 1.3 and Table 1.4 are 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: D Version (T_a = –40 to +85°C)

Group	Part No.	Package	ROM Capacity	RAM Capacity	Operating Frequency	Operating Temperature
RX23T	R5F523T5ADFL	PLQP0048KB-B	128 Kbytes	12 Kbytes	40MHz	-40 to + 85°C
	R5F523T5ADFD	PLQP0052JA-B				
	R5F523T5ADFM	PLQP0064KB-C				
	R5F523T3ADFL	PLQP0048KB-B	64 Kbytes			
	R5F523T3ADFD	PLQP0052JA-B				
	R5F523T3ADFM	PLQP0064KB-C				

Table 1.4 List of Products: G Version (T_a = –40 to +105°C)

Group	Part No.	Package	ROM Capacity	RAM Capacity	Operating Frequency	Operating Temperature
RX23T	R5F523T5AGFL	PLQP0048KB-B	128 Kbytes	12 Kbytes	40MHz	-40 to +105°C
	R5F523T5AGFD	PLQP0052JA-B				
	R5F523T5AGFM	PLQP0064KB-C				
	R5F523T3AGFL	PLQP0048KB-B	64 Kbytes			
	R5F523T3AGFD	PLQP0052JA-B				
	R5F523T3AGFM	PLQP0064KB-C				

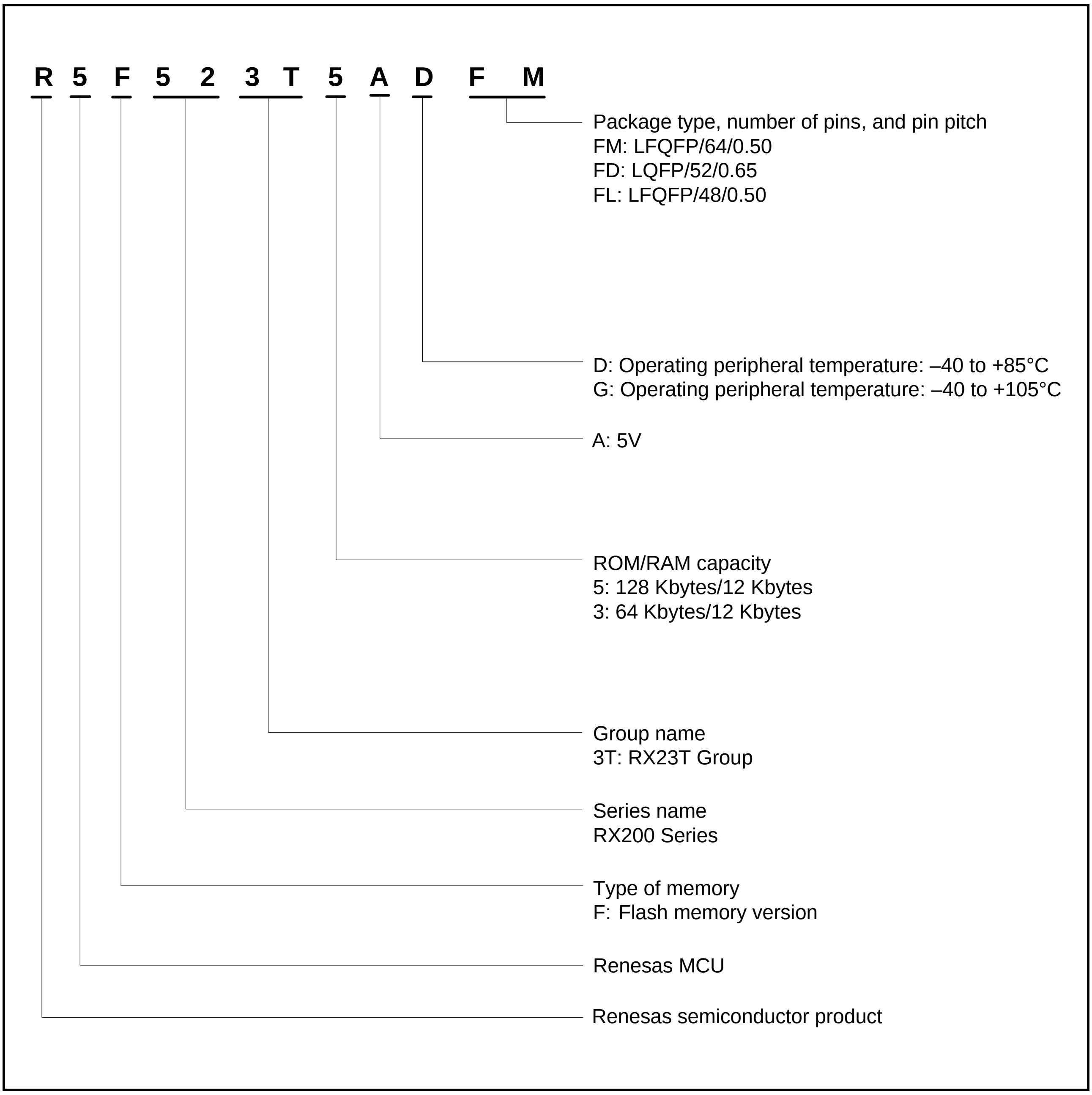
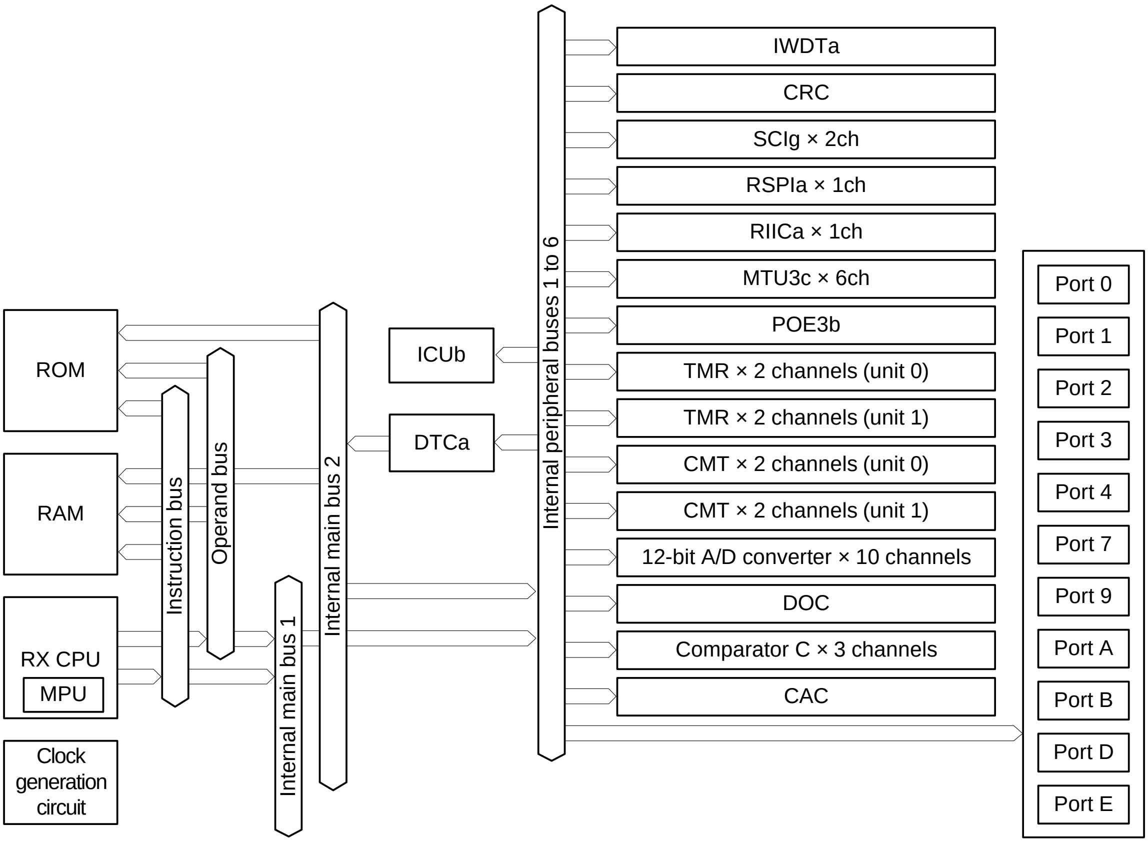


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICUb: Interrupt controller
DTCa: Data transfer controller
IWDTa: Independent watchdog timer
CRC: CRC (cyclic redundancy check) calculator
SCIg: Serial communications interface
RSPIa: Serial peripheral interface
RIICa I^2C bus interface

MTU3c Multi-function timer pulse unit 3
POE3b Port output enable 3
CMT Compare match timer
DOC Data operation circuit
CAC Clock frequency accuracy measurement circuit
MPU Memory protection unit
TMR 8-bit timer

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.5 lists the pin functions.

Table 1.5 Pin Functions (1/2)

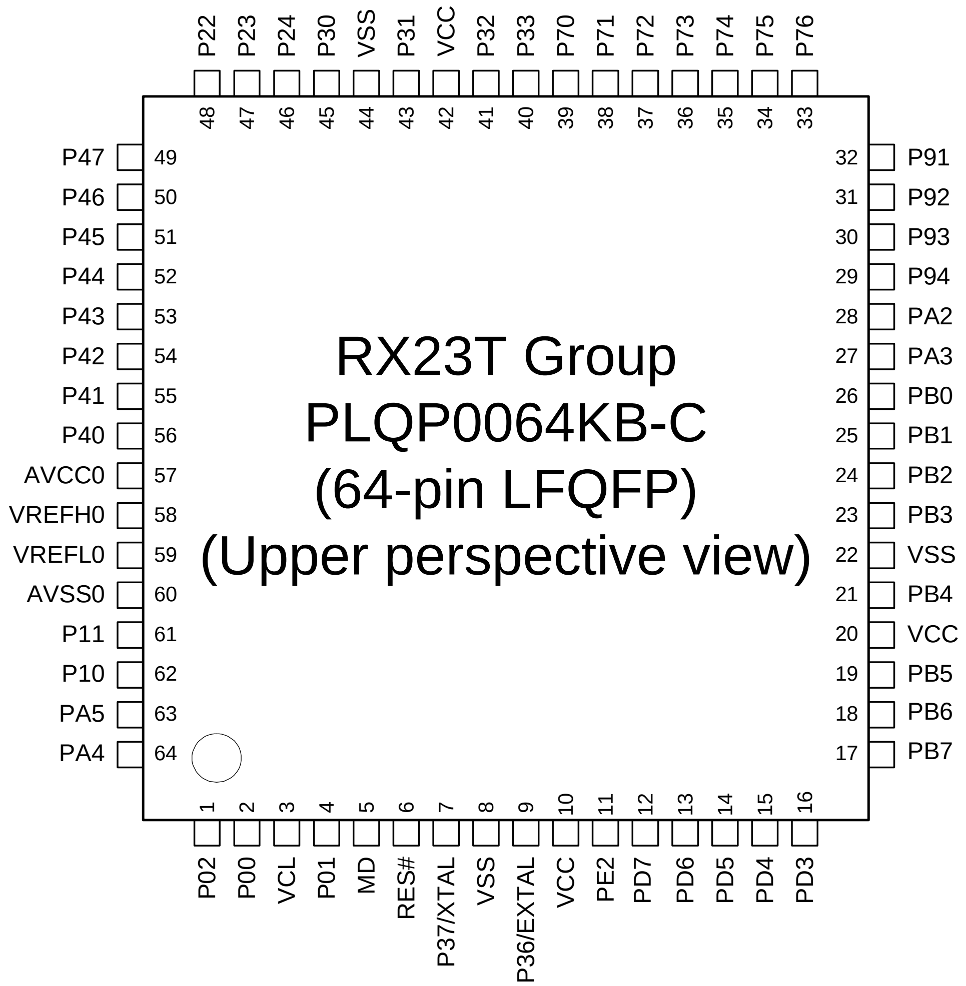
Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	—	Connect this pin to the VSS pin via the 4.7 µF smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for connecting a crystal. An external clock can be input through the EXTAL pin.
	EXTAL	Input	
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin must not be changed during operation.
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit.
On-chip emulator	FINED	I/O	FINE interface pin.
Interrupts	NMI	Input	Non-maskable interrupt request pin.
	IRQ0 to IRQ5	Input	Interrupt request pins.
Multi-function timer pulse unit 3	MTIOC0A, MTIOC0B MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins.
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins.
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins.
	MTIOC3A, MTIOC3B MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	MTIOC4A, MTIOC4B MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins.
	MTIC5U, MTIC5V, MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/external pulse input pins.
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.
	ADSM0	Output	A/D trigger output pin.
Port output enable 3	POE0#, POE8#, POE10#	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 the external clock to be input to the counter.
	TMRI0 to TMRI3	Input	Counter reset input pins.
Serial communications interface (SCIg)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5	I/O	Input/output pins for the clock.
	RXD1, RXD5	Input	Input pins for received data.
	TXD1, TXD5	Output	Output pins for transmitted data.
	CTS1#, CTS5#	Input	Input pins for controlling the start of transmission and reception.
	RTS1#, RTS5#	Output	Output pins for controlling the start of transmission and reception.
	• Simple I^2C mode		
	SSCL1, SSCL5	I/O	Input/output pins for the I^2C clock.
	SSDA1, SSDA5	I/O	Input/output pins for the I^2C data.

Table 1.5 Pin Functions (2/2)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIg)	• Simple SPI mode		
	SCK1, SCK5	I/O	Input/output pins for the clock.
	SMISO1, SMISO5	I/O	Input/output pins for slave transmit data.
	SMOSI1, SMOSI5	I/O	Input/output pins for master transmit data.
	SS1#, SS5#	Input	Chip-select input pins.
I^2C bus interface	SCL0	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open drain output.
Serial peripheral interface	RSPCKA	I/O	Input/output pin for the RSPI clock.
	MOSIA	I/O	Input/output pin for transmitting data from the RSPI master.
	MISOA	I/O	Input/output pin for transmitting data from the RSPI slave.
	SSLA0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1 to SSLA3	Output	Output pins to select the slave for the RSPI.
12-bit A/D converter	AN000 to AN007, AN016, AN017	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pin for the external trigger signals that start the A/D conversion.
	ADST0	Output	Output pin for A/D conversion status.
Comparator C	CMPC00, CMPC01, CMPC02	Input	Analog input pin for CMPC0
	CMPC10, CMPC11, CMPC12	Input	Analog input pin for CMPC1
	CMPC20, CMPC21, CMPC22	Input	Analog input pin for CMPC2
	COMP0 to COMP2	Output	Comparator detection result output pins.
	CVREFC0, CVREFC1	Input	Analog reference voltage supply pins for comparator C.
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter, comparator C, and the 8-bit D/A converter for generating comparator C reference voltage. Connect this pin to VCC when these modules are not used.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter, comparator C, and the 8-bit D/A converter for generating comparator C reference voltage. Connect this pin to VSS when these modules are not used.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter.
I/O ports	P00 to P02	I/O	3-bit input/output pins.
	P10, P11	I/O	2-bit input/output pins.
	P22 to P24	I/O	3-bit input/output pins.
	P30 to P33, P36, P37	I/O	6-bit input/output pins.
	P40 to P47	I/O	8-bit input/output pins.
	P70 to P76	I/O	7-bit input/output pins.
	P91 to P94	I/O	4-bit input/output pins.
	PA2 to PA5	I/O	4-bit input/output pins.
	PB0 to PB7	I/O	8-bit input/output pins.
	PD3 to PD7	I/O	5-bit input/output pins.
	PE2	Input	1-bit input pin.

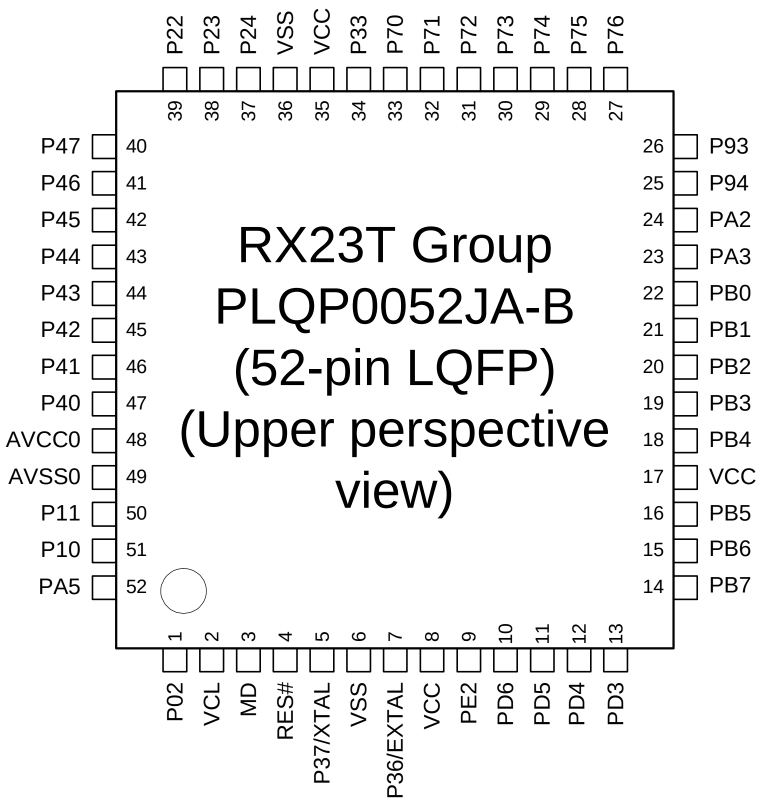
1.5 Pin Assignments

Figure 1.3 to Figure 1.5 show the pin assignments. Table 1.6 to Table 1.8 show the lists of pins and pin functions.



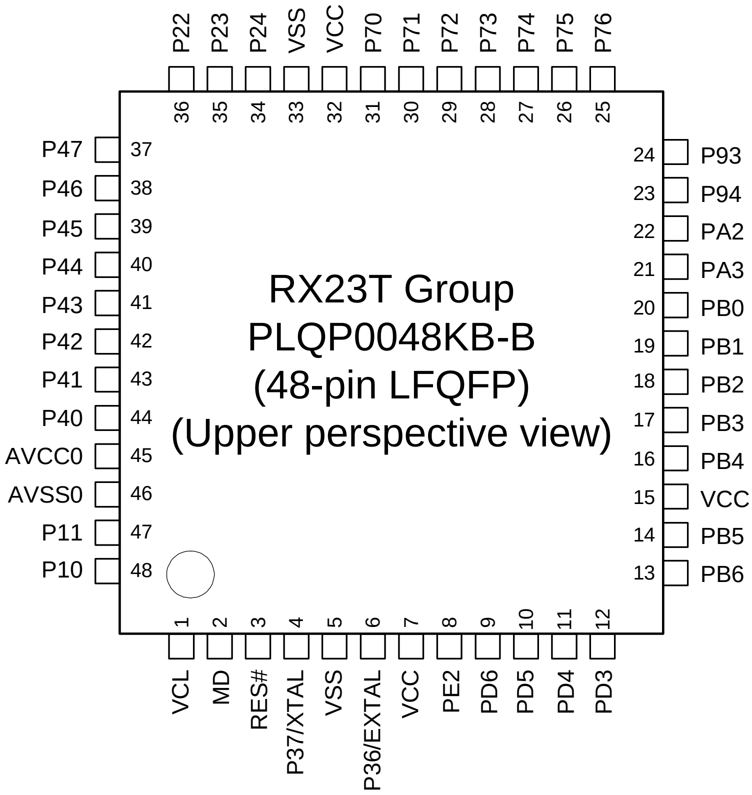
Note: This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (64-Pin LFQFP)”.

Figure 1.3 Pin Assignments of the 64-Pin LFQFP



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 (52-Pin LQFP)”.

Figure 1.4 Pin Assignments of the 52-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 LFQFP)”.

Figure 1.5 Pin Assignments of the 48-Pin LFQFP

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

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE, CAC)	Communications (SCIg, RSPI, RIIC)	Others
1		P02		CTS1#/RTS1#/SS1#	ADST0/IRQ5
2		P00			IRQ2
3	VCL				
4		P01	CACREF		IRQ4
5	MD				FINED
6	RES#				
7	XTAL	P37			
8	VSS				
9	EXTAL	P36			
10	VCC				
11		PE2	POE10#		NMI
12		PD7	TMRI1	SSLA1	
13		PD6	TMO1	SSLA0/CTS1#/RTS1#/SS1#	ADST0/IRQ5
14		PD5	TMRI0	RXD1/SMISO1/SSCL1	IRQ3
15		PD4	TMCI0	SCK1	IRQ2
16		PD3	TMO0	TXD1/SMOSI1/SSDA1	
17		PB7		SCK5	
18		PB6		RXD5/SMISO5/SSCL5	IRQ5
19		PB5		TXD5/SMOSI5/SSDA5	
20	VCC				
21		PB4	POE8#		IRQ3
22	VSS				
23		PB3	MTIOC0A/CACREF	SCK5/RSPCKA	
24		PB2	MTIOC0B/ADSM0	TXD5/SMOSI5/SSDA5/SDA0	
25		PB1	MTIOC0C	RXD5/SMISO5/SSCL5/SCL0	IRQ2
26		PB0	MTIOC0D	MOSIA	
27		PA3	MTIOC2A	SSLA0	
28		PA2	MTIOC2B	CTS5#/RTS5#/SS5#/SSLA1	IRQ4
29		P94	MTIOC0C/TMO1	MISOA	IRQ1
30		P93	MTIOC0B/TMRI1	SCK5/RSPCKA	IRQ0
31		P92	TMCI1	SSLA2	
32		P91		SSLA3	
33		P76	MTIOC4D		
34		P75	MTIOC4C		
35		P74	MTIOC3D		
36		P73	MTIOC4B		
37		P72	MTIOC4A		
38		P71	MTIOC3B		
39		P70	POE0#		IRQ5
40		P33	MTIOC3A/MTCLKA	SSLA3	
41		P32	MTIOC3C/MTCLKB	SSLA2	
42	VCC				
43		P31	MTIOC0A/MTCLKC	SSLA1	
44	VSS				
45		P30	MTIOC0B/MTCLKD	SSLA0	
46		P24	MTIC5U/TMCI2	RSPCKA	COMP0/IRQ3
47		P23	MTIC5V/CACREF/TMO2	MOSIA	COMP1/IRQ4
48		P22	MTIC5W/TMRI2	MISOA	COMP2/IRQ2
49		P47			AN007/CMPC12/ CMPC22
50		P46			AN006/CMPC02
51		P45			AN005/CMPC21
52		P44			AN004/CMPC11
53		P43			AN003/CMPC01

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

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE, CAC)	Communications (SCIg, RSPI, RIIC)	Others
54		P42			AN002/CMPC20
55		P41			AN001/CMPC10
56		P40			AN000/CMPC00
57	AVCC0				
58	VREFH0				
59	VREFL0				
60	AVSS0				
61		P11	MTIOC3A/MTCLKC/TMO3		IRQ1/AN016/ CVREFC0
62		P10	MTCLKD/TMRI3		IRQ0/AN017/ CVREFC1
63		PA5	MTIOC1A/TMCI3	MISOA	
64		PA4	MTIOC1B	RSPCKA	ADTRG0#

Table 1.7 List of Pins and Pin Functions (52-Pin LQFP)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE, CAC)	Communications (SCIg, RSPI, RIIC)	Others
1		P02		CTS1#/RTS1#/SS1#	ADST0/IRQ5
2	VCL				
3	MD				FINED
4	RES#				
5	XTAL	P37			
6	VSS				
7	EXTAL	P36			
8	VCC				
9		PE2	POE10#		NMI
10		PD6	TMO1	SSLA0/CTS1#/RTS1#/SS1#	ADST0/IRQ5
11		PD5	TMRI0	RXD1/SMISO1/SSCL1	IRQ3
12		PD4	TMCI0	SCK1	IRQ2
13		PD3	TMO0	TXD1/SMOSI1/SSDA1	
14		PB7		SCK5	
15		PB6		RXD5/SMISO5/SSCL5	IRQ5
16		PB5		TXD5/SMOSI5/SSDA5	
17	VCC				
18		PB4	POE8#		IRQ3
19		PB3	MTIOC0A/CACREF	SCK5/RSPCKA	
20		PB2	MTIOC0B/ADSM0	TXD5/SMOSI5/SSDA5/SDA0	
21		PB1	MTIOC0C	RXD5/SMISO5/SSCL5/SCL0	IRQ2
22		PB0	MTIOC0D	MOSIA	
23		PA3	MTIOC2A	SSLA0	
24		PA2	MTIOC2B	CTS5#/RTS5#/SS5#/SSLA1	IRQ4
25		P94	MTIOC0C/TMO1	MISOA	IRQ1
26		P93	MTIOC0B/TMRI1	SCK5/RSPCKA	IRQ0
27		P76	MTIOC4D		
28		P75	MTIOC4C		
29		P74	MTIOC3D		
30		P73	MTIOC4B		
31		P72	MTIOC4A		
32		P71	MTIOC3B		
33		P70	POE0#		IRQ5
34		P33	MTIOC3A/MTCLKA	SSLA3	
35	VCC				
36	VSS				
37		P24	MTIC5U/TMCI2	RSPCKA	COMP0/IRQ3
38		P23	MTIC5V/CACREF/TMO2	MOSIA	COMP1/IRQ4
39		P22	MTIC5W/TMRI2	MISOA	COMP2/IRQ2
40		P47			AN007/CMPC12/ CMPC22
41		P46			AN006/CMPC02
42		P45			AN005/CMPC21
43		P44			AN004/CMPC11
44		P43			AN003/CMPC01
45		P42			AN002/CMPC20
46		P41			AN001/CMPC10
47		P40			AN000/CMPC00
48	AVCC0				
49	AVSS0				
50		P11	MTIOC3A/MTCLKC/TMO3		IRQ1/AN016/ CVREFC0
51		P10	MTCLKD/TMRI3		IRQ0/AN017/ CVREFC1
52		PA5	MTIOC1A/TMCI3	MISOA	

Table 1.8 List of Pins and Pin Functions (48-Pin LFQFP)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, TMR, POE, CAC)	Communications (SCIg, RSPI, RIIC)	Others
1	VCL				
2	MD				FINED
3	RES#				
4	XTAL	P37			
5	VSS				
6	EXTAL	P36			
7	VCC				
8		PE2	POE10#		NMI
9		PD6	TMO1	SSLA0/CTS1#/RTS1#/SS1#	ADST0/IRQ5
10		PD5	TMRI0	RXD1/SMISO1/SSCL1	IRQ3
11		PD4	TMCI0	SCK1	IRQ2
12		PD3	TMO0	TXD1/SMOSI1/SSDA1	
13		PB6		RXD5/SMISO5/SSCL5	IRQ5
14		PB5		TXD5/SMOSI5/SSDA5	
15	VCC				
16		PB4	POE8#		IRQ3
17		PB3	MTIOC0A/CACREF	SCK5/RSPCKA	
18		PB2	MTIOC0B/ADSM0	TXD5/SMOSI5/SSDA5/SDA0	
19		PB1	MTIOC0C	RXD5/SMISO5/SSCL5/SCL0	IRQ2
20		PB0	MTIOC0D	MOSIA	
21		PA3	MTIOC2A	SSLA0	
22		PA2	MTIOC2B	CTS5#/RTS5#/SS5#/SSLA1	IRQ4
23		P94	MTIOC0C/TMO1	MISOA	IRQ1
24		P93	MTIOC0B/TMRI1	SCK5/RSPCKA	IRQ0
25		P76	MTIOC4D		
26		P75	MTIOC4C		
27		P74	MTIOC3D		
28		P73	MTIOC4B		
29		P72	MTIOC4A		
30		P71	MTIOC3B		
31		P70	POE0#		IRQ5
32	VCC				
33	VSS				
34		P24	MTIC5U/TMCI2	RSPCKA	COMP0/IRQ3
35		P23	MTIC5V/CACREF/TMO2	MOSIA	COMP1/IRQ4
36		P22	MTIC5W/TMRI2	MISOA	COMP2/IRQ2
37		P47			AN007/CMPC12/ CMPC22
38		P46			AN006/CMPC02
39		P45			AN005/CMPC21
40		P44			AN004/CMPC11
41		P43			AN003/CMPC01
42		P42			AN002/CMPC20
43		P41			AN001/CMPC10
44		P40			AN000/CMPC00
45	AVCC0				
46	AVSS0				
47		P11	MTIOC3A/MTCLKC/TMO3		IRQ1/AN016/ CVREFC0
48		P10	MTCLKD/TMRI3		IRQ0/AN017/ CVREFC1

2. CPU

The RXv2 instruction set architecture (RXv2) has upward compatibility with the RXv1 instruction set architecture (RXv1).

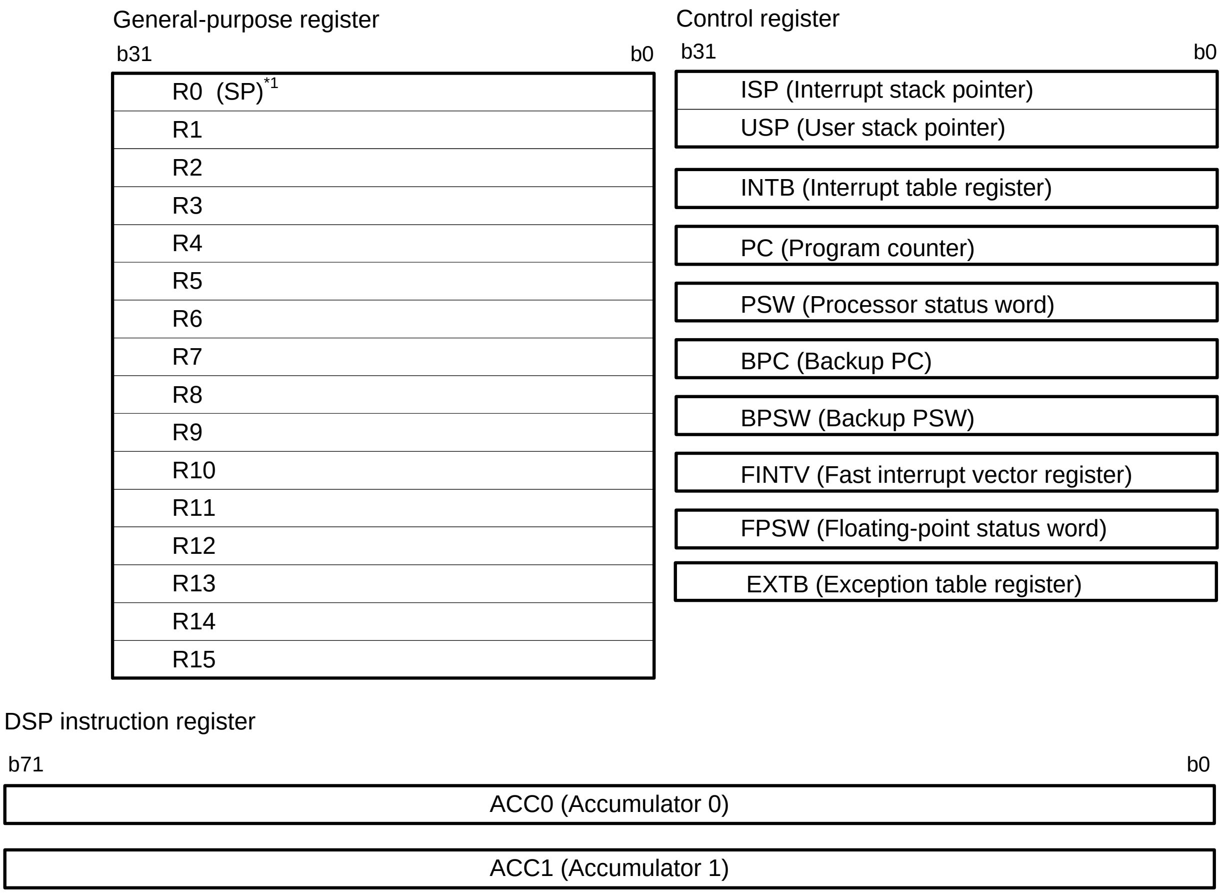
- Adoption of variable-length instruction format
  As with RXv1, the RXv2 CPU has short formats for frequently used instructions, facilitating the development of efficient programs that take up less memory.
- Powerful instruction set
  The RXv2 supports 109 selected instructions. Moreover, DSP instructions and floating-point operation instructions are added, thus realizing high-speed arithmetic processing.
- Versatile addressing modes
  The RXv2 CPU has 11 versatile addressing modes, with register-register operations, register-memory operations, and bitwise operations included. Data transfer between memory locations is also possible.

2.1 Features

- Minimum instruction execution rate: One clock cycle
- Address space: 4-Gbyte linear addresses
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Ten 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 109 instructions/11 addressing modes
  Basic instructions: 75
  Floating-point operation instructions: 11
  DSP instructions: 23
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The RXv2 CPU has sixteen general-purpose registers, ten control registers, and two 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.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

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

2.2.2 Control Registers

This CPU has the following ten control registers.

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

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

ISP b31 … b0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

USP b31 … b0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

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 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.2 Exception Table Register (EXTB)

b31 … b0

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

The exception table register (EXTB) specifies the address where the exception vector table starts.
Set the EXTB to a multiple of 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.3 Interrupt Table Register (INTB)

b31 … b0

Value after reset: Undefined

The interrupt table register (INTB) specifies the address where the interrupt vector table starts.
Set the INTB to a multiple of 4 to reduce the number of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.4 Program Counter (PC)

b31 … b0

Value after reset: Contents of addresses FFFFFFFCh to FFFFFFFFh

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

2.2.2.5 Processor Status Word (PSW)

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

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

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

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

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

C Flag (Carry Flag)

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

Z Flag (Zero Flag)

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

S Flag (Sign Flag)

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

O Flag (Overflow Flag)

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

I Bit (Interrupt Enable)

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

U Bit (Stack Pointer Select)

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

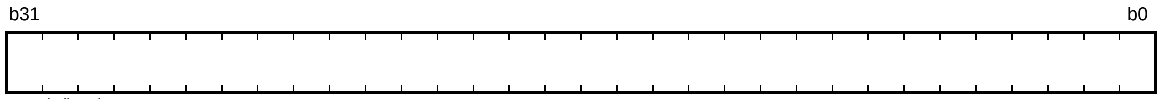
PM Bit (Processor Mode Select)

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

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

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

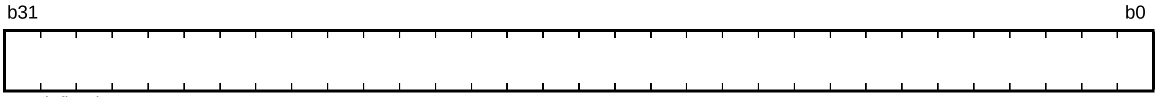
2.2.2.6 Backup PC (BPC)



Value after reset: Undefined

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

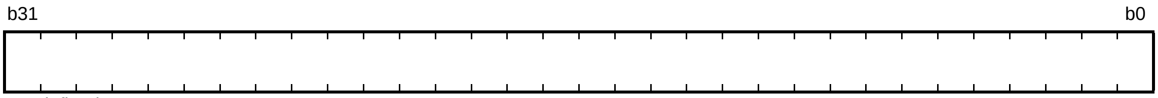
2.2.2.7 Backup PSW (BPSW)



Value after reset: Undefined

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

2.2.2.8 Fast Interrupt Vector Register (FINTV)



Value after reset: Undefined

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

2.2.2.9 Floating-Point Status Word (FPSW)

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

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

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

Bit	Symbol	Bit Name	Description	R/W
b31	FS	Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

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

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

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

These bits specify the floating-point rounding-mode.

Explanation of Floating-Point Rounding Modes

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

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

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

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

- The bit that has been set to 1 is cleared to 0 when the FPU instruction is executed.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Flag (0 Flush Bit of Denormalized Number)

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

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

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

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

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

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

FS Flag (Floating-Point Error Summary Flag)

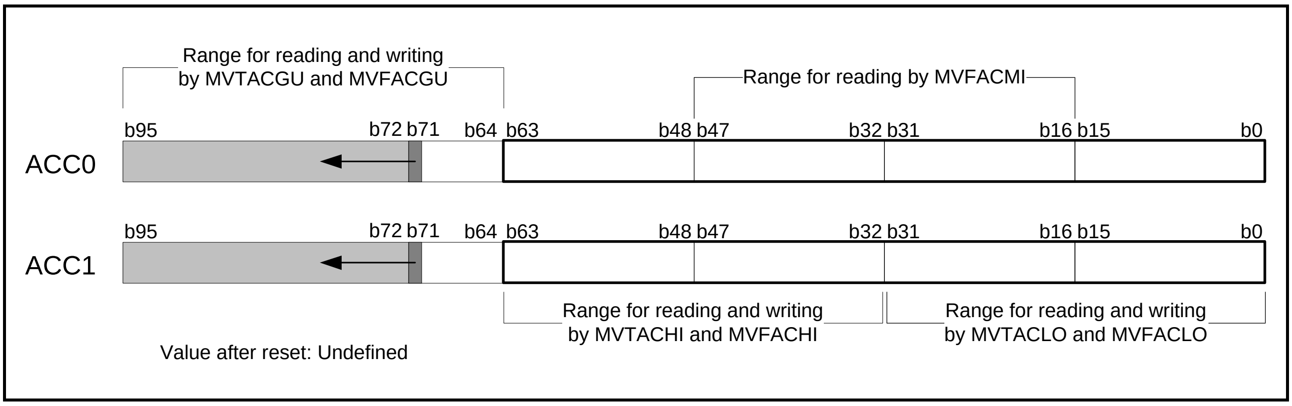
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Accumulator

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

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

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



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

2.3 Processor Mode

The RXv2 CPU supports two processor modes, supervisor and user. These processor modes and the memory protection function enable the realization of a hierarchical CPU resource protection and memory protection mechanism. Each processor mode imposes a level on rights of access to memory and the instructions that can be executed. Supervisor mode carries greater rights than user mode. The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

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

2.3.2 User Mode

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

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

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, 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 RXv2 CPU can handle four types of data: integer, floating-point, bit, and string.
For details, refer to RX Family RXv2 Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

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

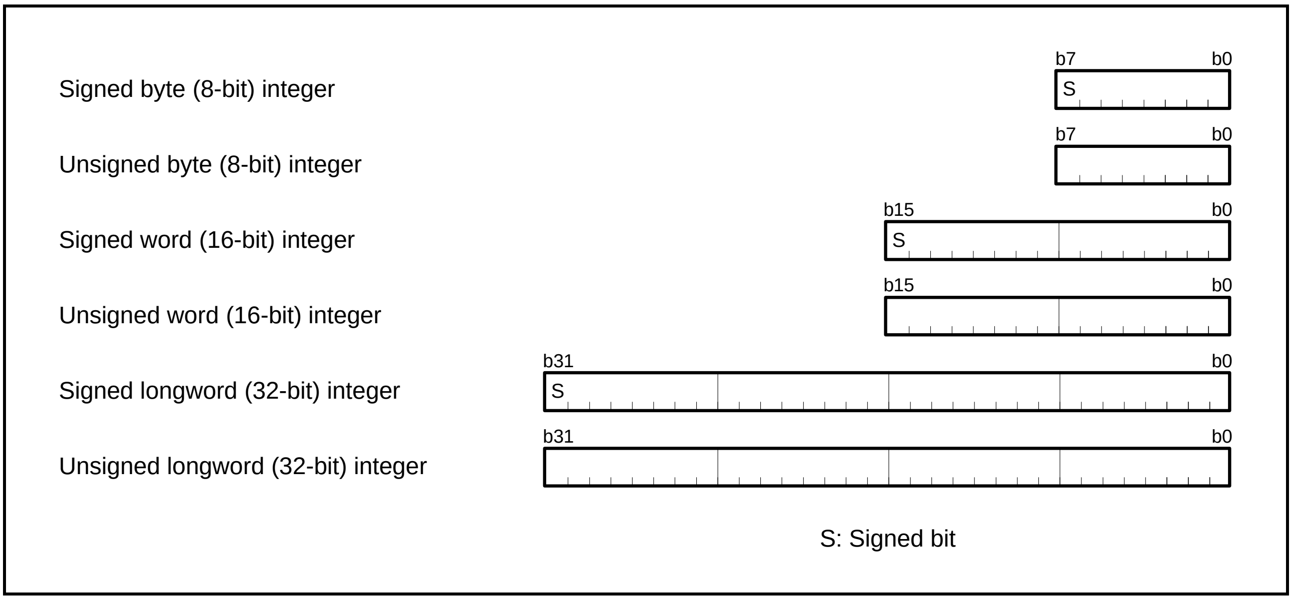


Figure 2.2 Integer

2.4.2 Floating-Points

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

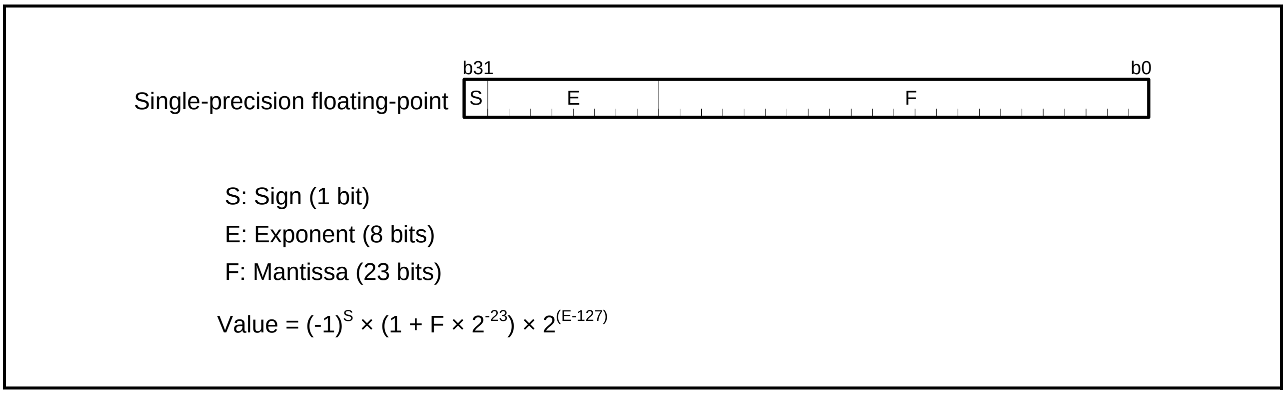


Figure 2.3 Floating-Point

The floating-point format supports the values listed below.

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

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

2.4.3 Bitwise Operations

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

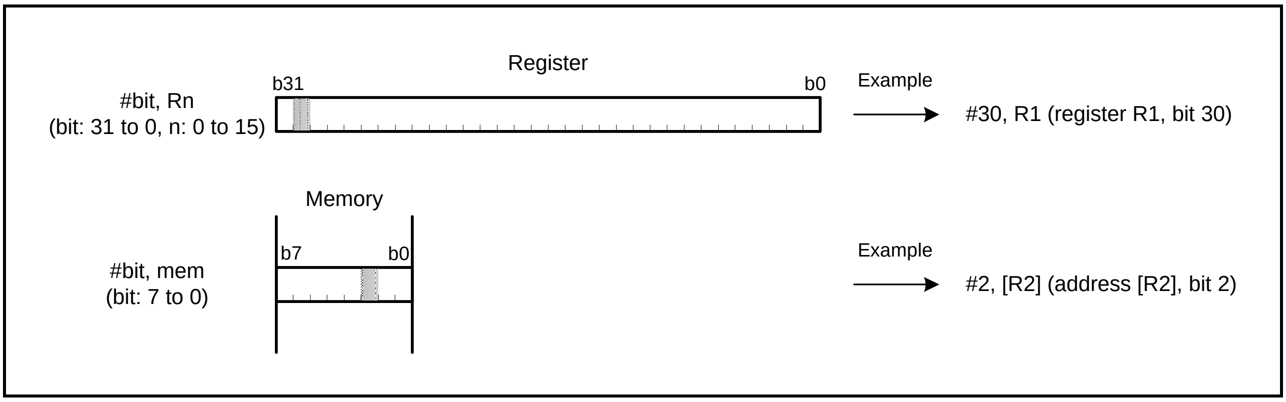


Figure 2.4 Bit

2.4.4 Strings

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

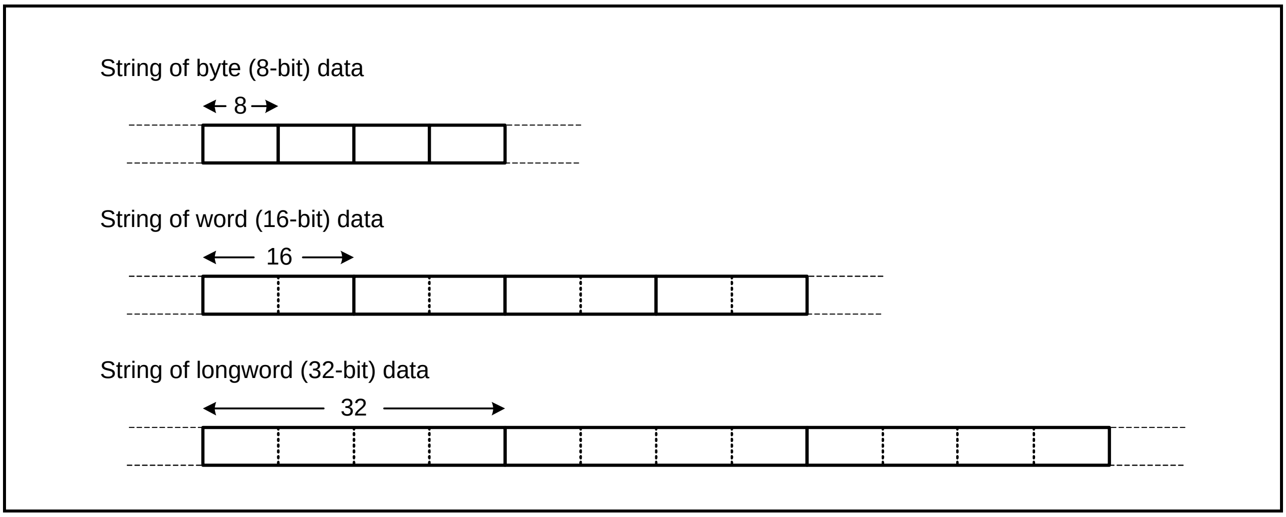


Figure 2.5 String

2.5 Endian

For the RXv2 CPU, instructions are little endian, but the treatment of data is selectable as little or big endian.

2.5.1 Switching the Endian

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

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

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

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

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

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

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

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

Address of dest \ Operation	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

Address of dest \ Operation	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

Address of src \ Operation	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

Address of src \ Operation	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

Address of dest \ Operation	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

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

2.5.2 Access to I/O Registers

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

2.5.3 Notes on Access to I/O Registers

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

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

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

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

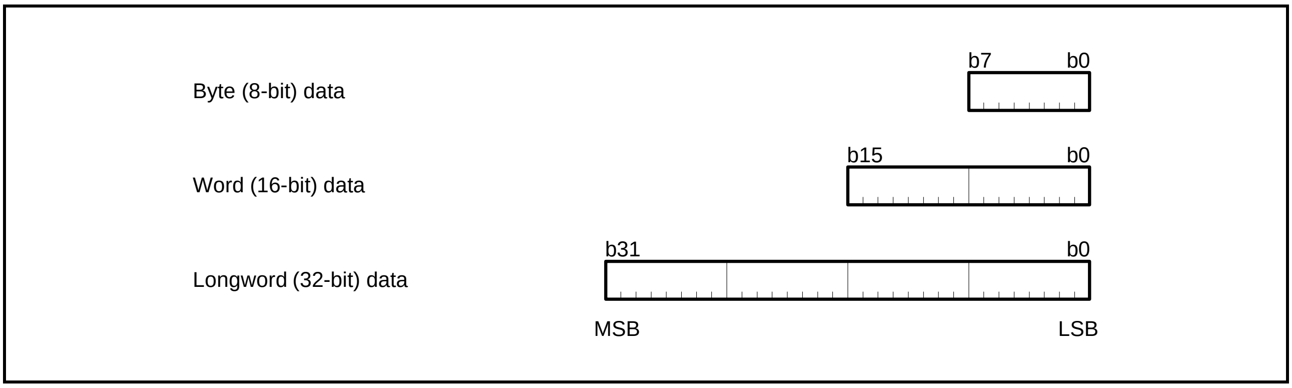


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

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

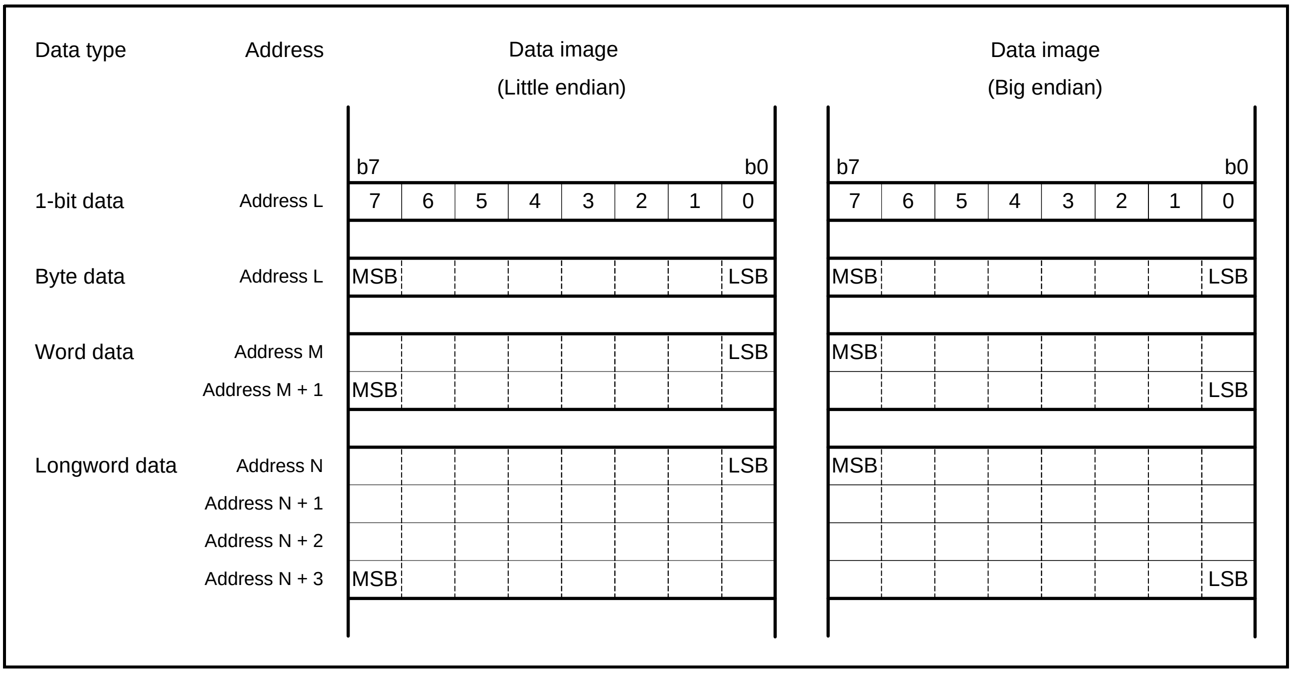


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

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

2.6 Vector Table

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

2.6.1 Exception Vector Table

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

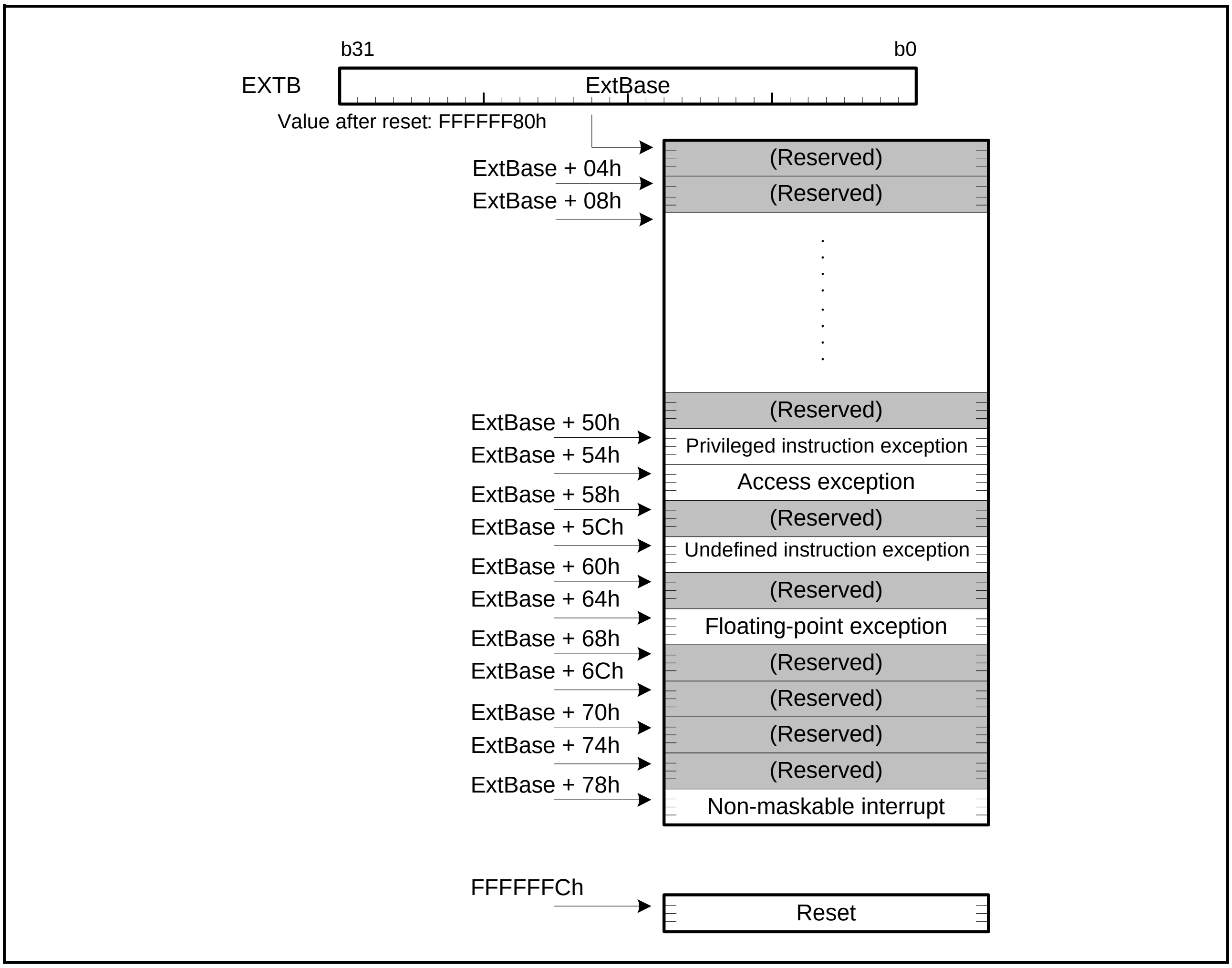


Figure 2.8 Exception Vector Table

2.6.2 Interrupt Vector Table

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

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

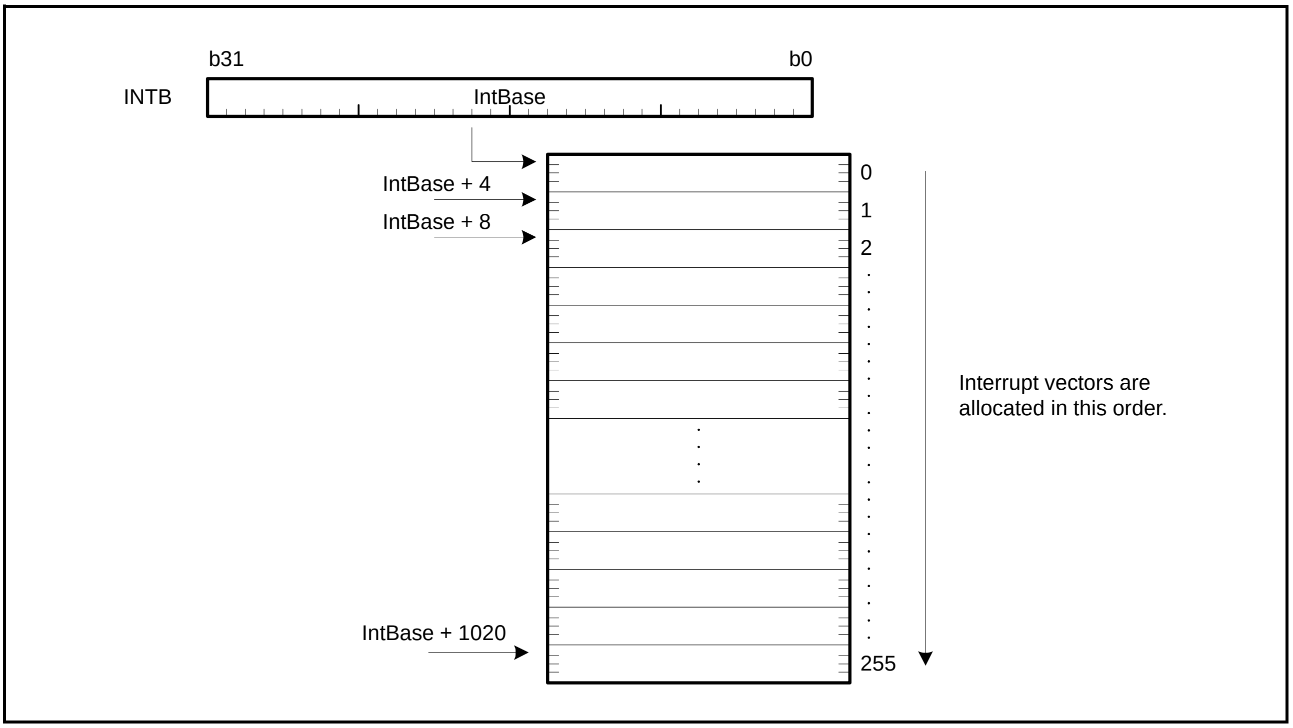


Figure 2.9 Interrupt Vector Table

2.7 Operation of Instructions

2.7.1 Restrictions on RMPA and String-Manipulation Instructions

2.7.1.1 Transfer Size and Data Prefetching

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

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

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

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

2.7.1.2 Access to I/O Registers

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

2.8 Number of Cycles

2.8.1 Instruction and Number of Cycle

Table 2.13 to Table 2.20 show the number of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

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

Table 2.13 Number of Cycles for Arithmetic/logic Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register)	• {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”	1
	• DIV “#IMM, Rd”/“Rs, Rd”	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	2 to 18*1
	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	2
	• SATR	3
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• DIV “[Rs], Rd / dsp[Rs], Rd”	5 to 22
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	4 to 20
	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• RMPA.B	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*2
	• RMPA.W	6+5×floor(n/2)+4×(n%2) n: Number of processing words*2
	• RMPA.L	6+4n n: Number of processing longwords*2

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

Table 2.14 Number of Cycles for Transfer Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Transfer instructions (register-register, immediate-register)	• MOV "#IMM, Rd"/"Rs, Rd" • {MOVU, REVL, REVW} "Rs, Rd" • SCC*nd* "Rd" • {STNZ, STZ} "#IMM, Rd"/ "Rs, Rd"	1
	• XCHG "Rs, Rd"	2
Transfer instructions (load operation)	• {MOV, MOVU} "[Rs], Rd"/"dsp[Rs], Rd"/"[Rs+], Rd"/"[-Rs], Rd"/"[Ri, Rb], Rd" • LDL "[Rs], Rd" • POP "Rd"	Throughput: 1 Latency: 2*1
	• POPC "CR"	Throughput: 3 Latency: 4*1
	• POPM "Rd-Rd2"	Throughput: n Latency: n+1 n: Number of registers*1, *2
Transfer instructions (store operation)	• MOV "Rs, [Rd]"/"Rs, dsp[Rd]"/"Rs, [Rd+]"/"Rs, [-Rd]"/"Rs, [Ri, Rb]"/"#IMM, dsp[Rd]"/"#IMM, [Rd]" • PUSH "Rs" • PUSHC "CR" • SCCnd "[Rd]"/"dsp[Rd]" • STC "Rs, [Rd]"	1
	• PUSHM "Rs-Rs2"	n n: Number of registers*3
Transfer instructions (memory-register)	• XCHG "[Rs], Rd"/"dsp[Rs], Rd"	2
Transfer instructions (memory-memory)	• MOV "[Rs], [Rd]"/"dsp[Rs], [Rd]"/"[Rs], dsp[Rd]"/"dsp[Rs], dsp[Rd]" • PUSH "[Rs]"/"dsp[Rs]"	3

Note 1. 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.

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

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

Table 2.15 Number of Cycles for Bit Manipulation Instructions

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

Table 2.16 Number of Cycles for Branch Instructions

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

?: Conditional operator

Note 1. When the load data is used by the subsequent instruction, the 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.

Table 2.17 Number of Cycles for Floating-Point Operation Instructions

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

Table 2.18 Number of Cycles for DSP Instructions

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

Table 2.19 Number of Cycles for String Manipulation Instructions

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

?: Conditional operator

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

Table 2.20 Number of Cycles for System Manipulation Instructions

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

2.8.2 Numbers of Cycles for Response to Interrupts

Table 2.21 lists numbers of cycles taken by processing for response to interrupts.

Table 2.21 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 2.21 will be applicable when access to memory from the CPU is processed with no waiting. The RAM and code flash memory in this MCU allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13 to Table 2.20.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 13.3.1, Acceptance Timing and Saved PC Value.

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) 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	**Operating Mode**
Low	Boot mode (SCI)
High	Single-chip mode

Note 1. Do not change the level on the MD pin while the MCU is operating.

The endian is selectable in single-chip mode. Endian is set by the MDE.MDE[2:0] bits in the option-setting memory. For the correspondence between the setting and endian, see Table 3.2.

Table 3.2 Selection of Endian

MDE.MDE[2:0] Bit Setting	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 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.

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) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 34, Flash Memory.
When a reset is released while the MD pin is low, boot mode is selected.

3.3.2.1 Boot Mode (SCI)

When a reset is released while the MD pin is low, boot mode (SCI) is selected. For details on boot mode (SCI), refer to section 34.7.1, Boot Mode (SCI).

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.

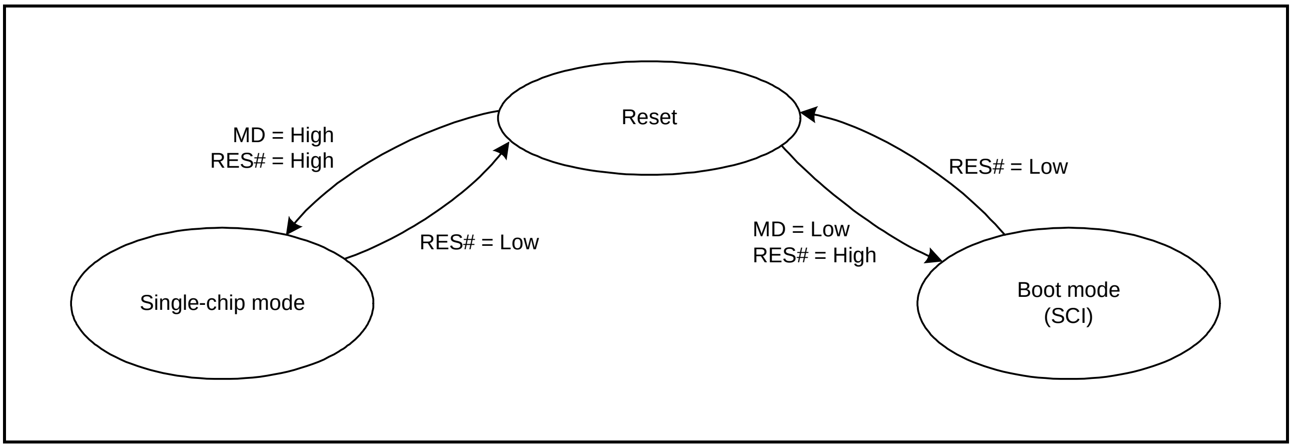


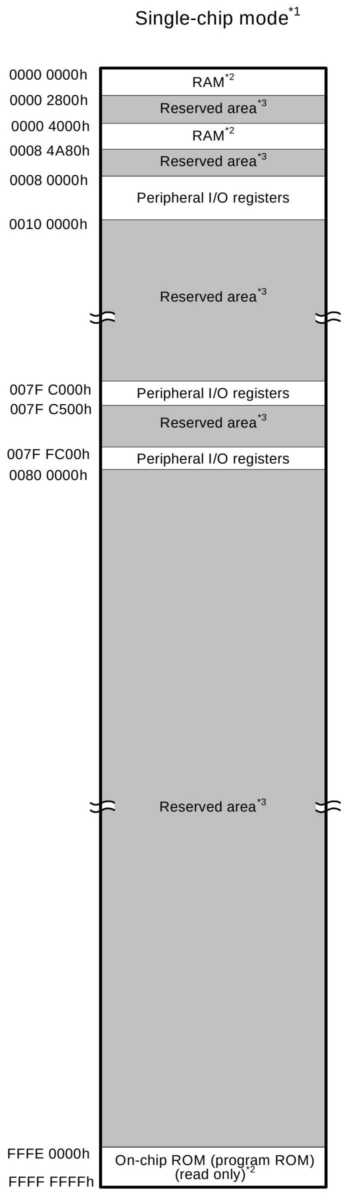
Figure 3.1 Mode-Setting Pin Levels and Operating Modes

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 maps in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.



Note 1. The address space in boot mode 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**
128 Kbytes	FFFE 0000h to FFFF FFFFh	12 Kbytes	0000 0000h to 0000 27FFh 0000 4000h to 0000 4A7Fh
64 Kbytes	FFFF 0000h to FFFF FFFFh		

Note: See Table 1.3 and Table 1.4 List of Products, for the product type name.

Note 3. Reserved areas should not be accessed.

Figure 4.1 Memory Map in Each Operating Mode

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.
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.

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

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

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

(5) Notes on Sleep Mode and Mode Transitions

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

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK	section 3.
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK	section 3.
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK	section 11.
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK	section 11.
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK	section 11.
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK	section 11.
0008 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 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK	section 9.
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK	section 9.
0008 0031h	SYSTEM	Memory Wait Cycle Setting Register	MEMWAIT	8	8	3 ICLK	section 9.
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK	section 9.
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK	section 9.
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK	section 9.
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3ICLK	section 9.
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK	section 9.
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK	section 9.
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK	section 9.
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK	section 11.
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK	section 9.
0008 00A5h	SYSTEM	High-Speed On-Chip Oscillator Wait Control Register	HOCOWTCR	8	8	3ICLK	section 9.
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK	section 6.
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK	section 6.
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK	section 8.
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK	section 8.
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK	section 8.
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK	section 8.
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK	section 12.
0008 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 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 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK	section 16.
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK	section 16.
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK	section 16.
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK	section 16.
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK	section 16.
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK	section 16.
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK	section 16.
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK	section 16.
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK	section 16.
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK	section 16.
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK	section 16.

Table 5.1 List of I/O Registers (Address Order) (2 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK	section 16.
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK	section 16.
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK	section 16.
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK	section 16.
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK	section 16.
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK	section 16.
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK	section 16.
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK	section 16.
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK	section 16.
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK	section 16.
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK	section 16.
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK	section 16.
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK	section 16.
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK	section 16.
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK	section 16.
0008 7010h	ICU	Interrupt Request Register 016	IR016	8	8	2 ICLK	section 15.
0008 7017h	ICU	Interrupt Request Register 023	IR023	8	8	2 ICLK	section 15.
0008 701Bh	ICU	Interrupt Request Register 027	IR027	8	8	2 ICLK	section 15.
0008 701Ch	ICU	Interrupt Request Register 028	IR028	8	8	2 ICLK	section 15.
0008 701Dh	ICU	Interrupt Request Register 029	IR029	8	8	2 ICLK	section 15.
0008 701Eh	ICU	Interrupt Request Register 030	IR030	8	8	2 ICLK	section 15.
0008 701Fh	ICU	Interrupt Request Register 031	IR031	8	8	2 ICLK	section 15.
0008 7020h	ICU	Interrupt Request Register 032	IR032	8	8	2 ICLK	section 15.
0008 7021h	ICU	Interrupt Request Register 033	IR033	8	8	2 ICLK	section 15.
0008 7022h	ICU	Interrupt Request Register 034	IR034	8	8	2 ICLK	section 15.
0008 702Ch	ICU	Interrupt Request Register 044	IR044	8	8	2 ICLK	section 15.
0008 702Dh	ICU	Interrupt Request Register 045	IR045	8	8	2 ICLK	section 15.
0008 702Eh	ICU	Interrupt Request Register 046	IR046	8	8	2 ICLK	section 15.
0008 702Fh	ICU	Interrupt Request Register 047	IR047	8	8	2 ICLK	section 15.
0008 7039h	ICU	Interrupt Request Register 057	IR057	8	8	2 ICLK	section 15.
0008 7040h	ICU	Interrupt Request Register 064	IR064	8	8	2 ICLK	section 15.
0008 7041h	ICU	Interrupt Request Register 065	IR065	8	8	2 ICLK	section 15.
0008 7042h	ICU	Interrupt Request Register 066	IR066	8	8	2 ICLK	section 15.
0008 7043h	ICU	Interrupt Request Register 067	IR067	8	8	2 ICLK	section 15.
0008 7044h	ICU	Interrupt Request Register 068	IR068	8	8	2 ICLK	section 15.
0008 7045h	ICU	Interrupt Request Register 069	IR069	8	8	2 ICLK	section 15.
0008 7058h	ICU	Interrupt Request Register 088	IR088	8	8	2 ICLK	section 15.
0008 7059h	ICU	Interrupt Request Register 089	IR089	8	8	2 ICLK	section 15.
0008 7066h	ICU	Interrupt Request Register 102	IR102	8	8	2 ICLK	section 15.
0008 7067h	ICU	Interrupt Request Register 103	IR103	8	8	2 ICLK	section 15.
0008 706Ch	ICU	Interrupt Request Register 108	IR108	8	8	2 ICLK	section 15.
0008 706Dh	ICU	Interrupt Request Register 109	IR109	8	8	2 ICLK	section 15.
0008 706Eh	ICU	Interrupt Request Register 110	IR110	8	8	2 ICLK	section 15.
0008 7072h	ICU	Interrupt Request Register 114	IR114	8	8	2 ICLK	section 15.
0008 7073h	ICU	Interrupt Request Register 115	IR115	8	8	2 ICLK	section 15.
0008 7074h	ICU	Interrupt Request Register 116	IR116	8	8	2 ICLK	section 15.
0008 7075h	ICU	Interrupt Request Register 117	IR117	8	8	2 ICLK	section 15.
0008 7076h	ICU	Interrupt Request Register 118	IR118	8	8	2 ICLK	section 15.
0008 7077h	ICU	Interrupt Request Register 119	IR119	8	8	2 ICLK	section 15.
0008 7078h	ICU	Interrupt Request Register 120	IR120	8	8	2 ICLK	section 15.
0008 7079h	ICU	Interrupt Request Register 121	IR121	8	8	2 ICLK	section 15.

Table 5.1 List of I/O Registers (Address Order) (3 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
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 70A8h	ICU	Interrupt Request Register 168	IR168	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.
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 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 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.

Table 5.1 List of I/O Registers (Address Order) (4 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
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.
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 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 716Ch	ICU	DTC Activation Enable Register 108	DTCER108	8	8	2 ICLK	section 14.
0008 716Dh	ICU	DTC Activation Enable Register 109	DTCER109	8	8	2 ICLK	section 14.
0008 716Eh	ICU	DTC Activation Enable Register 110	DTCER110	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 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 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.

Table 5.1 List of I/O Registers (Address Order) (5 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
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 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 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 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 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 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.
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 736Ch	ICU	Interrupt Source Priority Register 108	IPR108	8	8	2 ICLK	section 14.
0008 736Dh	ICU	Interrupt Source Priority Register 109	IPR109	8	8	2 ICLK	section 14.
0008 736Eh	ICU	Interrupt Source Priority Register 110	IPR110	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.

Table 5.1 List of I/O Registers (Address Order) (6 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
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 73A8h	ICU	Interrupt Source Priority Register 168	IPR168	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 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 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 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 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 Status Clear Register	NMICLR	8	8	2 ICLK	section 14.
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK	section 14.
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK	section 14.
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK	section 14.
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 or 3 PCLKB	section 23.
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2 or 3 PCLKB	section 23.
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 or 3 PCLKB	section 24.
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 or 3 PCLKB	section 24.
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 or 3 PCLKB	section 24.
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 or 3 PCLKB	section 24.
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 or 3 PCLKB	section 24.
0008 80C0h	DA	D/A Data Register 0	DADR0	16	16	2 or 3 PCLKB	section 30.
0008 80C4h	DA	D/A Control Register	DACR	8	8	2 or 3 PCLKB	section 30.
0008 80C5h	DA	DADR0 Format Select Register	DADPR	8	8	2 or 3 PCLKB	section 30.

Table 5.1 List of I/O Registers (Address Order) (7 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	section 22.
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8*1	2 or 3 PCLKB	section 22.
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	section 22.
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8*1	2 or 3 PCLKB	section 22.
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2 or 3 PCLKB	section 22.
0008 8209h	TMR1	Timer Counter	TCNT	8	8*1	2 or 3 PCLKB	section 22.
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	section 22.
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8*1	2 or 3 PCLKB	section 22.
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	section 22.
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8*1	2 or 3 PCLKB	section 22.
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	section 22.
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8*1	2 or 3 PCLKB	section 22.
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2 or 3 PCLKB	section 22.
0008 8219h	TMR3	Timer Counter	TCNT	8	8*1	2 or 3 PCLKB	section 22.
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	section 22.
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8*1	2 or 3 PCLKB	section 22.
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 or 3 PCLKB	section 28.
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 or 3 PCLKB	section 28.
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 or 3 PCLKB	section 28.
0008 8300h	RIIC0	I^2C Bus Control Register 1	ICCR1	8	8	2 or 3 PCLKB	section 26.
0008 8301h	RIIC0	I^2C Bus Control Register 2	ICCR2	8	8	2 or 3 PCLKB	section 26.
0008 8302h	RIIC0	I^2C Bus Mode Register 1	ICMR1	8	8	2 or 3 PCLKB	section 26.
0008 8303h	RIIC0	I^2C Bus Mode Register 2	ICMR2	8	8	2 or 3 PCLKB	section 26.
0008 8304h	RIIC0	I^2C Bus Mode Register 3	ICMR3	8	8	2 or 3 PCLKB	section 26.
0008 8305h	RIIC0	I^2C Bus Function Enable Register	ICFER	8	8	2 or 3 PCLKB	section 26.
0008 8306h	RIIC0	I^2C Bus Status Enable Register	ICSER	8	8	2 or 3 PCLKB	section 26.
0008 8307h	RIIC0	I^2C Bus Interrupt Enable Register	ICIER	8	8	2 or 3 PCLKB	section 26.
0008 8308h	RIIC0	I^2C Bus Status Register 1	ICSR1	8	8	2 or 3 PCLKB	section 26.
0008 8309h	RIIC0	I^2C Bus Status Register 2	ICSR2	8	8	2 or 3 PCLKB	section 26.
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 or 3 PCLKB	section 26.
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 or 3 PCLKB	section 26.
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 or 3 PCLKB	section 26.
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 or 3 PCLKB	section 26.
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 or 3 PCLKB	section 26.
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 or 3 PCLKB	section 26.
0008 8310h	RIIC0	I^2C Bus Bit Rate Low-Level Register	ICBRL	8	8	2 or 3 PCLKB	section 26.
0008 8311h	RIIC0	I^2C Bus Bit Rate High-Level Register	ICBRH	8	8	2 or 3 PCLKB	section 26.
0008 8312h	RIIC0	I^2C Bus Transmit Data Register	ICDRT	8	8	2 or 3 PCLKB	section 26.
0008 8313h	RIIC0	I^2C Bus Receive Data Register	ICDRR	8	8	2 or 3 PCLKB	section 26.
0008 8380h	RSPI0	RSPI Control Register	SPCR	8	8	2 or 3 PCLKB	section 27.
0008 8381h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	2 or 3 PCLKB	section 27.
0008 8382h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	2 or 3 PCLKB	section 27.
0008 8383h	RSPI0	RSPI Status Register	SPSR	8	8	2 or 3 PCLKB	section 27.

Table 5.1 List of I/O Registers (Address Order) (8 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
						ICLK ≥ PCLK	
0008 8384h	RSPI0	RSPI Data Register	SPDR	32	16, 32	2 or 3 PCLKB	section 27.
0008 8388h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	2 or 3 PCLKB	section 27.
0008 8389h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	2 or 3 PCLKB	section 27.
0008 838Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	2 or 3 PCLKB	section 27.
0008 838Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	2 or 3 PCLKB	section 27.
0008 838Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	2 or 3 PCLKB	section 27.
0008 838Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 or 3 PCLKB	section 27.
0008 838Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	2 or 3 PCLKB	section 27.
0008 838Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	2 or 3 PCLKB	section 27.
0008 8390h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	2 or 3 PCLKB	section 27.
0008 8392h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	2 or 3 PCLKB	section 27.
0008 8394h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	2 or 3 PCLKB	section 27.
0008 8396h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	2 or 3 PCLKB	section 27.
0008 8398h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	2 or 3 PCLKB	section 27.
0008 839Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	2 or 3 PCLKB	section 27.
0008 839Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	2 or 3 PCLKB	section 27.
0008 839Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	2 or 3 PCLKB	section 27.
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 or 3 PCLKB	section 29.
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2 or 3 PCLKB	section 29.
0008 9006h	S12AD	A/D Channel Select Register A1	ADANSA1	16	16	2 or 3 PCLKB	section 29.
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Select Register 0	ADADS0	16	16	2 or 3 PCLKB	section 29.
0008 900Ah	S12AD	A/D-Converted Value Addition/Average Function Select Register 1	ADADS1	16	16	2 or 3 PCLKB	section 29.
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2 or 3 PCLKB	section 29.
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 or 3 PCLKB	section 29.
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2 or 3 PCLKB	section 29.
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2 or 3 PCLKB	section 29.
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2 or 3 PCLKB	section 29.
0008 9016h	S12AD	A/D Channel Select Register B1	ADANSB1	16	16	2 or 3 PCLKB	section 29.
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2 or 3 PCLKB	section 29.
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 or 3 PCLKB	section 29.
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2 or 3 PCLKB	section 29.
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2 or 3 PCLKB	section 29.
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2 or 3 PCLKB	section 29.
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2 or 3 PCLKB	section 29.
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2 or 3 PCLKB	section 29.
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2 or 3 PCLKB	section 29.
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2 or 3 PCLKB	section 29.
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2 or 3 PCLKB	section 29.
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2 or 3 PCLKB	section 29.
0008 9040h	S12AD	A/D Data Register 16	ADDR16	16	16	2 or 3 PCLKB	section 29.
0008 9042h	S12AD	A/D Data Register 17	ADDR17	16	16	2 or 3 PCLKB	section 29.
0008 9066h	S12AD	A/D Sample-and-hold Circuit Control Register	ADSHCR	16	16	2 or 3 PCLKB	section 29.
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2 or 3 PCLKB	section 29.
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2 or 3 PCLKB	section 29.
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2 or 3 PCLKB	section 29.
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2 or 3 PCLKB	section 29.
0008 908Ah	S12AD	A/D High-Side/Low-Side Reference Voltage Control Register	ADHVREFCNT	8	8	2 or 3 PCLKB	section 29.
0008 90DDh	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2 or 3 PCLKB	section 29.
0008 90DFh	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 or 3 PCLKB	section 29.

Table 5.1 List of I/O Registers (Address Order) (9 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 or 3 PCLKB	section 29.
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 or 3 PCLKB	section 29.
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 or 3 PCLKB	section 29.
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 or 3 PCLKB	section 29.
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 or 3 PCLKB	section 29.
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 or 3 PCLKB	section 29.
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 or 3 PCLKB	section 29.
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 or 3 PCLKB	section 29.
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 25.
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 25.
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 25.
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 25.
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 25.
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 25.
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 25.
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 25.
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 25.
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 25.
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 25.
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 25.
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	section 25.
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 25.
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	section 25.
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	section 25.
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	section 25.
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	section 25.
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	section 25.
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	section 25.
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	section 25.
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 25.
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 25.
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 25.
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 25.
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 25.
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 25.
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 25.
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 25.
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 25.
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 25.
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 25.
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 25.
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	section 25.
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 25.
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	4 or 5 PCLKB	section 25.
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	section 25.
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	section 25.
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	4 or 5 PCLKB	section 25.
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	section 25.
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	section 25.
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	section 25.
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 or 3 PCLKB	section 10.

Table 5.1 List of I/O Registers (Address Order) (10 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 or 3 PCLKB	section 10.
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 or 3 PCLKB	section 10.
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 or 3 PCLKB	section 10.
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 or 3 PCLKB	section 10.
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 or 3 PCLKB	section 10.
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 or 3 PCLKB	section 10.
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 or 3 PCLKB	section 10.
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 or 3 PCLKB	section 32.
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 or 3 PCLKB	section 32.
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 or 3 PCLKB	section 32.
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C040h	PORT0	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C047h	PORT7	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C049h	PORT9	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C04Ah	PORTA	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	2 or 3 PCLKB	section 18.
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.

Table 5.1 List of I/O Registers (Address Order) (11 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 C080h	PORT0	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C082h	PORT1	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C086h	PORT3	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C087h	PORT3	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C08Eh	PORT7	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C08Fh	PORT7	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C092h	PORT9	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C093h	PORT9	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C094h	PORTA	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C095h	PORTA	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C09Ah	PORTD	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C09Bh	PORTD	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C0C0h	PORT0	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C4h	PORT4	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C7h	PORT7	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C9h	PORT9	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CAh	PORTA	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E3h	PORT3	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 18.
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 or 3 PCLKB	section 19.
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2 or 3 PCLKB	section 19.
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2 or 3 PCLKB	section 19.
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2 or 3 PCLKB	section 19.
0008 C148h	MPC	P10 Pin Function Control Register	P10PFS	8	8	2 or 3 PCLKB	section 19.
0008 C149h	MPC	P11 Pin Function Control Register	P11PFS	8	8	2 or 3 PCLKB	section 19.
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 or 3 PCLKB	section 19.
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2 or 3 PCLKB	section 19.
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2 or 3 PCLKB	section 19.
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2 or 3 PCLKB	section 19.
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2 or 3 PCLKB	section 19.
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2 or 3 PCLKB	section 19.
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2 or 3 PCLKB	section 19.
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2 or 3 PCLKB	section 19.
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2 or 3 PCLKB	section 19.
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2 or 3 PCLKB	section 19.

Table 5.1 List of I/O Registers (Address Order) (12 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2 or 3 PCLKB	section 19.
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2 or 3 PCLKB	section 19.
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2 or 3 PCLKB	section 19.
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2 or 3 PCLKB	section 19.
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2 or 3 PCLKB	section 19.
0008 C178h	MPC	P70 Pin Function Control Register	P70PFS	8	8	2 or 3 PCLKB	section 19.
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2 or 3 PCLKB	section 19.
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2 or 3 PCLKB	section 19.
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2 or 3 PCLKB	section 19.
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2 or 3 PCLKB	section 19.
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2 or 3 PCLKB	section 19.
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2 or 3 PCLKB	section 19.
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2 or 3 PCLKB	section 19.
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2 or 3 PCLKB	section 19.
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2 or 3 PCLKB	section 19.
0008 C18Ch	MPC	P94 Pin Function Control Register	P94PFS	8	8	2 or 3 PCLKB	section 19.
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 or 5 PCLKB	section 6.
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 or 5 PCLKB	section 6.
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 or 5 PCLKB	section 9.
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 or 5 PCLKB	section 8.
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4 or 5 PCLKB	section 8.
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 or 5 PCLKB	section 8.
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4 or 5 PCLKB	section 8.
0008 C4C0h	POE	Input Level Control/Status Register 1	ICSR1	16	8, 16	2 or 3 PCLKB	section 21.
0008 C4C2h	POE	Output Level Control/Status Register 1	OCSR1	16	8, 16	2 or 3 PCLKB	section 21.
0008 C4C8h	POE	Input Level Control/Status Register 3	ICSR3	16	8, 16	2 or 3 PCLKB	section 21.
0008 C4CAh	POE	Software Port Output Enable Register	SPOER	8	8	2 or 3 PCLKB	section 21.
0008 C4CBh	POE	Port Output Enable Control Register 1	POECR1	8	8	2 or 3 PCLKB	section 21.
0008 C4CCh	POE	Port Output Enable Control Register 2	POECR2	16	16	2 or 3 PCLKB	section 21.
0008 C4D0h	POE	Port Output Enable Control Register 4	POECR4	16	16	2 or 3 PCLKB	section 21.
0008 C4D2h	POE	Port Output Enable Control Register 5	POECR5	16	16	2 or 3 PCLKB	section 21.
0008 C4D6h	POE	Input Level Control/Status Register 4	ICSR4	16	8, 16	2 or 3 PCLKB	section 21.
0008 C4DAh	POE	Active Level Setting Register 1	ALR1	16	8, 16	2 or 3 PCLKB	section 21.

Table 5.1 List of I/O Registers (Address Order) (13 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
0008 C4DCh	POE	Input Level Control/Status Register 6	ICSR6	16	16	2 or 3 PCLKB	section 21.
0008 C4E6h	POE	Port Output Enable Comparator Detection Flag Register	POECMPFR	16	16	2 or 3 PCLKB	section 21.
0008 C4E8h	POE	Port Output Enable Comparator Request Select Register	POECMPSEL	16	16	2 or 3 PCLKB	section 21.
000A 0C80h	CMPC0	Comparator Control Register 0	CMPCTL	8	8	1 or 2 PCLKB	section 31.
000A 0C84h	CMPC0	Comparator Input Select Register 0	CMPSEL0	8	8	1 or 2 PCLKB	section 31.
000A 0C88h	CMPC0	Comparator Reference Voltage Select Register 0	CMPSEL1	8	8	1 or 2 PCLKB	section 31.
000A 0C8Ch	CMPC0	Comparator Output Monitor Register 0	CMPMON	8	8	1 or 2 PCLKB	section 31.
000A 0C90h	CMPC0	Comparator External Output Enable Register 0	CMPIOC	8	8	1 or 2 PCLKB	section 31.
000A 0CA0h	CMPC1	Comparator Control Register 1	CMPCTL	8	8	1 or 2 PCLKB	section 31.
000A 0CA4h	CMPC1	Comparator Input Select Register 1	CMPSEL0	8	8	1 or 2 PCLKB	section 31.
000A 0CA8h	CMPC1	Comparator Reference Voltage Select Register 1	CMPSEL1	8	8	1 or 2 PCLKB	section 31.
000A 0CACh	CMPC1	Comparator Output Monitor Register 1	CMPMON	8	8	1 or 2 PCLKB	section 31.
000A 0CB0h	CMPC1	Comparator External Output Enable Register 1	CMPIOC	8	8	1 or 2 PCLKB	section 31.
000A 0CC0h	CMPC2	Comparator Control Register 2	CMPCTL	8	8	1 or 2 PCLKB	section 31.
000A 0CC4h	CMPC2	Comparator Input Select Register 2	CMPSEL0	8	8	1 or 2 PCLKB	section 31.
000A 0CC8h	CMPC2	Comparator Reference Voltage Select Register 2	CMPSEL1	8	8	1 or 2 PCLKB	section 31.
000A 0CCCh	CMPC2	Comparator Output Monitor Register 2	CMPMON	8	8	1 or 2 PCLKB	section 31.
000A 0CD0h	CMPC2	Comparator External Output Enable Register 2	CMPIOC	8	8	1 or 2 PCLKB	section 31.
000C 1200h	MTU3	Timer Control Register	TCR	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1201h	MTU4	Timer Control Register	TCR	8	8	4 or 5 PCLKA	section 20.
000C 1202h	MTU3	Timer Mode Register 1	TMDR1	8	8, 16	4 or 5 PCLKA	section 20.
000C 1203h	MTU4	Timer Mode Register 1	TMDR1	8	8	4 or 5 PCLKA	section 20.
000C 1204h	MTU3	Timer I/O Control Register H	TIORH	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1205h	MTU3	Timer I/O Control Register L	TIORL	8	8	4 or 5 PCLKA	section 20.
000C 1206h	MTU4	Timer I/O Control Register H	TIORH	8	8, 16	4 or 5 PCLKA	section 20.
000C 1207h	MTU4	Timer I/O Control Register L	TIORL	8	8	4 or 5 PCLKA	section 20.
000C 1208h	MTU3	Timer Interrupt Enable Register	TIER	8	8, 16	4 or 5 PCLKA	section 20.
000C 1209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	4 or 5 PCLKA	section 20.
000C 120Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	4 or 5 PCLKA	section 20.
000C 120Dh	MTU	Timer Gate Control Register	TGCRA	8	8	4 or 5 PCLKA	section 20.
000C 120Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8, 16	4 or 5 PCLKA	section 20.
000C 120Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	4 or 5 PCLKA	section 20.
000C 1210h	MTU3	Timer Counter	TCNT	16	16, 32	4 or 5 PCLKA	section 20.
000C 1212h	MTU4	Timer Counter	TCNT	16	16	4 or 5 PCLKA	section 20.
000C 1214h	MTU	Timer Cycle Data Register A	TCDRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 1216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	4 or 5 PCLKA	section 20.
000C 1218h	MTU3	Timer General Register A	TGRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 121Ah	MTU3	Timer General Register B	TGRB	16	16	4 or 5 PCLKA	section 20.
000C 121Ch	MTU4	Timer General Register A	TGRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 121Eh	MTU4	Timer General Register B	TGRB	16	16	4 or 5 PCLKA	section 20.
000C 1220h	MTU	Timer Subcounters A	TCNTSA	16	16, 32	4 or 5 PCLKA	section 20.
000C 1222h	MTU	Timer Cycle Buffer Register A	TCBRA	16	16	4 or 5 PCLKA	section 20.
000C 1224h	MTU3	Timer General Register C	TGRC	16	16, 32	4 or 5 PCLKA	section 20.
000C 1226h	MTU3	Timer General Register D	TGRD	16	16	4 or 5 PCLKA	section 20.
000C 1228h	MTU4	Timer General Register C	TGRC	16	16, 32	4 or 5 PCLKA	section 20.
000C 122Ah	MTU4	Timer General Register D	TGRD	16	16	4 or 5 PCLKA	section 20.
000C 122Ch	MTU3	Timer Status Register	TSR	8	8, 16	4 or 5 PCLKA	section 20.
000C 122Dh	MTU4	Timer Status Register	TSR	8	8	4 or 5 PCLKA	section 20.
000C 1230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8, 16	4 or 5 PCLKA	section 20.
000C 1231h	MTU	Timer Interrupt Skipping Counters 1A	TITCNT1A	8	8	4 or 5 PCLKA	section 20.
000C 1232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	4 or 5 PCLKA	section 20.

Table 5.1 List of I/O Registers (Address Order) (14 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
000C 1234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	4 or 5 PCLKA	section 20.
000C 1236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	4 or 5 PCLKA	section 20.
000C 1238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8, 16	4 or 5 PCLKA	section 20.
000C 1239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 or 5 PCLKA	section 20.
000C 123Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	4 or 5 PCLKA	section 20.
000C 123Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	4 or 5 PCLKA	section 20.
000C 123Ch	MTU	Timer Interrupt Skipping Counters 2A	TITCNT2A	8	8	4 or 5 PCLKA	section 20.
000C 1240h	MTU4	Timer A/D Converter Start Request Control Register	TADCR	16	16	4 or 5 PCLKA	section 20.
000C 1244h	MTU4	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16, 32	4 or 5 PCLKA	section 20.
000C 1246h	MTU4	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	4 or 5 PCLKA	section 20.
000C 1248h	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 124Ah	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4 or 5 PCLKA	section 20.
000C 124Ch	MTU3	Timer Control Register 2	TCR2	8	8	4 or 5 PCLKA	section 20.
000C 124Dh	MTU4	Timer Control Register 2	TCR2	8	8	4 or 5 PCLKA	section 20.
000C 1260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	4 or 5 PCLKA	section 20.
000C 1270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	4 or 5 PCLKA	section 20.
000C 1272h	MTU3	Timer General Register E	TGRE	16	16	4 or 5 PCLKA	section 20.
000C 1274h	MTU4	Timer General Register E	TGRE	16	16	4 or 5 PCLKA	section 20.
000C 1276h	MTU4	Timer General Register F	TGRF	16	16	4 or 5 PCLKA	section 20.
000C 1280h	MTU	Timer Start Register A	TSTRA	8	8, 16	4 or 5 PCLKA	section 20.
000C 1281h	MTU	Timer Synchronous Register A	TSYRA	8	8	4 or 5 PCLKA	section 20.
000C 1282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	4 or 5 PCLKA	section 20.
000C 1284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	4 or 5 PCLKA	section 20.
000C 1290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	4 or 5 PCLKA	section 20.
000C 1291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	4 or 5 PCLKA	section 20.
000C 1292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	4 or 5 PCLKA	section 20.
000C 1293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	4 or 5 PCLKA	section 20.
000C 1294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	4 or 5 PCLKA	section 20.
000C 1299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	4 or 5 PCLKA	section 20.
000C 1300h	MTU0	Timer Control Register	TCR	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1301h	MTU0	Timer Mode Register 1	TMDR1	8	8	4 or 5 PCLKA	section 20.
000C 1302h	MTU0	Timer I/O Control Register H	TIORH	8	8, 16	4 or 5 PCLKA	section 20.
000C 1303h	MTU0	Timer I/O Control Register L	TIORL	8	8	4 or 5 PCLKA	section 20.
000C 1304h	MTU0	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1306h	MTU0	Timer Counter	TCNT	16	16	4 or 5 PCLKA	section 20.
000C 1308h	MTU0	Timer General Register A	TGRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 130Ah	MTU0	Timer General Register B	TGRB	16	16	4 or 5 PCLKA	section 20.
000C 130Ch	MTU0	Timer General Register C	TGRC	16	16, 32	4 or 5 PCLKA	section 20.
000C 130Eh	MTU0	Timer General Register D	TGRD	16	16	4 or 5 PCLKA	section 20.
000C 1320h	MTU0	Timer General Register E	TGRE	16	16, 32	4 or 5 PCLKA	section 20.
000C 1322h	MTU0	Timer General Register F	TGRF	16	16	4 or 5 PCLKA	section 20.
000C 1324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8, 16	4 or 5 PCLKA	section 20.
000C 1326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 or 5 PCLKA	section 20.
000C 1328h	MTU0	Timer Control Register 2	TCR2	8	8	4 or 5 PCLKA	section 20.
000C 1380h	MTU1	Timer Control Register	TCR	8	8, 16	4 or 5 PCLKA	section 20.
000C 1381h	MTU1	Timer Mode Register 1	TMDR1	8	8	4 or 5 PCLKA	section 20.
000C 1382h	MTU1	Timer I/O Control Register	TIOR	8	8	4 or 5 PCLKA	section 20.
000C 1384h	MTU1	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1385h	MTU1	Timer Status Register	TSR	8	8	4 or 5 PCLKA	section 20.
000C 1386h	MTU1	Timer Counter	TCNT	16	16	4 or 5 PCLKA	section 20.

Table 5.1 List of I/O Registers (Address Order) (15 / 16)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
000C 1388h	MTU1	Timer General Register A	TGRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 138Ah	MTU1	Timer General Register B	TGRB	16	16	4 or 5 PCLKA	section 20.
000C 1390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	4 or 5 PCLKA	section 20.
000C 1391h	MTU1	Timer Mode Register 3	TMDR3	8	8	4 or 5 PCLKA	section 20.
000C 1394h	MTU1	Timer Control Register 2	TCR2	8	8	4 or 5 PCLKA	section 20.
000C 13A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	4 or 5 PCLKA	section 20.
000C 13A4h	MTU1	Timer Longword General Register	TGRALW	32	32	4 or 5 PCLKA	section 20.
000C 13A8h	MTU1	Timer Longword General Register	TGRBLW	32	32	4 or 5 PCLKA	section 20.
000C 1400h	MTU2	Timer Control Register	TCR	8	8, 16	4 or 5 PCLKA	section 20.
000C 1401h	MTU2	Timer Mode Register 1	TMDR1	8	8	4 or 5 PCLKA	section 20.
000C 1402h	MTU2	Timer I/O Control Register	TIOR	8	8	4 or 5 PCLKA	section 20.
000C 1404h	MTU2	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5 PCLKA	section 20.
000C 1405h	MTU2	Timer Status Register	TSR	8	8	4 or 5 PCLKA	section 20.
000C 1406h	MTU2	Timer Counter	TCNT	16	16	4 or 5 PCLKA	section 20.
000C 1408h	MTU2	Timer General Register A	TGRA	16	16, 32	4 or 5 PCLKA	section 20.
000C 140Ah	MTU2	Timer General Register B	TGRB	16	16	4 or 5 PCLKA	section 20.
000C 140Ch	MTU2	Timer Control Register 2	TCR2	8	8	4 or 5 PCLKA	section 20.
000C 1480h	MTU5	Timer Counter U	TCNTU	16	16, 32	4 or 5 PCLKA	section 20.
000C 1482h	MTU5	Timer General Register U	TGRU	16	16	4 or 5 PCLKA	section 20.
000C 1484h	MTU5	Timer Control Register U	TCRU	8	8	4 or 5 PCLKA	section 20.
000C 1485h	MTU5	Timer Control Register 2U	TCR2U	8	8	4 or 5 PCLKA	section 20.
000C 1486h	MTU5	Timer I/O Control Register U	TIORU	8	8	4 or 5 PCLKA	section 20.
000C 1490h	MTU5	Timer Counter V	TCNTV	16	16, 32	4 or 5 PCLKA	section 20.
000C 1492h	MTU5	Timer General Register V	TGRV	16	16	4 or 5 PCLKA	section 20.
000C 1494h	MTU5	Timer Control Register V	TCRV	8	8	4 or 5 PCLKA	section 20.
000C 1495h	MTU5	Timer Control Register 2V	TCR2V	8	8	4 or 5 PCLKA	section 20.
000C 1496h	MTU5	Timer I/O Control Register V	TIORV	8	8	4 or 5 PCLKA	section 20.
000C 14A0h	MTU5	Timer Counter W	TCNTW	16	16, 32	4 or 5 PCLKA	section 20.
000C 14A2h	MTU5	Timer General Register W	TGRW	16	16	4 or 5 PCLKA	section 20.
000C 14A4h	MTU5	Timer Control Register W	TCRW	8	8	4 or 5 PCLKA	section 20.
000C 14A5h	MTU5	Timer Control Register 2W	TCR2W	8	8	4 or 5 PCLKA	section 20.
000C 14A6h	MTU5	Timer I/O Control Register W	TIORW	8	8	4 or 5 PCLKA	section 20.
000C 14B2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	4 or 5 PCLKA	section 20.
000C 14B4h	MTU5	Timer Start Register	TSTR	8	8	4 or 5 PCLKA	section 20.
000C 14B6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	4 or 5 PCLKA	section 20.
000C 1D30h	MTU	A/D Conversion Start Request Select Register 0	TADSTRGR0	8	8	4 or 5 PCLKA	section 20.
007F C100h	FLASH	Flash P/E Mode Control Register	FPMCR	8	8	2 or 3 FCLK	section 34.
007F C104h	FLASH	Flash Area Select Register	FASR	8	8	2 or 3 FCLK	section 34.
007F C108h	FLASH	Flash Processing Start Address Register L	FSARL	16	16	2 or 3 FCLK	section 34.
007F C110h	FLASH	Flash Processing Start Address Register H	FSARH	16	16	2 or 3 FCLK	section 34.
007F C114h	FLASH	Flash Control Register	FCR	8	8	2 or 3 FCLK	section 34.
007F C118h	FLASH	Flash Processing End Address Register L	FEARL	16	16	2 or 3 FCLK	section 34.
007F C120h	FLASH	Flash Processing End Address Register H	FEARH	16	16	2 or 3 FCLK	section 34.
007F C124h	FLASH	Flash Reset Register	FRESETR	8	8	2 or 3 FCLK	section 34.
007F C12Ch	FLASH	Flash Status Register 1	FSTATR1	8	8	2 or 3 FCLK	section 34.
007F C130h	FLASH	Flash Write Buffer Register 0	FWB0	16	16	2 or 3 FCLK	section 34.
007F C138h	FLASH	Flash Write Buffer Register 1	FWB1	16	16	2 or 3 FCLK	section 34.
007F C140h	FLASH	Flash Write Buffer Register 2	FWB2	16	16	2 or 3 FCLK	section 34.
007F C144h	FLASH	Flash Write Buffer Register 3	FWB3	16	16	2 or 3 FCLK	section 34.
007F C180h	FLASH	Protection Unlock Register	FPR	8	8	2 or 3 FCLK	section 34.
007F C184h	FLASH	Protection Unlock Status Register	FPSR	8	8	2 or 3 FCLK	section 34.

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles ICLK ≥ PCLK	Reference Section
007F C1C0h	FLASH	Flash Start-Up Setting Monitor Register	FSCMR	16	16	2 or 3 FCLK	section 34.
007F C1C8h	FLASH	Flash Access Window Start Address Monitor Register	FAWSMR	16	16	2 or 3 FCLK	section 34.
007F C1D0h	FLASH	Flash Access Window End Address Monitor Register	FAWEMR	16	16	2 or 3 FCLK	section 34.
007F C1D8h	FLASH	Flash Initial Setting Register	FISR	8	8	2 or 3 FCLK	section 34.
007F C1DCh	FLASH	Flash Extra Area Control Register	FEXCR	8	8	2 or 3 FCLK	section 34.
007F C1E0h	FLASH	Flash Error Address Monitor Register L	FEAML	16	16	2 or 3 FCLK	section 34.
007F C1E8h	FLASH	Flash Error Address Monitor Register H	FEAMH	16	16	2 or 3 FCLK	section 34.
007F C1F0h	FLASH	Flash Status Register 0	FSTATR0	8	8	2 or 3 FCLK	section 34.
007F C350h	FLASHCONST	Unique ID Register 0	UIDR0	32	32	2 or 3 FCLK	section 34.
007F C354h	FLASHCONST	Unique ID Register 1	UIDR1	32	32	2 or 3 FCLK	section 34.
007F C358h	FLASHCONST	Unique ID Register 2	UIDR2	32	32	2 or 3 FCLK	section 34.
007F C35Ch	FLASHCONST	Unique ID Register 3	UIDR3	32	32	2 or 3 FCLK	section 34.
007F FFB2h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 or 3 FCLK	section 34.

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 22.4 lists register allocation for 16-bit access.

6. Resets

6.1 Overview

The following resets are implemented: 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 (LVDAb) and section 35, 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.LVD1E, LVDLVLR.LVD1LVL[3:0])	○	○	○	○	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	—	—	—
The voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	—	—
Registers related to voltage monitor function 2 (LVD2CR0, LVD2E, LVDLVLR.LVD2LVL[1:0])	○	○	○	○	○	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	—	—
The software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	—
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○

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

Note 1. Initialized at a power-on.

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.

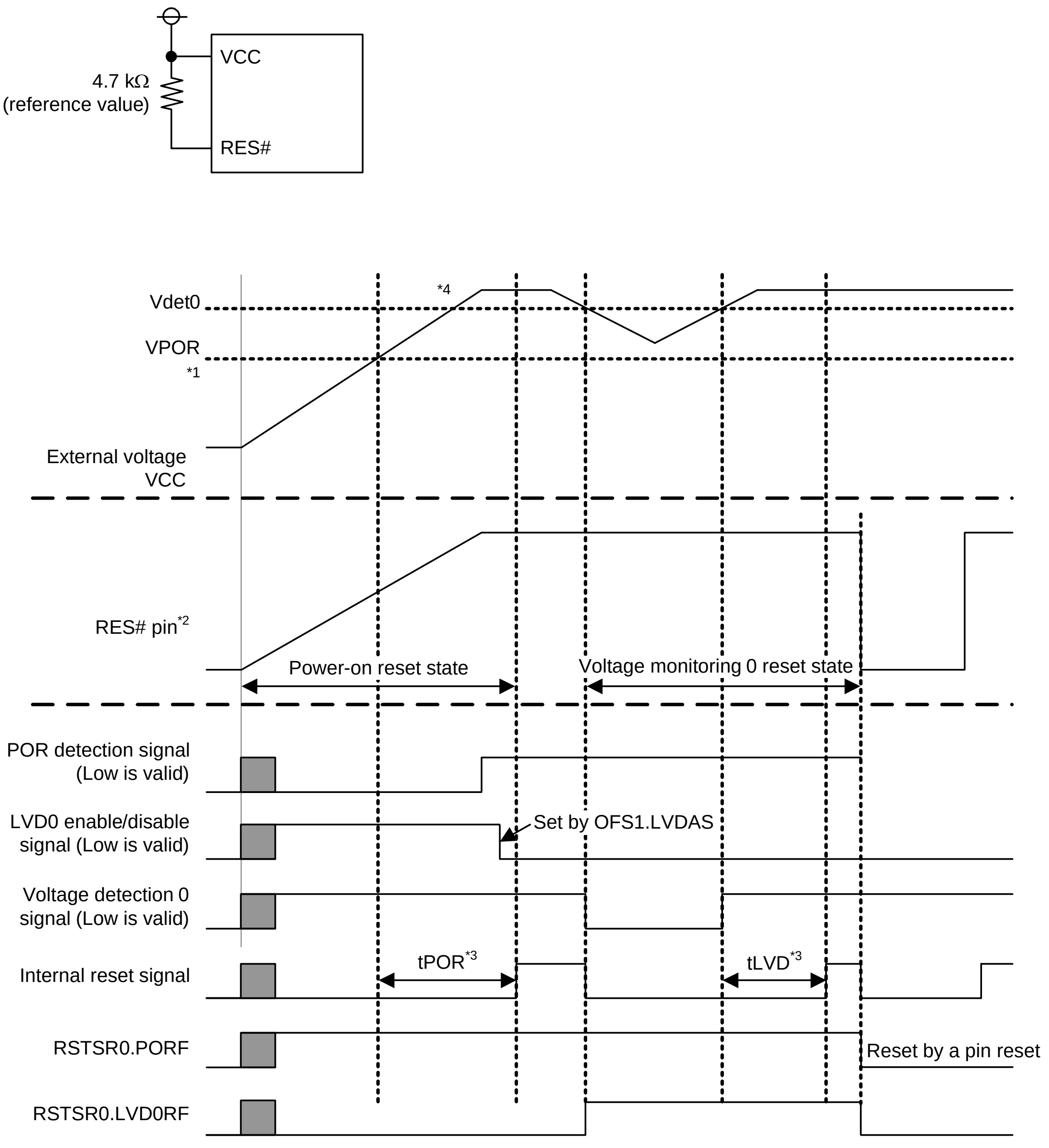
6.3 Operation

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 35, 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. A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least VIH. For details on VIH, refer to section 35, Electrical Characteristics. 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 for the external power supply and the MCU circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by 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 (LVDAb).



Note: For details on the electrical characteristics, see the Electrical Characteristics section.

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. Ensure that the voltage on the RES# pin is always at least VIH.

Note 3. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.

Note 4. 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.

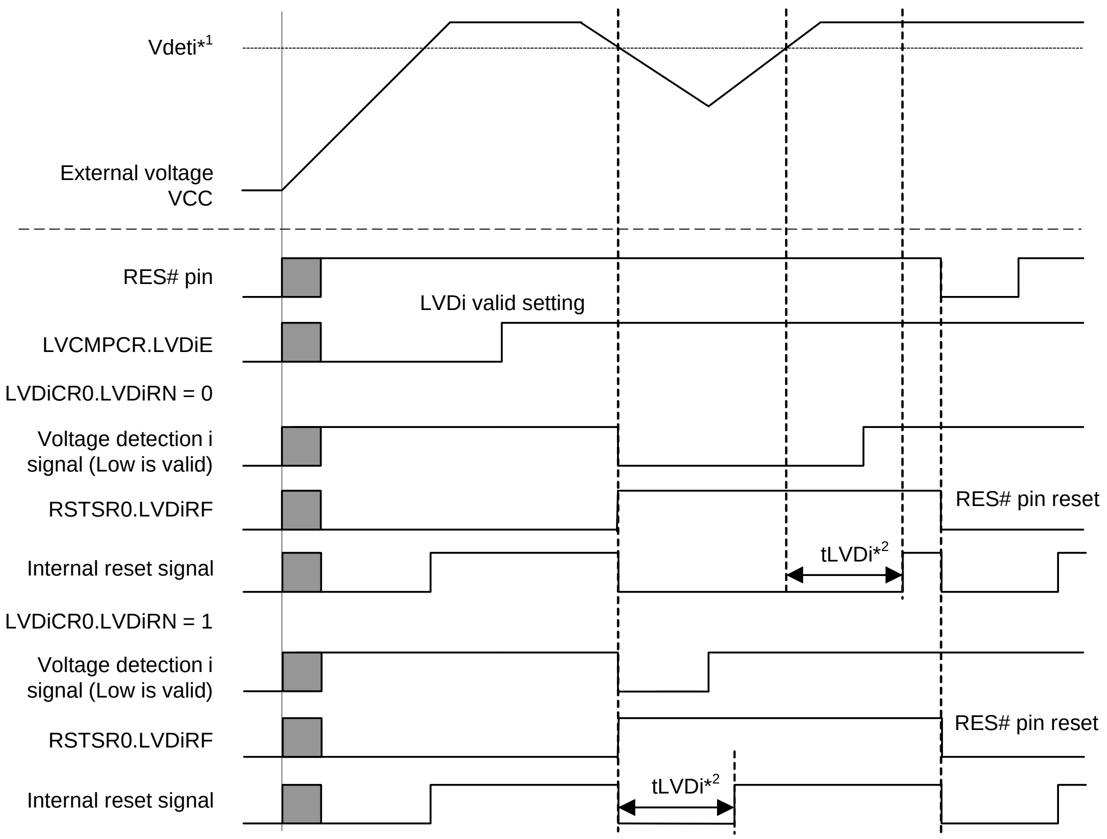
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of 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 (LVDAb).



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

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

6.3.4 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by 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 24, 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.

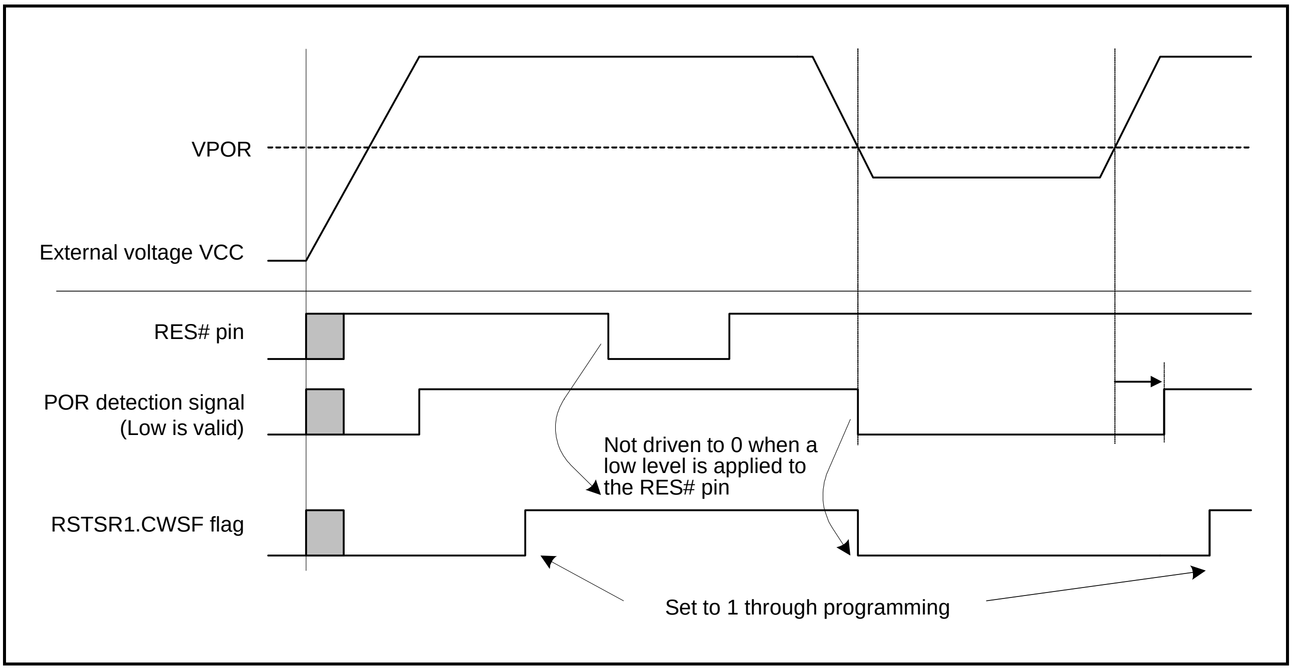


Figure 6.3 Example of Cold/Warm Start Determination Operation

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.

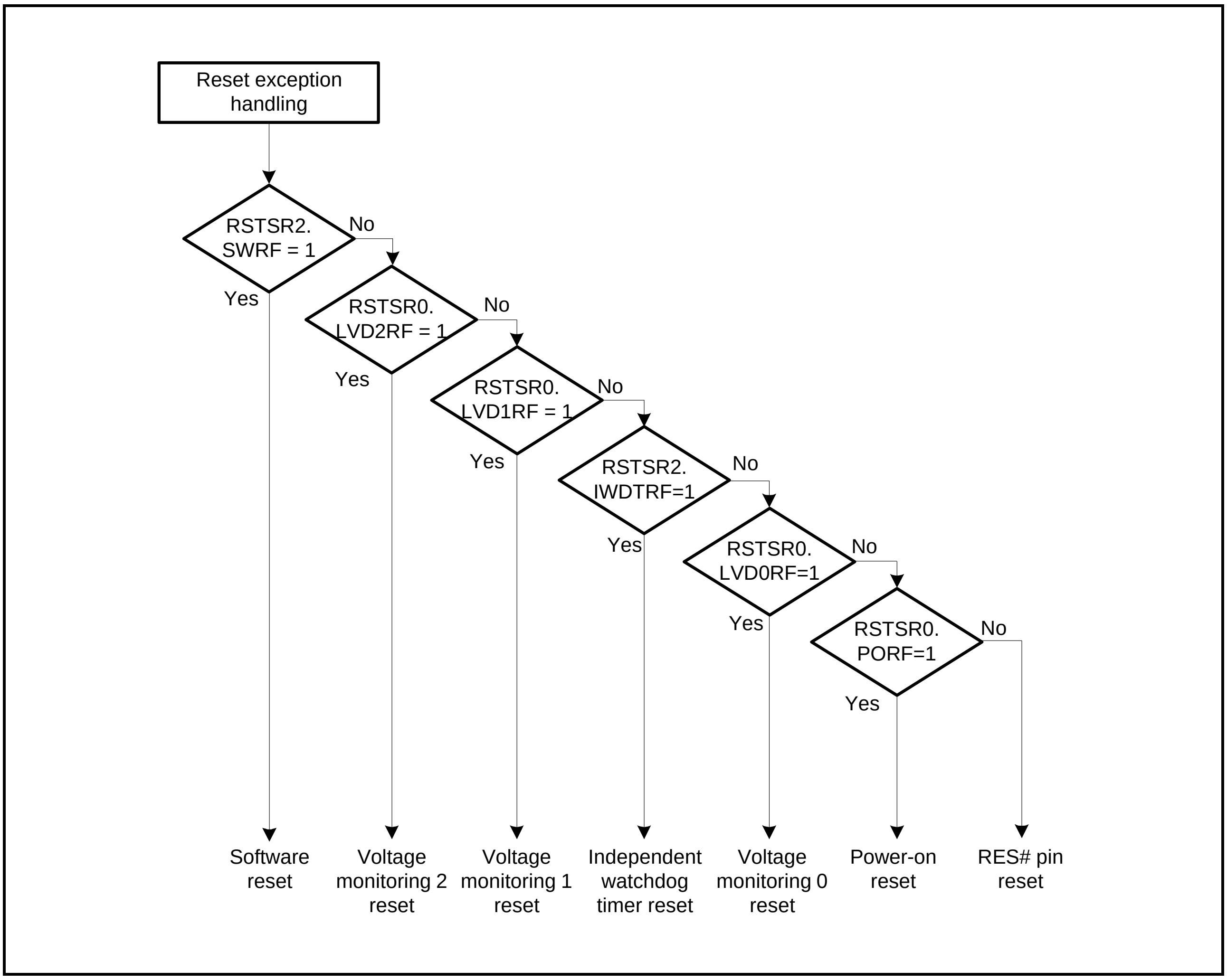


Figure 6.4 Example of Reset Generation Source Determination Flow

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.

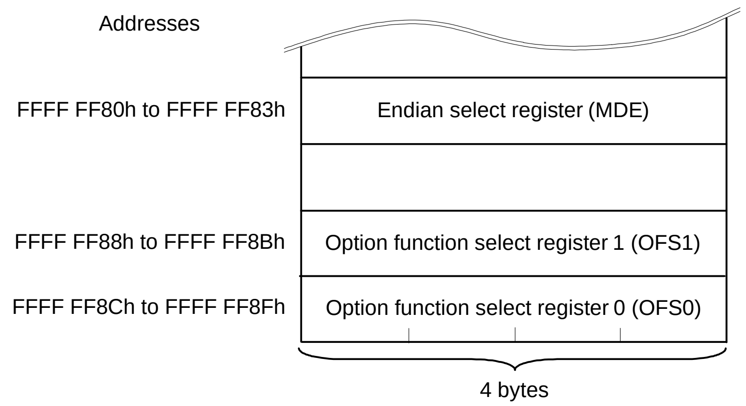


Figure 7.1 Option-Setting Memory Area

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															

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: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request 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 deep sleep mode	R
b31 to b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

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

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

The setting in the OFS0 register is ignored in boot mode, and this register functions similarly when it is set to 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 128, 512, 1024, or 2048 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 24, 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 128 to 524288 IWDT-dedicated clock cycles.
For details, see section 24, 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 24, 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 24, 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 24, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects whether to stop counting when entering sleep, software standby, or deep sleep mode.
For details, see section 24, 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
—	—	—	—	—	—	—	HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: 3.84 V is selected 1 0: 2.51 V is selected Settings other than above are prohibited when the voltage detection 0 circuit is used.	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

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

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ignored in boot mode, and this register functions similarly when it is set to 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 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.
Also, when the HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

7.2.3 Endian Select Register (MDE)

Address(es): FFFF FF80h

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

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

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

The MDE register selects the endian for the CPU.
MDE 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 MDE register, the MDE register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

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

7.3 Usage Note

7.3.1 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 0fffffff8h

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

8. Voltage Detection Circuit (LVDAb)

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 two levels using option function select register 1 (OFS1).
In voltage detection 1, the detection voltage can be selected from nine levels using the voltage detection level select register (LVDLVLR).
In voltage detection 2, the detection voltage can be selected from four levels using the LVDLVLR register.
Voltage monitoring 0 reset, voltage monitoring 1 reset/interrupt, and voltage monitoring 2 reset/interrupt can be used.
Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit. Figure 8.3 is a block diagram of the voltage monitoring 2 interrupt/reset circuit.

Table 8.1 LVD Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detection target	Voltage drops past Vdet0	When voltage rises above or drops below Vdet1	When voltage rises above or drops below Vdet2
	Detection voltage	Voltage selectable from 2 levels using OFS1	Voltage selectable from 9 levels using the LVDLVLR.LVD1LVL[3:0] bits	Voltage selectable from four levels using the LVDLVLR.LVD2LVL[1:0] bits
	Monitoring flag	Not available	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 after specified time with Vdet2 > VCC
	Interrupt	Not available	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

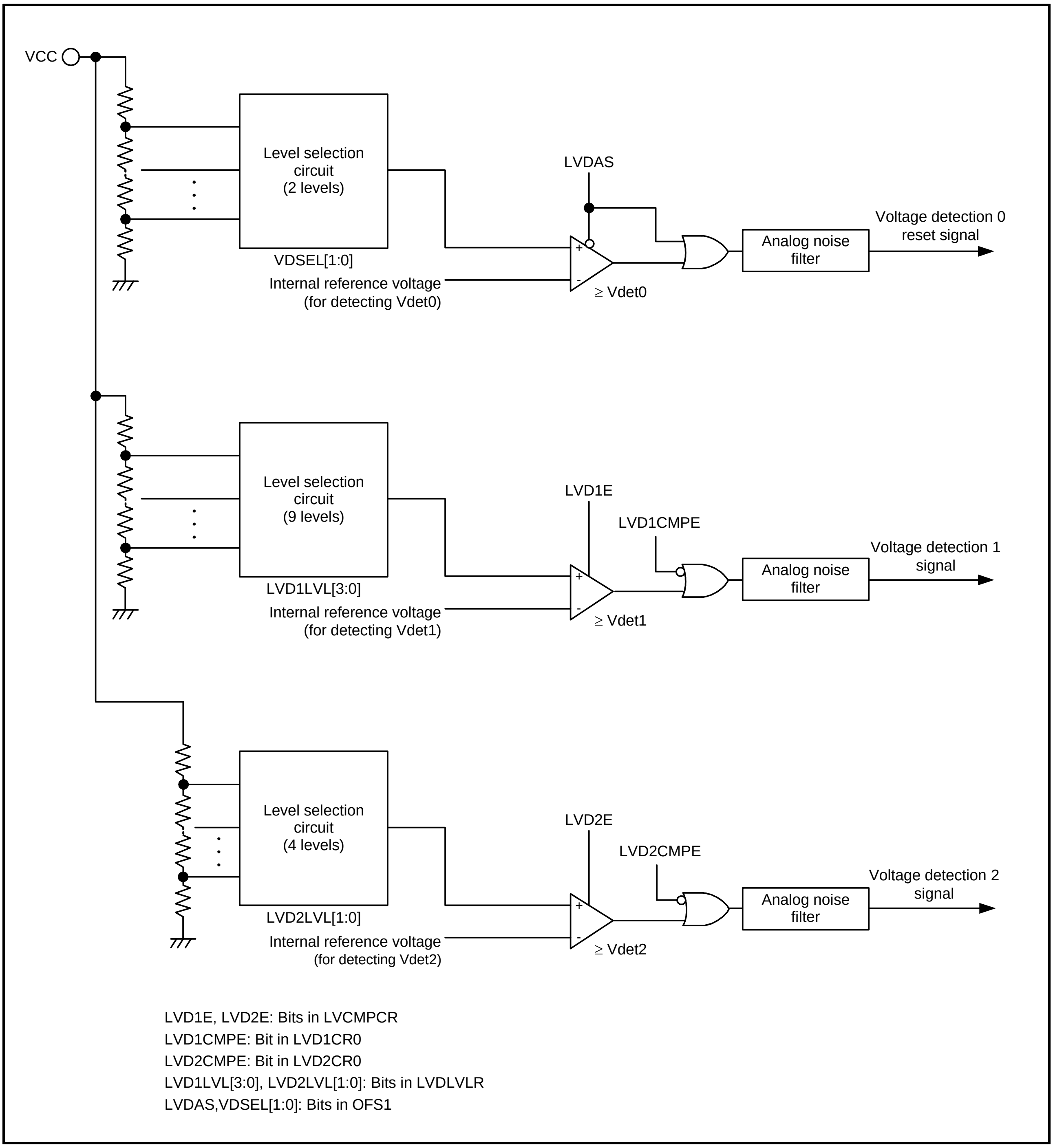


Figure 8.1 Block Diagram of the LVD

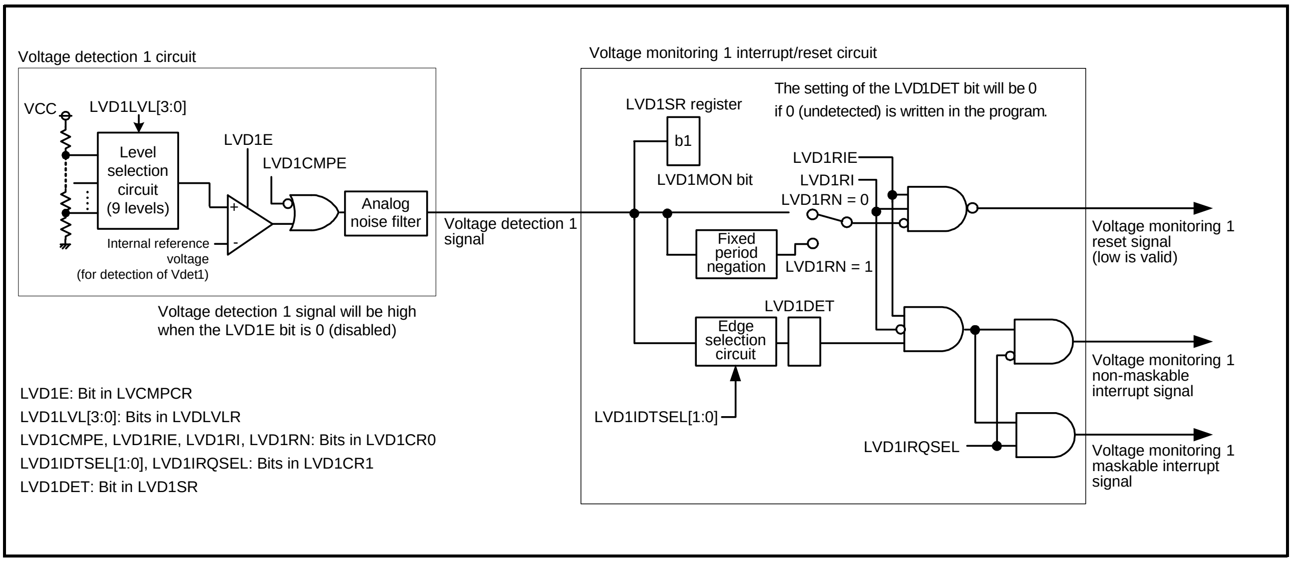


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

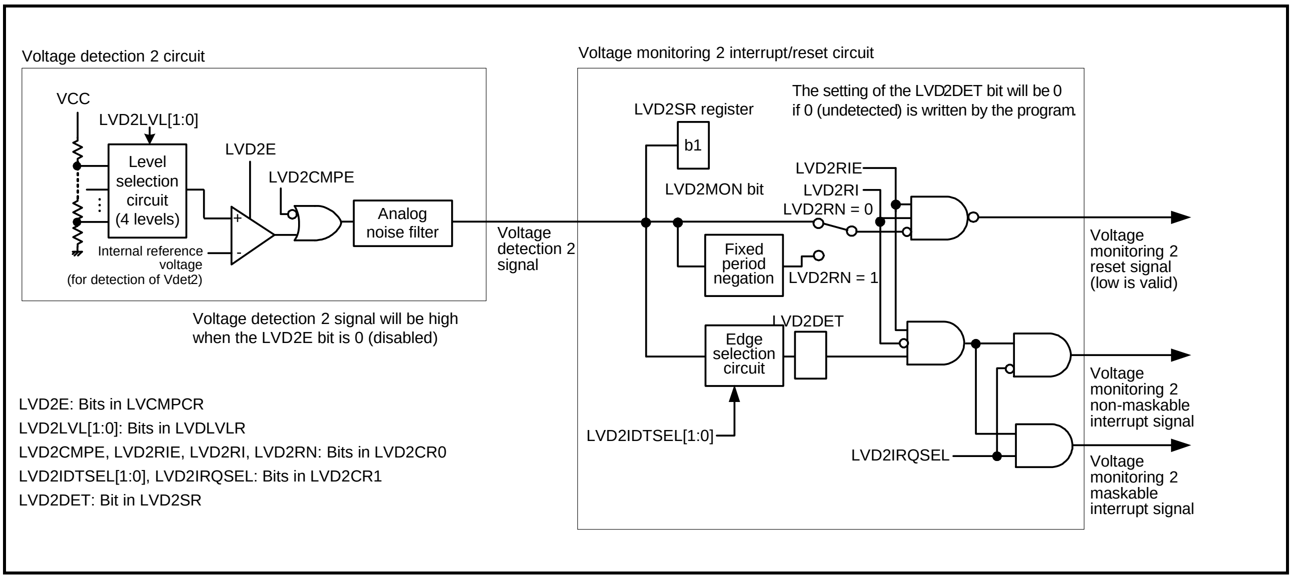


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL[1:0]	Voltage Monitoring 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 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 Status Register (LVD1SR)

Address(es): 0008 00E1h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1DET	Voltage Monitoring 1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R/(W) *1
b1	LVD1MON	Voltage Monitoring 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 or LVD1MON circuit 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 two system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

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

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

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

8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL[1:0]	Voltage Monitoring 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet2 (rise) is detected 0 1: When VCC < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 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 Status Register (LVD2SR)

Address(es): 0008 00E3h

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

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

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

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

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

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

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

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

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

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

LVD1E Bit (Voltage Detection 1 Enable)

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

LVD2E Bit (Voltage Detection 2 Enable)

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

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

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

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: 4.29 V 0 0 0 1: 4.14 V 0 0 1 0: 4.02 V 0 0 1 1: 3.84 V 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.79 V 1 0 0 0: 2.68 V Settings other than those listed above are prohibited.	R/W
b5, b4	LVD2LVL[1:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b5 b4 0 0: 4.29V 0 1: 4.14V 1 0: 4.02V 1 1: 3.84V	R/W
b7, b6	—	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.

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

When a setting is made so that the voltage detection level range set by the LVD1LVL[3:0] bits overlaps with the range set by the LVD2LVL[1:0] bits, it cannot be specified which of LVD1 and LVD2 is used for voltage detection. For details on the voltage detection level range, refer to section 35, Electrical Characteristics.

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	—	—	—	LVD1CMPE	—	LVD1RIE
Value after reset:	1	0	0	0	X	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1RIE	Voltage Monitoring 1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison results output disabled 1: Voltage monitoring 1 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt occurs when the voltage passes Vdet1 1: Voltage monitoring 1 reset occurs when the voltage falls below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 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 voltage monitoring 1 reset.	R/W

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

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (voltage detection 1 circuit enabled) and the LVD1CMPE bit is set to 1 (voltage monitoring 1 circuit comparison results output enabled).
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.

LVD1RN Bit (Voltage Monitoring 1 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), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, 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).

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	—	—	—	LVD2CMPE	—	LVD2RIE
Value after reset:	1	0	0	0	X	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison results output disabled 1: Voltage monitoring 2 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2 Reset 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 voltage monitoring 2 reset.	R/W

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

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

The LVD2RIE bit is enabled when the LVCMPCR.LVD2E bit is set to 1 (voltage detection 2 circuit enabled) and the LVD2CMPE bit is set to 1 (voltage monitoring 2 circuit comparison results output enabled).
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.

LVD2RN Bit (Voltage Monitoring 2 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), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, 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).

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

(1) Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).
(2) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).
(3) After waiting for td(E-A), set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).

8.3.3 Monitoring Vdet2

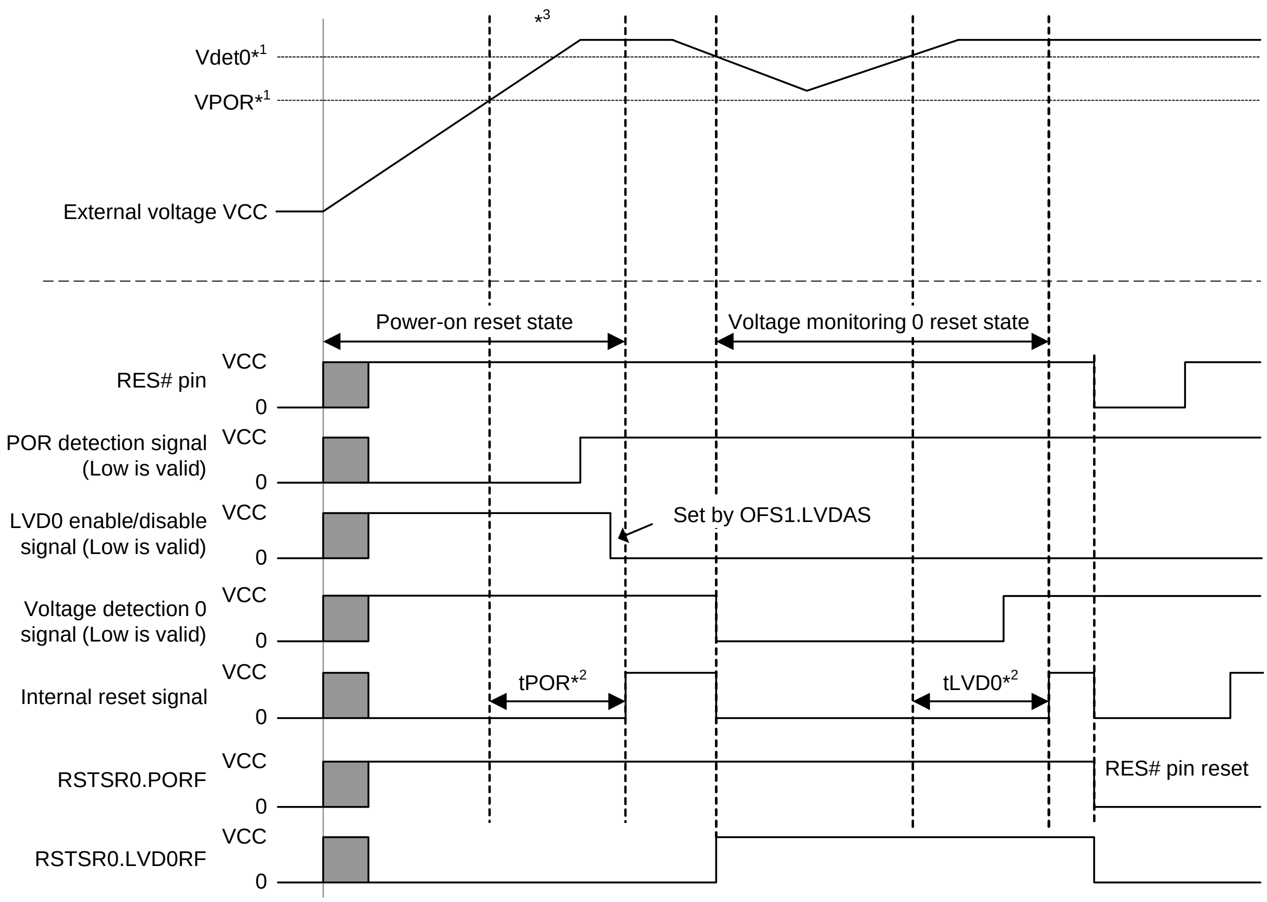
After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

(1) Specify the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits (voltage detection 2 level).
(2) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).
(3) After waiting for td(E-A), set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).

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 the Electrical Characteristics section.

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 8.2 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.3 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.

Table 8.2 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt	Voltage Monitoring 1 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
2*1	Set the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
3	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.	—
4	—	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).	
6*1	Wait for at least td(E-A).	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).	
8	Set the LVD1SR.LVD1DET bit to 0.	—
9	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).	—

Note 1. Steps 1, 2, 5, and 6 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 LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] 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 9.

Table 8.3 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt	Voltage Monitoring 1 Reset
1	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).	—
2	Set the LVD1CR0.LVD1CMPE bit to 0 (voltage monitoring 1 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled).	
4	—	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).
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 LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] 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.

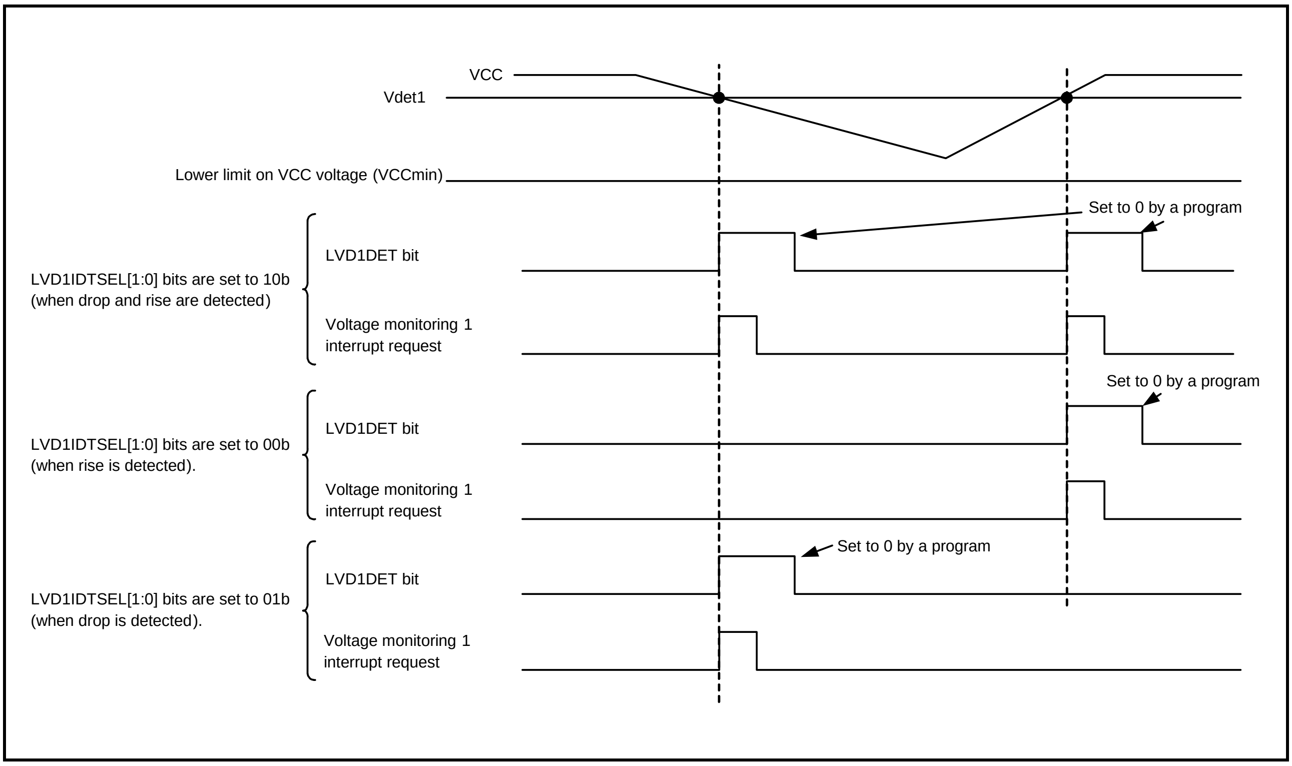


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitoring 2

Table 8.4 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Table 8.5 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.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset

Step	Voltage Monitoring 2 Interrupt	Voltage Monitoring 2 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits.	
2*1	Set the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.
3	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.	—
4	—	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).	
6*1	Wait for at least td(E-A).	
7	Set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).	
8	Set the LVD2SR.LVD2DET bit to 0.	—
9	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled)	—

Note 1. Steps 1, 2, 5, and 6 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 LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] 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 9.

Table 8.5 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	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).	—
2	Set the LVD2CR0.LVD2CMPE bit to 0 (voltage monitoring 2 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD2E bit to 0 (voltage monitoring 2 circuit disabled).	
4	—	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).
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 LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] 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.

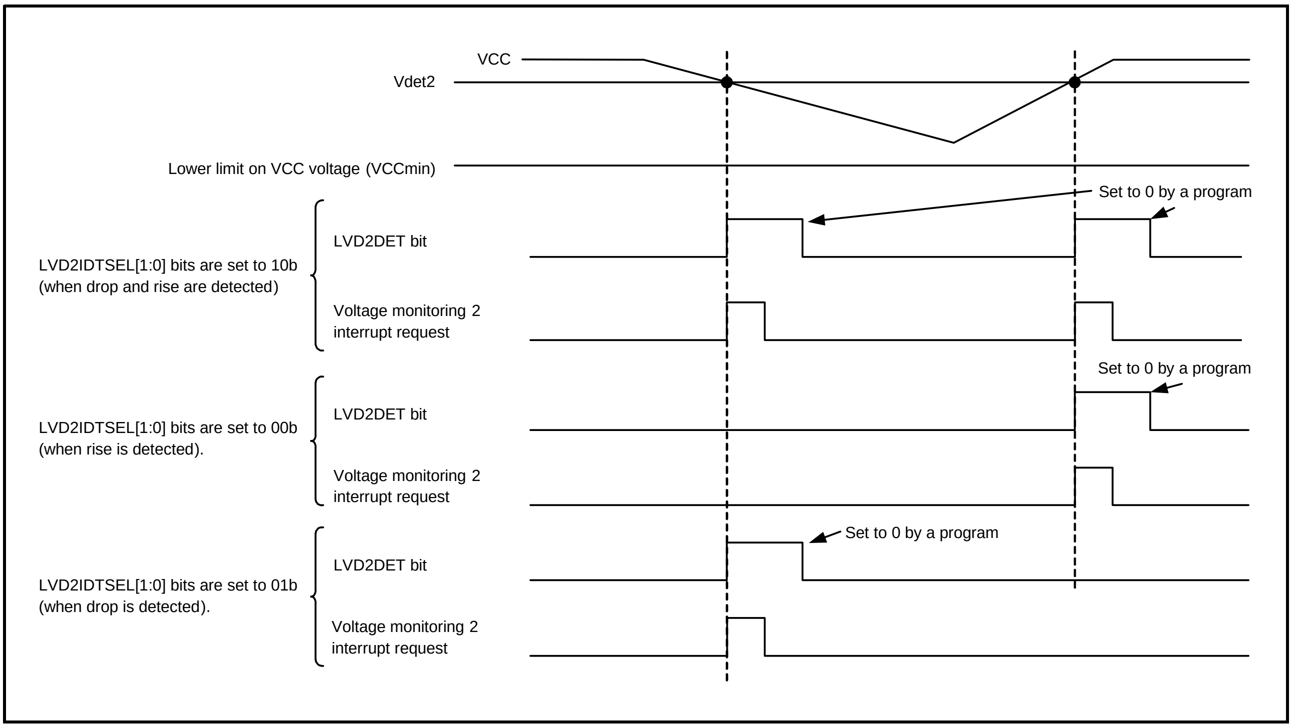


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.

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

Table 9.1 Specifications of Clock Generation Circuit

Item	Specification
Uses	• Generates the system clock (ICLK) to be supplied to the CPU, DTC, ROM, and RAM. • Generates the peripheral module clocks (PCLKA, PCLKB, and PCLKD) to be supplied to peripheral modules. The peripheral module clock PCLKA is the operating clock for the MTU3, the peripheral module clock PCLKD is for the S12AD, and PCLKB is 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 IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT.
Operating frequencies*1	• ICLK: 40 MHz (max) • PCLKA: 40 MHz (max) • PCLKB: 40 MHz (max) • PCLKD: 40 MHz (max) • FCLK: 1 to 32 MHz (ROM) • CACCLK: Same frequency as each oscillator • IWDTCLK: 15 kHz
Main clock oscillator	• Resonator frequency: 1 to 20 MHz • External clock input frequency: 20 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal • Connection pins: EXTAL, XTAL • Oscillation stop detection function: When a main clock oscillation stop is detected, the system clock source is switched to LOCO and MTU output can be forcedly driven to high-impedance. • Drive capacity switching function
PLL circuit	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 to 12.5 MHz • Frequency multiplication ratio: Selectable from 4 to 10 (increments of 0.5) • VCO oscillation frequency: 24 to 40MHz
High-speed on-chip oscillator (HOCO)	Oscillation frequency: 32 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 4 MHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 15 kHz

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

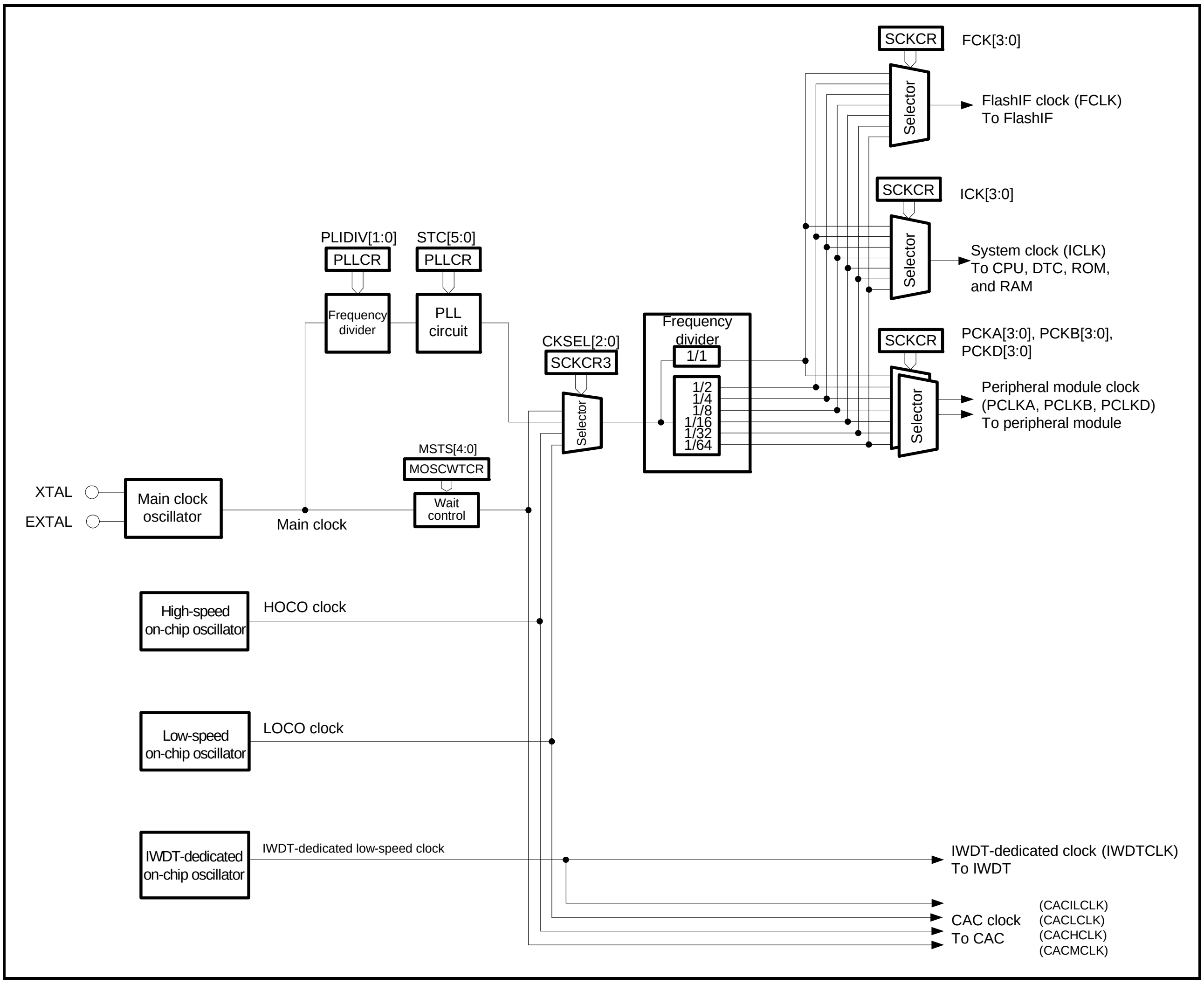


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the I/O pins of the clock generation circuit.

Table 9.2 I/O Pins of Clock Generation Circuit

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

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0] *2	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 are read as 0. The write value should be 0.	R/W
b11 to b8	PCKB[3:0] *2	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	PCKA[3:0] *2	Peripheral Module Clock A (PCLKA) Select	b15 b12 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
b19 to b16	—	Reserved	Set the same value as the set value of the ICK[3:0] or PCKB[3:0] bits, whichever division ratio is greater.	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] *1	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]*2	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.
Note 1. Do not set the frequency division ratio of ICLK = 1 when a clock of frequency higher than 32 MHz is selected by the SCKCR3.CKSEL[2:0] bits and MEMWAIT.MEMWAIT = 0.
Note 2. Do not set the frequency higher than the system clock (ICLK).

This register cannot be rewritten while the flash memory is being programmed or erased.
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.

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

These bits select the frequency of peripheral module clock D (PCLKD).

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

These bits select the frequency of peripheral module clock B (PCLKB).

PCKA[3:0] Bits (Peripheral Module Clock A (PCLKA) Select)

These bits select the frequency of peripheral module clock A (PCLKA).

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Do not select a clock of frequency higher than 32 MHz when divided by 1 is selected by SCKCR.ICK MEMWAIT.MEMWAIT = 0.

This register cannot be rewritten while the flash memory is being programmed or erased.

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

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

9.2.3 PLL Control Register (PLLCR)

Address(es): 0008 0028h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 0 0: ×4.5 0 0 1 0 0 1: ×5 0 0 1 0 1 0: ×5.5 0 0 1 0 1 1: ×6 0 0 1 1 0 0: ×6.5 0 0 1 1 0 1: ×7 0 0 1 1 1 0: ×7.5 0 0 1 1 1 1: ×8 0 1 0 0 0 0: ×8.5 0 1 0 0 0 1: ×9 0 1 0 0 1 0: ×9.5 0 1 0 0 1 1: ×10 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

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

These bits select the frequency division ratio of the PLL clock source.
Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 12.5 MHz.

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

These bits select the frequency multiplication factor of the PLL circuit.
Set these bits so that the PLL oscillation frequency is within the range of 24 MHz to 40 MHz.

9.2.4 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

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

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

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

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

After setting the PLLEN bit to 0 (PLL is operating), confirm that the OSCOVFSR.PLOVF bit is 1 before switching the system clock to the PLL clock.

That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation.

- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCOVFSR.PLOVF bit is 1 before stopping the PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.PLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

When the PLL clock is selected by the SCKCR3.CKSEL[2:0] bits, do not set the PLLEN bit (PLL is stopped) to 1.

9.2.5 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 this register after setting up the main clock oscillator wait control register.

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.
After setting the MOSTP bit to 0 (main clock oscillator is operating), read the OSCOVFSR.MOOVF bit to confirm that is has become 1, and then use the main clock.
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.

- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCOVFSR.MOOVF bit is 1 before stopping the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF bit is 1 and execute a WAIT instruction in order to operate the main clock oscillator and place the MCU in software standby mode.
- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

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

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)
- When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)
- When the PLL is operating (PLLCR2.PLLEN = 0)

9.2.6 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 clock after the LOCO clock oscillation stabilization time (t_{LOCO}) 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.

While the LOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the LCSTP bit to 1 (LOCO is stopped).

9.2.7 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 is operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT is stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator is operating) to 1 (IWDT-dedicated on-chip oscillator is 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 is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time (t_{ILOCO}) has elapsed.

If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait time 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.

9.2.8 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 the high-speed on-chip wait control register before setting this register.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.
When changing the HCSTP bit from 1 to 0 (i.e. changing the HOCO clock from stopped to operating), confirm that the OSCOVFSR.HCOVF bit is 1 before switching the system clock to the HOCO clock.
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.

- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 before restarting the HOCO.
- Confirm that the HOCO is operating and that the OSCOVFSR.HCOVF bit is 1 before stopping the HOCO.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

While the HOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the HCSTP bit to 1 (HOCO is stopped).
While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the HCSTP bit to 0 (HOCO is operating).

9.2.9 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)

Address(es): 0008 00A5h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	HSTS[2:0]	High-Speed On-Chip Oscillator Oscillation Stabilization Wait Time	b2 b0 1 0 0: Wait time = 78 cycles 1 0 1: Wait time = 142 cycles 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.PRC2 bit to 1 (write enabled) before rewriting this register.

The HOCOWTCR register can be rewritten under the following cases. Otherwise, do not rewrite this register.

- When the HOCOCR.HCSTP bit is set to 0 (operating), and the OSCOVFSR.HCOVF flag is read and confirmed to be 1.
- When the HOCOCR.HCSTP bit is set to 1 (stopped), and the OSCOVFSR.HCOVF flag is read and confirmed to be 0.

HSTS[2:0] Bits (High-Speed On-Chip Oscillator Oscillation Stabilization Wait Time)

These bits are used to select the oscillation stabilization wait time of the HOCO when setting HOCO operation (the HOCOCR.HCSTP bit to 0) and when canceling software standby mode.

Supply of the HOCO clock is started to the MCU internally after the number of LOCO cycles set by the HSTS[2:0] bits has been counted. Counting of LOCO cycles proceeds regardless of the setting of the LOCOCR.LOSTP bit and hardware automatically controls running and stopping the LOCO.

The clock is not supplied to the MCU internally until counting is completed.

After counting is completed, supply of the clock is started to the MCU internally and the OSCOVFSR.HCOVF flag is set to 1.

9.2.10 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: Main clock is stopped 1: Oscillation is stable and the clock can be used as the system clock*1	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b3	HCOVF	HOCO Clock Oscillation Stabilization Flag	0: HOCO is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock*1	R
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.
If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the main clock is stable.
[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 (main clock oscillator is operating) when the MOSTP bit is 1 (main clock oscillator is stopped), the corresponding time set in the MOSCWTCR register has elapsed and supply of the main clock is started to the MCU internally.

[Clearing condition]

- After the MOSCCR.MOSTP bit is set to 1, the processing to stop the oscillation of the main clock oscillator is completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL clock is stable.
[Setting condition]
After the PLLCR2.PLLEN is set to 0 (PLL is operating) when the PLLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL clock is started to the MCU internally.
[Clearing condition]
After the PLLCR2.PLLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the HOCO clock is stable.
[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped), supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

- After the HOCOCR.HCSTP bit is set to 1, the processing to stop the oscillation of the HOCO is completed.

9.2.11 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 setting 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.
When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 (LOCO is stopped) to the LOCOCR.LCSTP bit is invalid.
When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop has been 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.

9.2.12 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 and cannot be modified.	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 set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least three 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 set 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) or PLL (100b) 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. The OSTDF flag should be set to 0 after switching the clock source to a source other than the main clock oscillator and the PLL.

[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 nor 100b.

9.2.13 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	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time	b4 b0 0 0 0 0 0: Wait time = 2 cycles (0.5 μs) 0 0 0 0 1: Wait time = 1024 cycles (256 μs) 0 0 0 1 0: Wait time = 2048 cycles (512 μs) 0 0 0 1 1: Wait time = 4096 cycles (1.024 ms) 0 0 1 0 0: Wait time = 8192 cycles (2.048 ms) 0 0 1 0 1: Wait time = 16384 cycles (4.096 ms) 0 0 1 1 0: Wait time = 32768 cycles (8.192 ms) 0 0 1 1 1: Wait time = 65536 cycles (16.384 ms) Settings other than above are prohibited. Wait time when LOCO = 4.0 MHz (0.25 μs, TYP.)	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.

MSTS[4:0] Bits (Main Clock Oscillator Wait Time)

Set these bits to select the oscillation stabilization wait time of the main clock oscillator.

Set the main clock oscillation stabilization time to longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is externally input, set these bits to 00000b because the oscillation stabilization time is not required.

The wait time set by the MSTS[4:0] bits is counted using the LOCO clock. The LOCO automatically oscillates when necessary, regardless of the value of the LOCOCR.LCSTP bit.

After the set wait time has elapsed, supply of the main clock is started to the MCU internally and the OSCOVFSR.MOOVF flag becomes 1. If the set wait time is short, supply of the main clock is started before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCOVFSR.MOOVF flag is 0. Do not rewrite this register under any other conditions.

9.2.14 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODR V21	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	MODRV21	Main Clock Oscillator Drive Capability Switch	0: 1 MHz or higher and lower than 10 MHz 1: 10 MHz to 20 MHz	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.

MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

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.

9.2.15 Memory Wait Cycle Setting Register (MEMWAIT)

Address(es): 0008 0031h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MEMWAIT	Memory Wait Cycle Setting*1	0: No wait states 1: Wait states	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.
Note 1. Do not select the MEMWAIT bit = 0 (no wait states) when divided by 1 is selected by the SCKCR.ICK[3:0] bits and a clock of frequency is higher than 32 MHz is selected as the system clock (ICLK) by the SCKCR3.CKSEL[2:0] bits. When a clock of frequency is lower than 32 MHz is selected as the ICLK, it is not necessary to set the MEMWAIT bit to 1 (wait states).

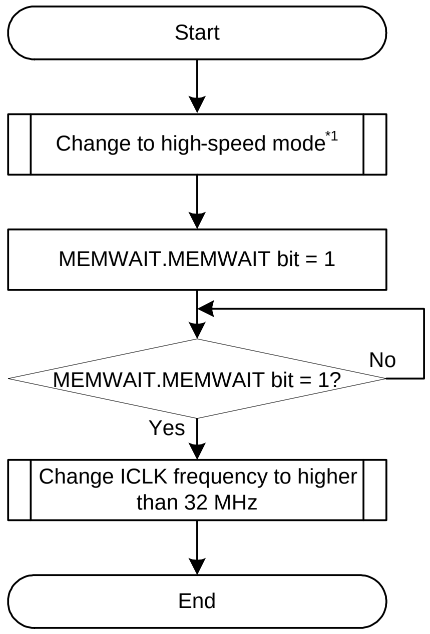
The MEMWAIT register is used to control the wait cycle of the ROM.

MEMWAIT Bit (Memory Wait Cycle Setting)

This bit is used to set the wait cycle of the ROM.
This bit is set to “no wait states” immediately after a reset.
When selecting a clock of frequency higher than 32 MHz as the system clock (ICLK), set the bit to 1 (wait sates).
When setting the MEMWAIT bit to 1 (wait states), make sure that high-speed mode is selected. After the value of the MEMWAIT bit is changed to 1, change the system clock to a clock of frequency higher than 32 MHz.
When setting the MEMWAIT bit to 0 (no wait states), make sure that the frequency of the system clock (ICLK) is 32 MHz or lower. When changing the operating power control state, make sure that the value of the MEMWAIT bit is changed to 0.
Table 9.3 lists the restrictions on setting the MEMWAIT bit, and Figure 9.2 and Figure 9.3 show the procedure for changing the MEMWAIT bit.

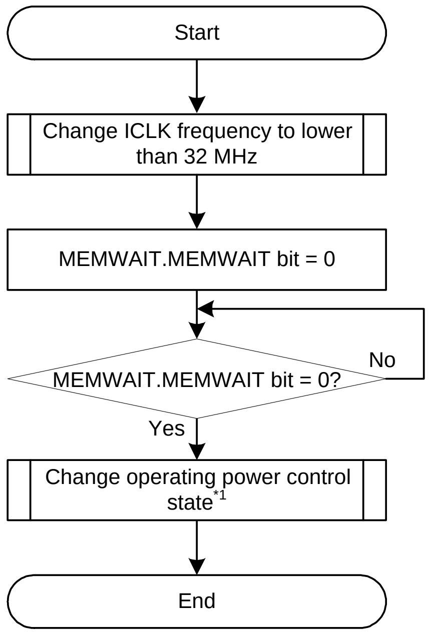
Table 9.3 Restrictions on Setting the MEMWAIT Bit

MEMWAIT Bit	Operating Power Control State		
	High-Speed Operating Mode		Middle-Speed Operating Mode
	ICLK ≤ 32 MHz	ICLK > 32 MHz	
0	Can be set	Cannot be set	Can be set
1	Can be set	Can be set	Cannot be set



Note 1. Resetting is not necessary in high-speed mode.

Figure 9.2 Example of MEMWAIT Bit Setting Procedure When Changing ICLK Frequency to Higher than 32 MHz



Note 1. Resetting is not necessary when the state is not changed.

Figure 9.3 Example of MEMWAIT Bit Setting Procedure When Changing ICLK Frequency to Lower than 32 MHz

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

Figure 9.4 shows an example of connecting a crystal.
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.

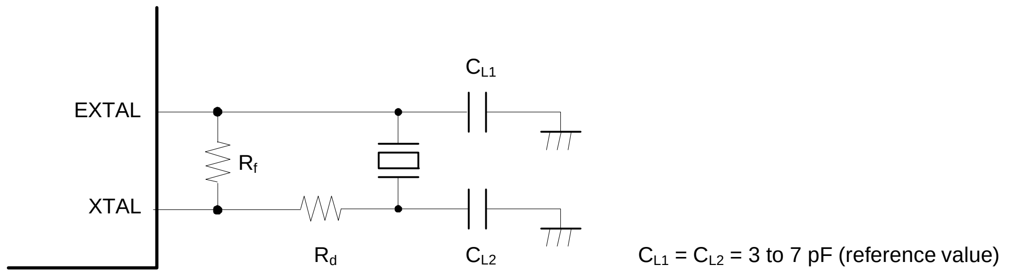


Figure 9.4 Example of Crystal Connection

Table 9.4 Damping Resistance (Reference Values)

Frequency (MHz)	2	8	16	20
Rd (Ω)	0	0	0	0

Figure 9.5 shows an equivalent circuit of the crystal. Use a crystal that has the characteristics shown in Table 9.5 as a reference.

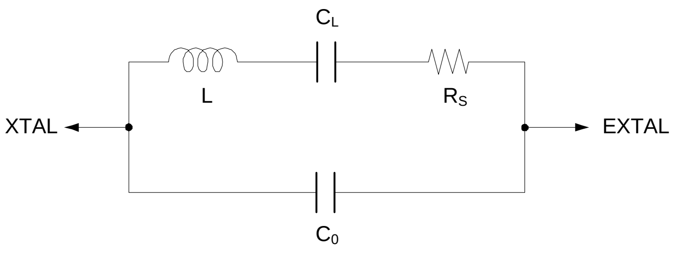


Figure 9.5 Equivalent Circuit of Crystal

Table 9.5 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16
R_S max (Ω)	200	120	56
C_0 max (pF)	1.3	1.3	1.4

9.3.2 External Clock Input

Figure 9.6 shows 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.

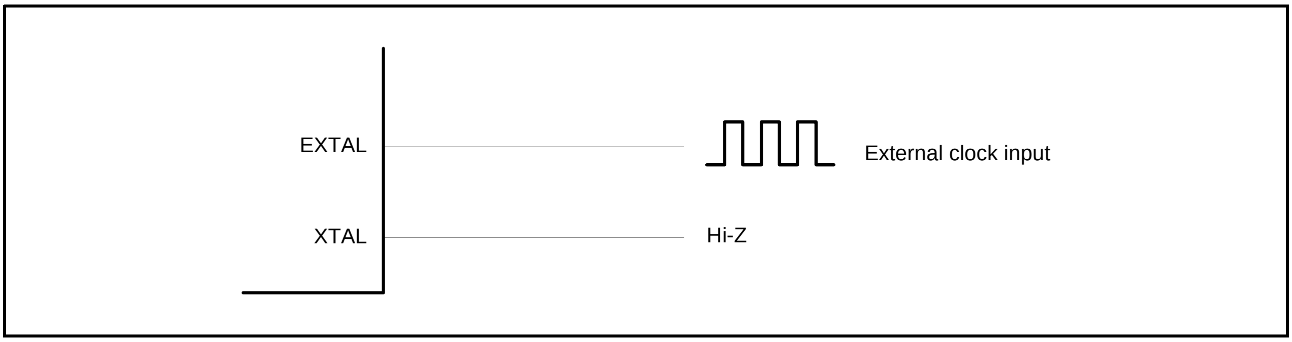


Figure 9.6 Connection Example of External Clock

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 (main clock oscillator is operating).

9.4 Oscillation Stop Detection Function

9.4.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock.

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, refer to section 20, Multi-Function Timer Pulse Unit (MTU3c) and section 21, Port Output Enable 3 (POE3b).

In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (refer to section 35, 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].

If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bit value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.

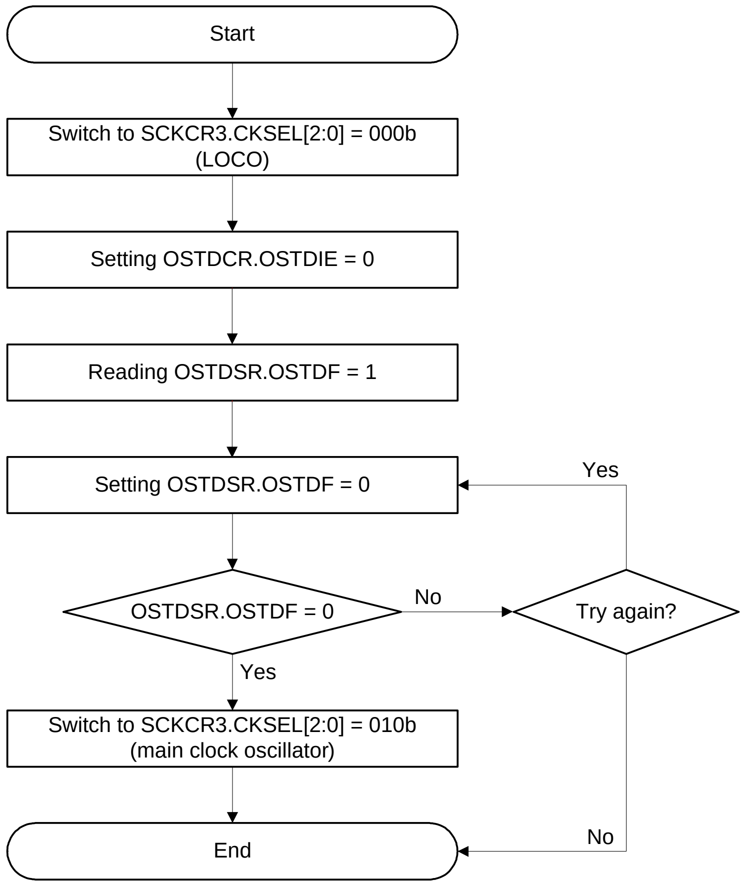
Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and set the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation 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.

When the system clock with the main clock selected as the system clock source, CAC main clock (CACMCLK) is selected, these clocks are switched to the LOCO clock by the oscillation stop detection. 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 (ICLK) select bits (SCKCR.ICK[3:0]).

When the PLL clock is selected as the system clock source, these clocks operate at the PLL free-running frequency by the oscillation stop detection.



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 9.7 Flow of Recovery from Detection of Oscillator Stop

9.4.2 Oscillation Stop Detection Interrupts

The main clock oscillator stop is notified to port output enable (POE) if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). 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 PCLKB before writing to this ICSR3.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so setting the oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE) to 0. 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, refer to section 14, Interrupt Controller (ICUb).

When the PLL detects an oscillation stop and is running at its own oscillation frequency, this indicates the occurrence of some system failure. An emergency measure should be taken to handle the failure.

9.5 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.6 Internal Clock

Clock sources of internal clock signals are the main clock, HOCO clock, LOCO clock, PLL clock, and dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of peripheral modules: Peripheral module clock (PCLKA, 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 IWDT: IWDT-dedicated low-speed clock (IWDTCLK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0], the clock source selected by the SCKCR3.CKSEL[2:0] bits, and the bits that select the frequency of the PLL circuit (PLLCR.STC[5:0] and PLIDIV[1:0] bits). 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, DTC, ROM, and RAM.
The ICLK frequency is specified by the SCKCR.ICK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

9.6.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, and PCLKD) are the operating clocks for use by peripheral modules.
The PCLKA, PCLKB, and PCLKD frequencies are specified by the SCKCR.PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits, the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.
The peripheral module clock used as the operating clock is PCLKD for S12AD, and PCLKA and PCLKB are for other modules.

9.6.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
The FCLK frequency is specified by the SCKCR.FCK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

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, 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 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 (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) supplied to each module can be selected by the SCKCR register. 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) The relationship of frequencies of the system clock (ICLK), peripheral module clocks A, B, and D (PCLKA, PCLKB, and PCLKD), and FlashIF clock (FCLK) must be set as follows.
ICLK:FCLK = N:1 (N is an integer)
ICLK:PCLKA, PCLKB, and PCLKD = N:1 (N is an integer)

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

9.7.2 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, 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 9.8 to prevent electromagnetic induction from interfering with correct oscillation.

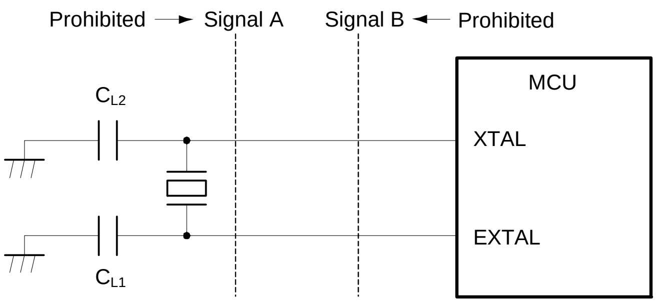


Figure 9.8 Notes on Board Design for Oscillation Circuit

9.7.4 Notes on Resonator Connection Pins

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When using these pins as general ports, be sure to stop the main clock (MOSCCR.MOSTP = 1). However, do not use the EXTAL and XTAL pins as general ports P36 and P37 in a system that uses the main clock.

When the main clock is used, do not set P36 and P37 to output.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • HOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • HOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

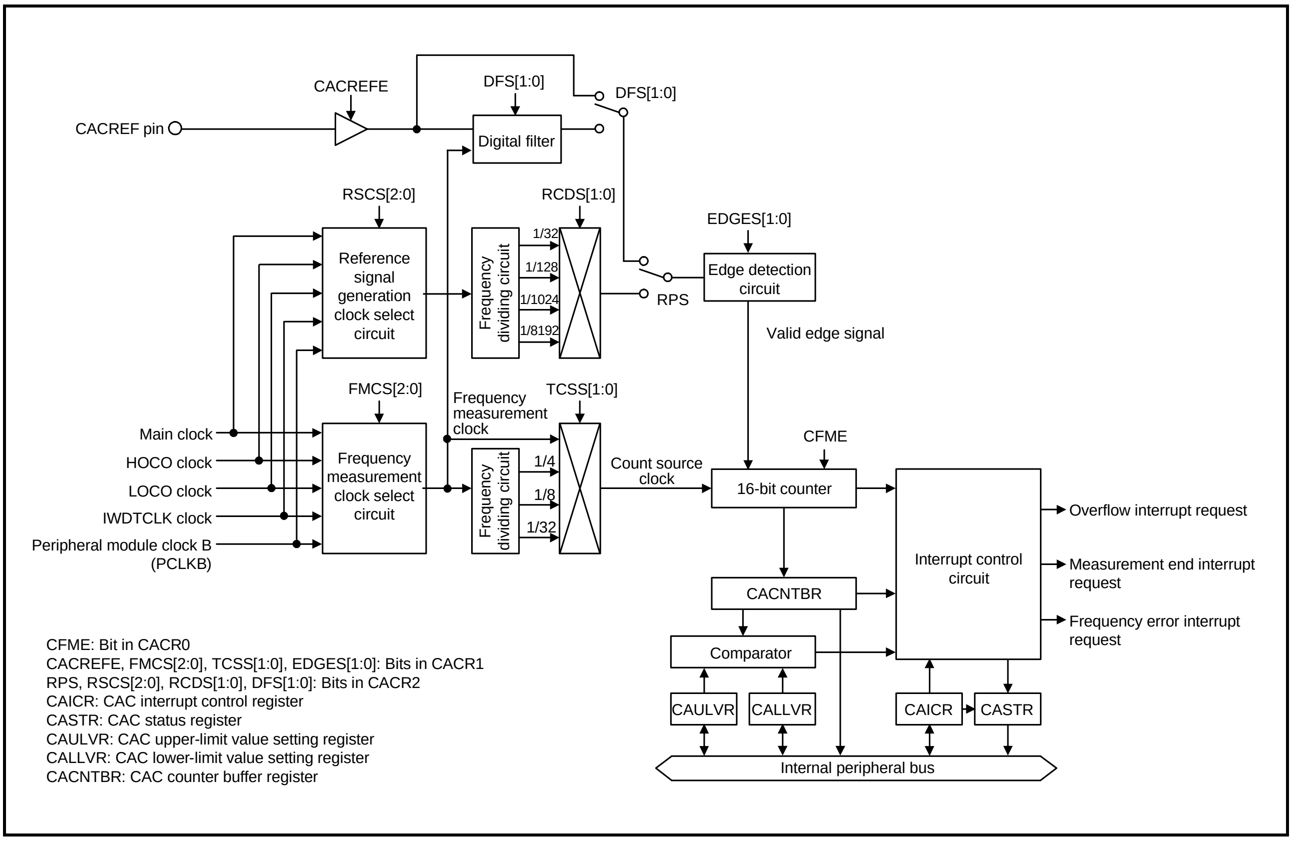


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): 0008 B000h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): 0008 B001h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note 1. Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

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

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): 0008 B002h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the frequency measuring clock. 1 0: The sampling clock for the digital filter is the frequency measuring clock divided by 4. 1 1: The sampling clock for the digital filter is the frequency measuring clock divided by 16.	R/W

Note 1. Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

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

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

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

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

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

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

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): 0008 B006h

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

CAULVR is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): 0008 B008h

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

CALLVR is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): 0008 B00Ah

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

CACNTBR is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.
The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

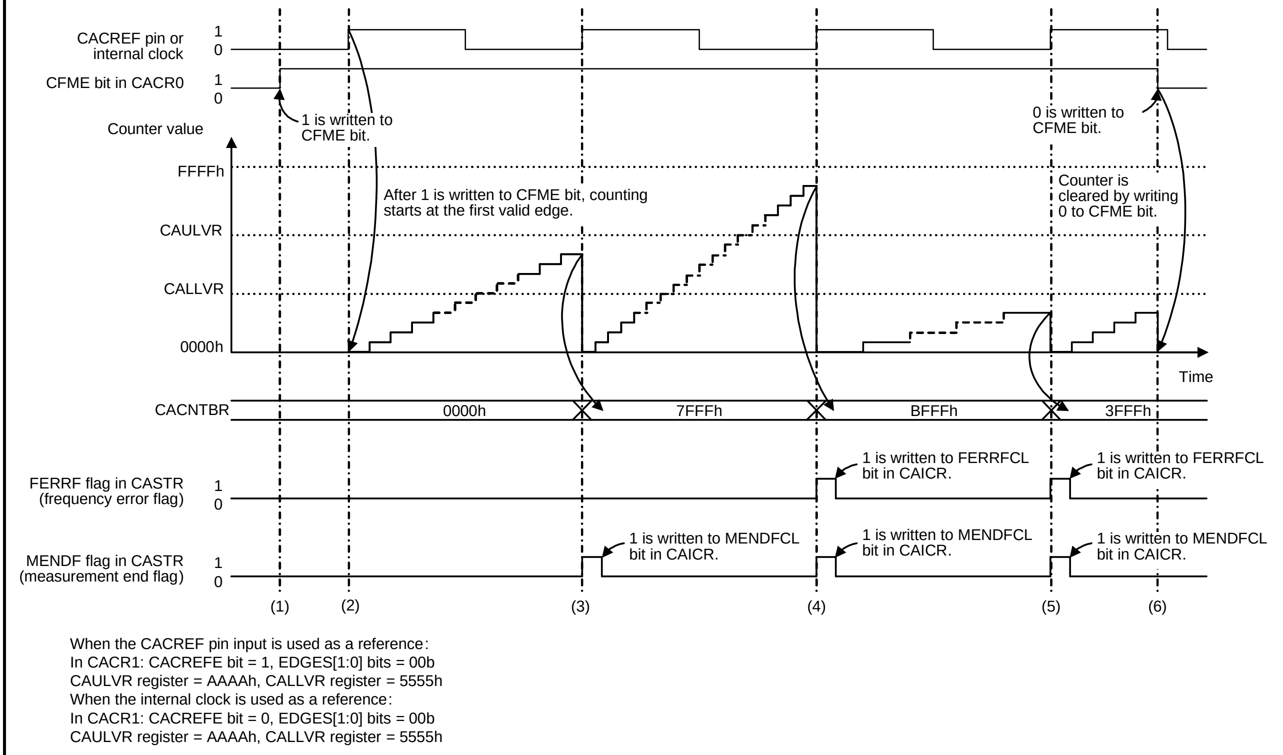


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.

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

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

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

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

(6) While the CFME bit in CACR0 is 1, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in CACNTBR may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.
After a reset, this MCU returns to normal mode, but modules except the DTC, and RAM are stopped.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Clock divider functions	The frequency division ratio can be set independently for the system clock (ICLK), high speed peripheral module clock (PCLKA), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Each peripheral module can be stopped independently by the module stop control register.
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 • Deep sleep mode • Software standby mode
Operating power control modes	• Power consumption can be reduced in normal operation, sleep mode, and deep sleep mode by selecting an appropriate operating power control mode according to the operating frequency and operating voltage. • Two operating power control modes are available High-speed operating mode Middle-speed operating mode

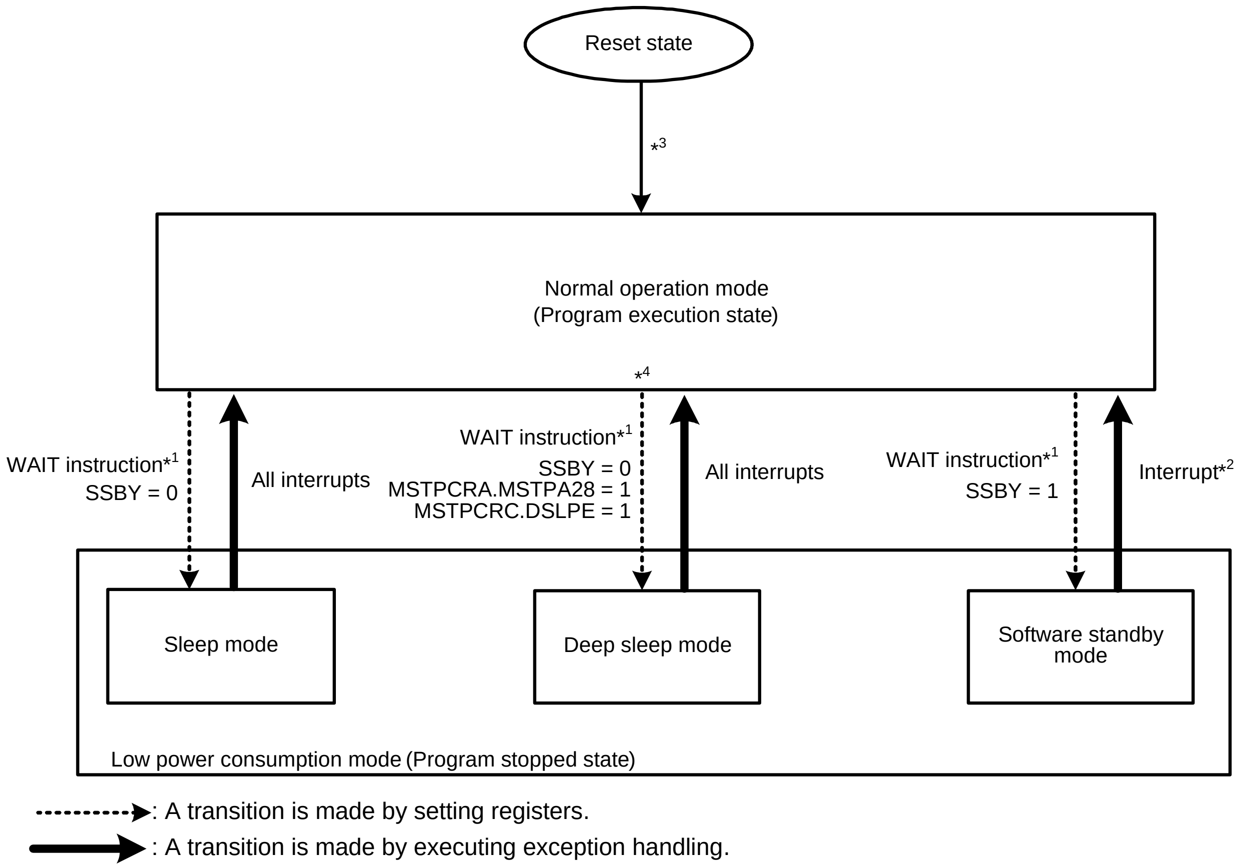
Note 1. For details, refer to section 9, Clock Generation Circuit.

Table 11.2 Operating Conditions of Each Power Consumption Mode

	Sleep Mode	Deep Sleep Mode	Software Standby Mode
Entry trigger	Control register + instruction	Control register + instruction	Control register + instruction
Exit trigger	Interrupt	Interrupt	Interrupt*1
After exiting from each mode, CPU begins from*2	Interrupt handling	Interrupt handling	Interrupt handling
Main clock oscillator	Operating possible	Operating possible	Stopped
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*3	Operating possible*3	Operating possible*3
PLL	Operating possible	Operating possible	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0000 27FFh, 0000 4000h to 0000 4A7Fh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
DTC	Operating possible*5	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*3	Operating possible*3	Operating possible*3
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)*4
I/O ports	Operating	Operating	Retained
Comparator C	Operating possible	Operating possible	Operating possible*6

"Operating possible" means that operating or stopped can be controlled by the register setting.
"Stopped (Retained)" means that internal register values are retained and internal operations are suspended.

Note 1. "Interrupts" here indicates an external pin interrupt (the NMI or IRQ0 to IRQ5) or any of peripheral interrupts (the IWDT, and voltage monitoring interrupts).
Note 2. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
Note 3. Operating 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, operating 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 4. The peripheral logic states are retained.
Note 5. During sleep mode, do not write to the system control related registers (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order).
Note 6. Using the digital filter function is prohibited. Operation for outputting the comparison result to the COMPn pin is possible.



Note 1. If an interrupt is accepted while entering low power consumption mode, the transition to the program stop state is exited and the interrupt exception handling is executed.

Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ5) or any of peripheral interrupts (the IWDT and voltage monitoring interrupts).

Note 3. The LOCO is the clock source following a transition from the reset state to normal mode.

Note 4. Makes a transition from sleep mode, deep sleep mode, or software standby mode to normal operating mode by an interrupt.
For deep sleep mode and software standby mode, the clock source after exiting is the same as that of the clock before entering software standby mode.

Transition to the reset state is made at any state instantly after a RES# pin reset, power on reset, voltage monitoring reset, IWDT reset, or software reset is generated.

Figure 11.1 Mode Transitions

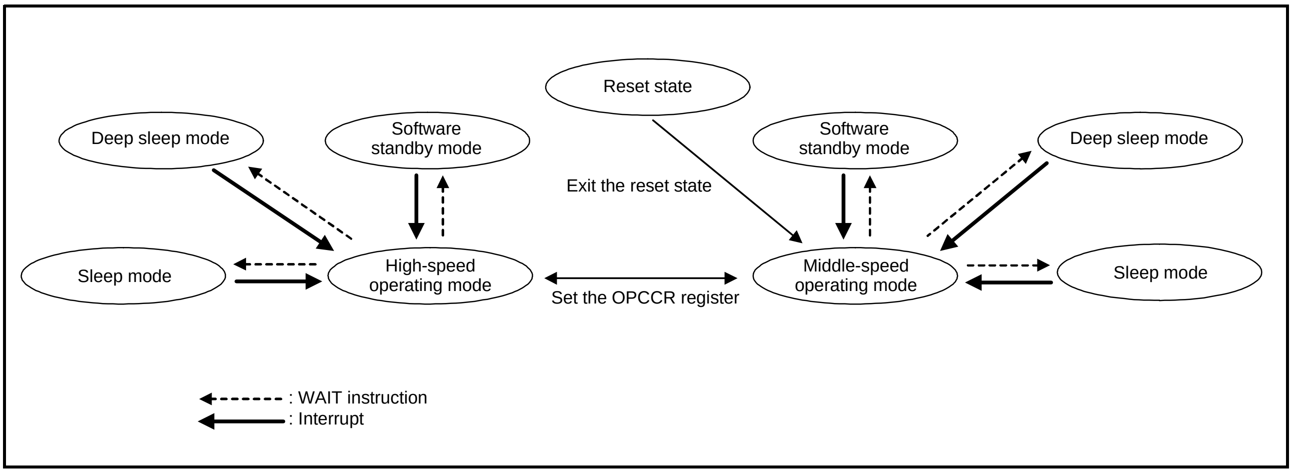


Figure 11.2 Operating Modes

- It is possible to return from sleep mode to the previous operating state used before entering sleep mode.
- After exiting the reset state, operation starts in middle-speed operating mode.

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

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

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	SSBY	Software Standby	0: Set entry to sleep mode or deep sleep mode after the WAIT instruction is executed 1: Set entry 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 MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal mode after an interrupt has triggered and exits from software standby mode, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to the SSBY bit.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) in the oscillation stop detection control register is 1, the set value of the SSBY bit is invalid. Even if the SSBY bit is 1, the MCU will enter sleep mode or deep sleep mode after 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
	—	—	—	MSTPA 28	—	—	—	—	—	—	—	—	MSTPA 19	—	MSTPA 17	—
Value after reset:	1	1	1	0	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
	MSTPA 15	MSTPA 14	—	—	—	—	MSTPA 9	—	—	—	MSTPA 5	MSTPA 4	—	—	—	—
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 and 2 (Unit 1) Module Stop	Target module: TMR3, TMR2 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	MSTPA5	8-bit Timer 1 and 0 (Unit 0) Module Stop	Target module: TMR1, TMR0 0: This module clock is enabled 1: This module clock is disabled	R/W
b8 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 3 Module Stop	Target module: MTU3 (MTU0 to MTU5) 0: This module clock is enabled 1: This module clock is disabled	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 1 (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: This module clock is enabled 1: This module clock is disabled	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: This module clock is enabled 1: This module clock is disabled	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: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	D/A Converter for generating comparator C reference voltage Module Stop	Target module: DA 0: This module clock is enabled 1: This module clock is disabled	R/W
b27 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPA28	Data Transfer Controller Module Stop	Target module: DTC 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b29	—	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.

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	—	—	MSTPB23	—	MSTPB21	—	—	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPB10	—	—	—	MSTPB6	—	—	—	—	—	—
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
b5 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: This module clock is enabled 1: This module clock is disabled	R/W
b9 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPB10	Comparator C Module Stop	Target module: Comparator C 0: This module clock is enabled 1: This module clock is disabled	R/W
b16 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: This module clock is enabled 1: This module clock is disabled	R/W
b20 to b18	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: This module clock is enabled 1: This module clock is disabled	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: This module clock is enabled 1: This module clock is disabled	R/W
b25, b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: This module clock is enabled 1: This module clock is disabled	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: This module clock is enabled 1: This module clock is disabled	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
	DSLPE	—	—	—	—	—	—	—	—	—	—	—	MSTPC19	—	—	—
Value after reset:	0	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 27FFh, 0000 4000h to 0000 4A7Fh) 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: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	DSLPE	Deep Sleep Mode Enable	0: Deep sleep mode is disabled 1: Deep sleep mode is enabled	R/W

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

Note 1. The corresponding MSTPC0 bit should not be set to 1 during access to the RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is 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.

DSLPE Bit (Deep Sleep Mode Enable)

The DSLPE bit enables or disables a transition to deep sleep mode.

When the CPU executes the WAIT instruction with the DSLPE bit set to 1 and the SBYCR.SSBY and MSTPCRA.MSTPA28 bits meet specified conditions, the MCU enters deep sleep mode. For details, refer to section 11.6.2, Deep Sleep Mode.

11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCM TSF	—	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 0 0: High-speed operating mode 0 1 0: Middle-speed operating mode 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	0: Transition completed 1: During transition	R
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.

The OPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

The OPCCR register cannot be rewritten under the following conditions:

- When the OPCCR.OPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by ‘SYSTEM’ in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

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 deep sleep mode. Table 11.3 shows the relationship between operating power control modes, the OPCM[2:0] bit settings, and the operating frequency and voltage ranges.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

This flag indicates the switching control state during and after operating power mode transition.
This flag becomes 1 when the value of the OPCM[2:0] bits is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the OPCM[2:0] bits when this flag is 0.

Table 11.3 Operating Frequency and Voltage Ranges in Operating Power Control Modes

Operating Power Control Mode	OPCM [2:0] Bits	Operating Voltage Range	Operating Frequency Range					
			Flash Memory Read Frequency					Flash Memory Programming/ Erasure Frequency
			ICLK	FCLK	PCLKD	PCLKB	PCLKA	FCLK
High-speed operating mode	000b	2.7 to 5.5 V	Up to 40 MHz	Up to 32 MHz	Up to 40 MHz	Up to 40 MHz	Up to 40 MHz	1 to 32 MHz
Middle-speed operating mode	010b	2.7 to 5.5 V	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	1 to 12 MHz

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

Each operating power control mode is described below.

- High-Speed Operating Mode

The maximum operating frequency during FLASH read is 40MHz for ICLK, PCLKA, PCLKB, and PCLKD; 32MHz for FCLK.
During FLASH programming/erasure, the operating frequency range is 1 to 32 MHz.

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

- Middle-Speed Operating Mode

As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.
The maximum operating frequency during FLASH read is 12 MHz for ICLK, FCLK, PCLKA, PCLKB, and PCLKD.
During FLASH programming/erasure, the operating frequency range is 1 to 12MHz.
The power consumption of this mode is lower than that of high speed mode under the same conditions.
After a reset is canceled, operation is started from this mode.

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits. The peripheral module clocks can be set by the BCK[3:0], PCKA[3:0], PCKB[3:0], and PCKD[3:0] bits.
The flash memory clock can be set by the FCK[3:0] bits.
For details, refer to 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. When the corresponding MSTPmi bit is set to 0, the module exits the module state and restarts 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 DTC, and on-chip RAM are in the module stop state. Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

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 mode, sleep mode, and deep sleep 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

- From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
(Middle-speed operation in middle-speed operating mode)

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

- From middle-speed operating mode to high-speed operating mode

Middle-speed operation in middle-speed operating mode
↓

Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)

↓

Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)

↓

Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)

↓

Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode

↓

High-speed operation in high-speed operating mode

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Entry 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 IWDTCSTPR.SLCSTP bit 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 IWDTCSTPR.SLCSTP bit is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value 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, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.1.2 Exit from Sleep Mode

Exit from sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from sleep mode 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 exited.
- Initiated by a RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in sleep mode and sleep mode is not exited by the independent watchdog timer reset.

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

11.6.2 Deep Sleep Mode

11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.*1

In deep sleep mode, the CPU and the DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.

Counting by the IWDT stops if a transition to deep 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 deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to deep 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 deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.

To use deep sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*2 of the CPU to 0.

(2) Set the interrupt request destination*3 to be used for exit from deep sleep mode.

(3) Set the priority*4 of the interrupt to be used for exit from deep sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*2 of the CPU.

(4) Set the IERm.IENj bit*4 to 1 for the interrupt.

(5) Read the I/O register that is written last and confirm that the written value has been reflected.

(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*2 of the CPU to 1).

Note 1. Transition to deep sleep mode might not be possible, depending on the operating state of the DTC. Before setting the MSTPCRA.MSTPA28 bit to 1, set the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC.

Note 2. For details, refer to section 2, CPU.

Note 3. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.

Note 4. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.2.2 Exit from Deep Sleep Mode

Exit from deep sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from deep sleep mode 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), deep sleep mode is not exited.
- Initiated by the RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by the independent watchdog timer
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in deep sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in deep sleep mode and deep sleep mode is not exited by the independent watchdog timer reset.

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

11.6.3 Software Standby Mode

11.6.3.1 Entry 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 stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Set 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 IWDTCSTPR.SLCSTP bit 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 IWDTCSTPR.SLCSTP bit is 0.
To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value 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, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.3.2 Exit from Software Standby Mode

Exit from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ5), peripheral function interrupts (the IWDT, and voltage monitoring), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Initiated by an interrupt
  When an interrupt request from among the NMI, IRQ0 to IRQ5, IWDT, voltage monitoring interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU.
  Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.3 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by 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 entry to software standby mode is completed. After that, exit from software standby mode is initiated by the rising edge of the IRQn pin.
To exit software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 14, Interrupt Controller (ICUb).

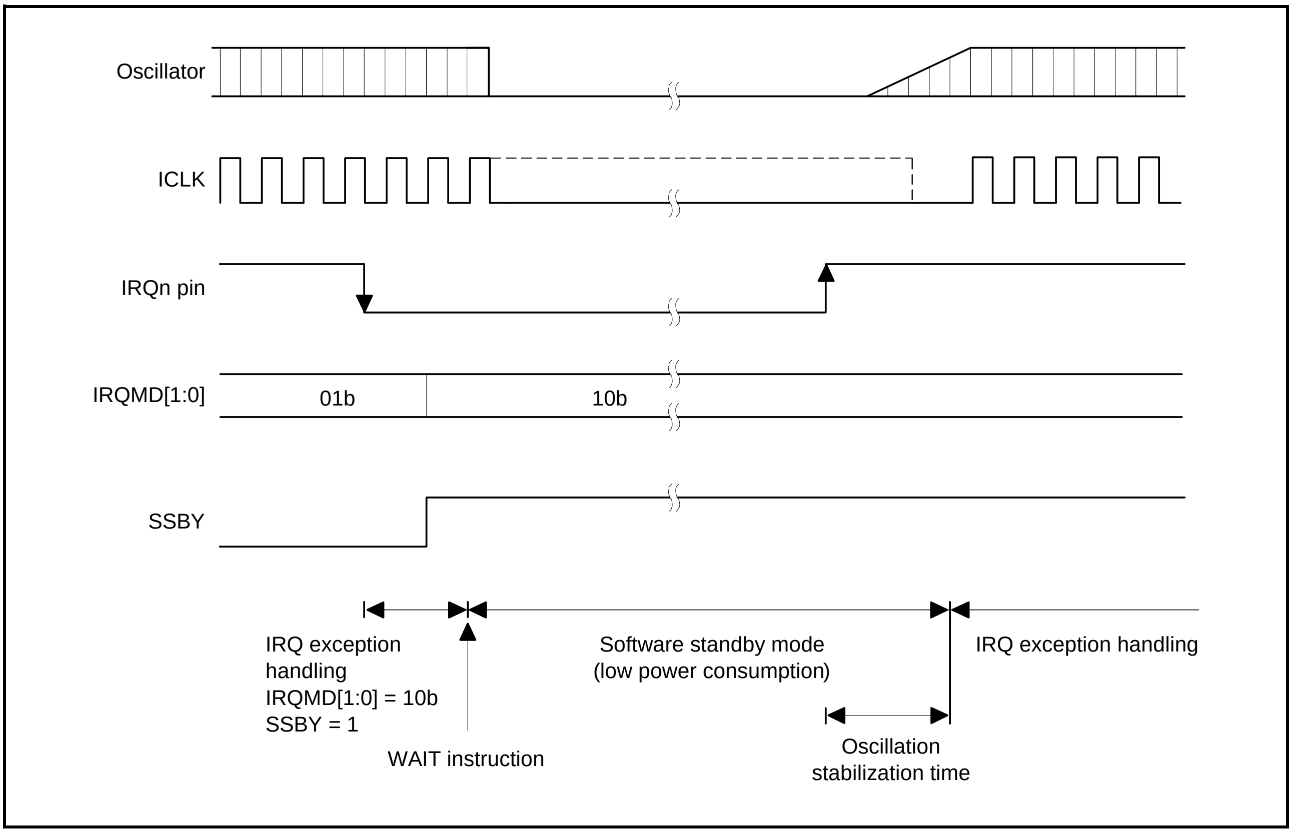


Figure 11.3 Example of Software Standby Mode Application

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 if output signals are high level.

11.7.2 Module Stop State of DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC.
For details, refer to 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 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 being executed before the register setting is modified may cause unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last register setting is done.

11.7.6 Rewrite the Register by DTC in Sleep Mode

Depending on the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. To avoid this, do not set up the DTC to rewrite any registers related to the IWDT in sleep mode.

12. Register Write Protection Function

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

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

Table 12.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, LOCOCR, ILOCOCR, HOCOCR, OSTDCR, OSTDSR, MEMWAIT
PRC1	• Register related to the operating modes: SYSCR1 • Registers related to low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR • Registers related to the clock generation circuit: MOFCR, MOSCWTCR • Software reset register: SWRR
PRC2	• Register related to the clock generation circuit: HOCOWTCR
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	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption functions, the clock generation circuit, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	PRC2	Protect Bit 2	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	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 8 higher-order bits and the desired value to the 8 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 to 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 RXv2 CPU supports eight types of exceptions. The types of exception events are shown in Figure 13.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

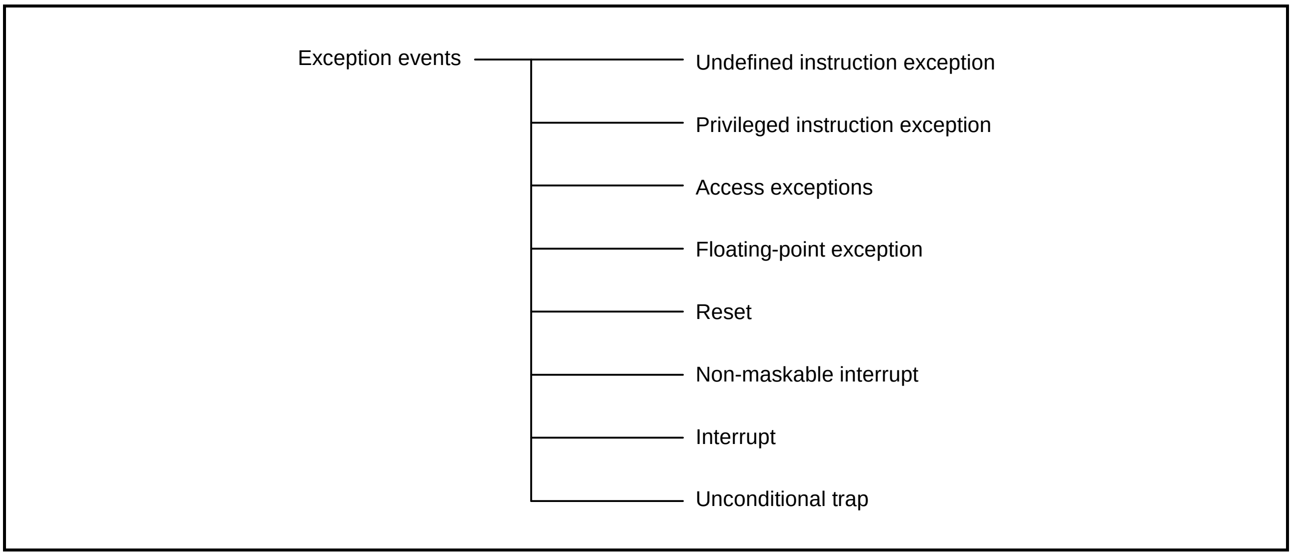


Figure 13.1 Types of Exception Events

13.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

13.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

13.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

13.1.4 Floating-Point Exception

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

13.1.5 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

13.1.6 Non-Maskable Interrupt

The non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

13.1.7 Interrupt

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

13.1.8 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

13.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 13.2 shows the processing procedure when an exception other than a reset is accepted.

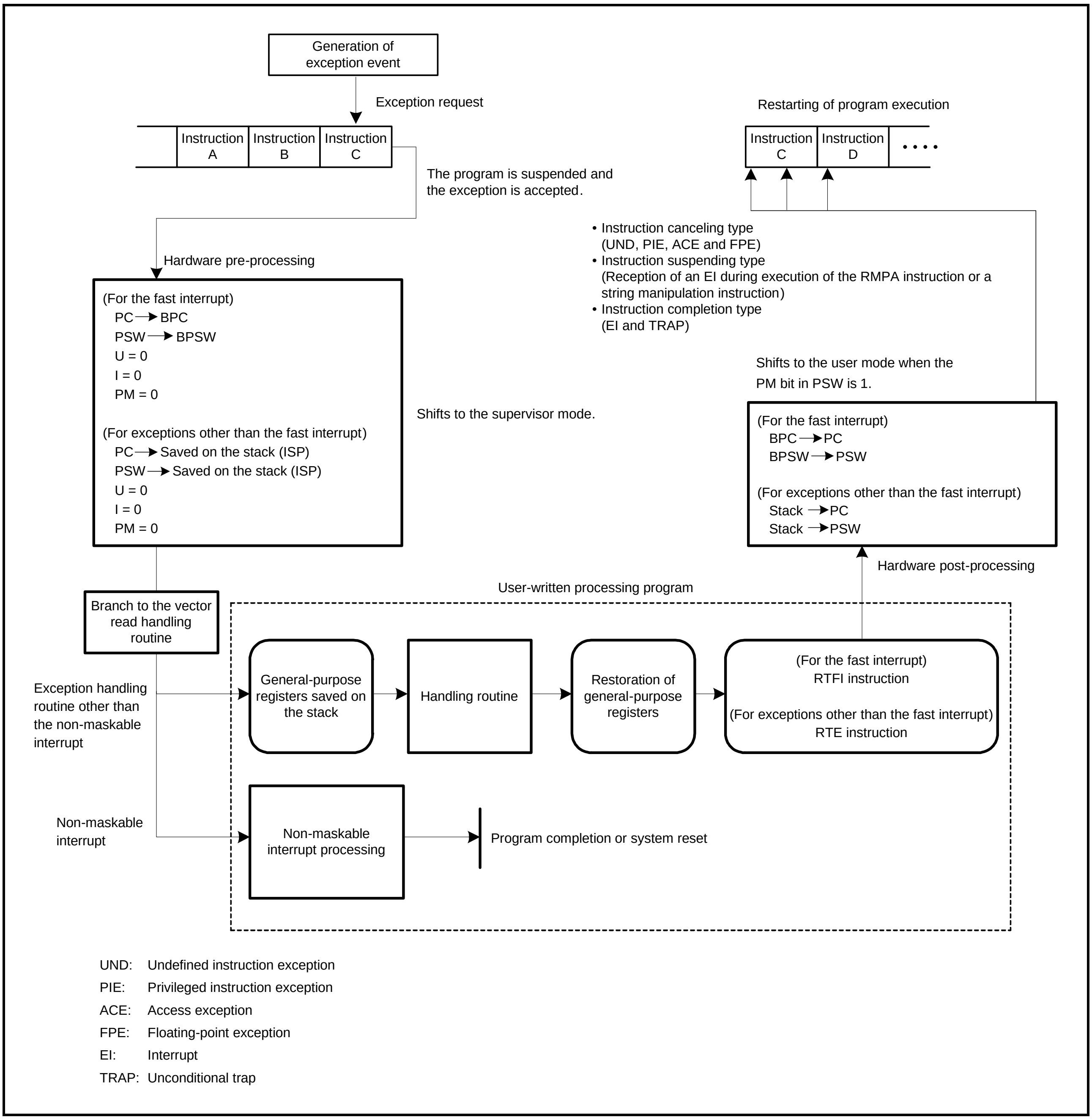


Figure 13.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RXv2 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 RXv2 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 RXv2 CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

13.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

13.3.1 Acceptance Timing and Saved PC Value

Table 13.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 13.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

13.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 13.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 13.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Exception vector table (EXTB)	Stack
Privileged instruction exception		Exception vector table (EXTB)	Stack
Access exception		Exception vector table (EXTB)	Stack
Floating-point exception		Exception vector table (EXTB)	Stack
Reset		Exception vector table (EXTB)	Nowhere
Non-maskable interrupt		Exception vector table (EXTB)	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Interrupt vector table (INTB)	Stack
Unconditional trap		Interrupt vector table (INTB)	Stack

13.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The values in FPSW are not saved by hardware pre-processing. Therefore, if floating-point instructions are to be used within an exception handling routine, the user must save these values on the stack within the exception handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

13.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.2 Privileged Instruction Exception

1. The value 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 the value of EXTB + address 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.3 Access Exceptions

1. The value 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 the value of EXTB + address 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.4 Floating-Point Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.5 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

13.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are 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 the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.7 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are 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 interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.8 Unconditional Trap

1. The value 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 interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 13.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Prohibited
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 13.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Operand access exception
Low	8	Floating-point exception

14. Interrupt Controller (ICUb)

14.1 Overview

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

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 IRQ5 • Number of sources: 6 • 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
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC control	The DTC 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, deep sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ5 interrupts.

Note 1. For the DTC activation source, refer to Table 14.3, Interrupt Vector Table.

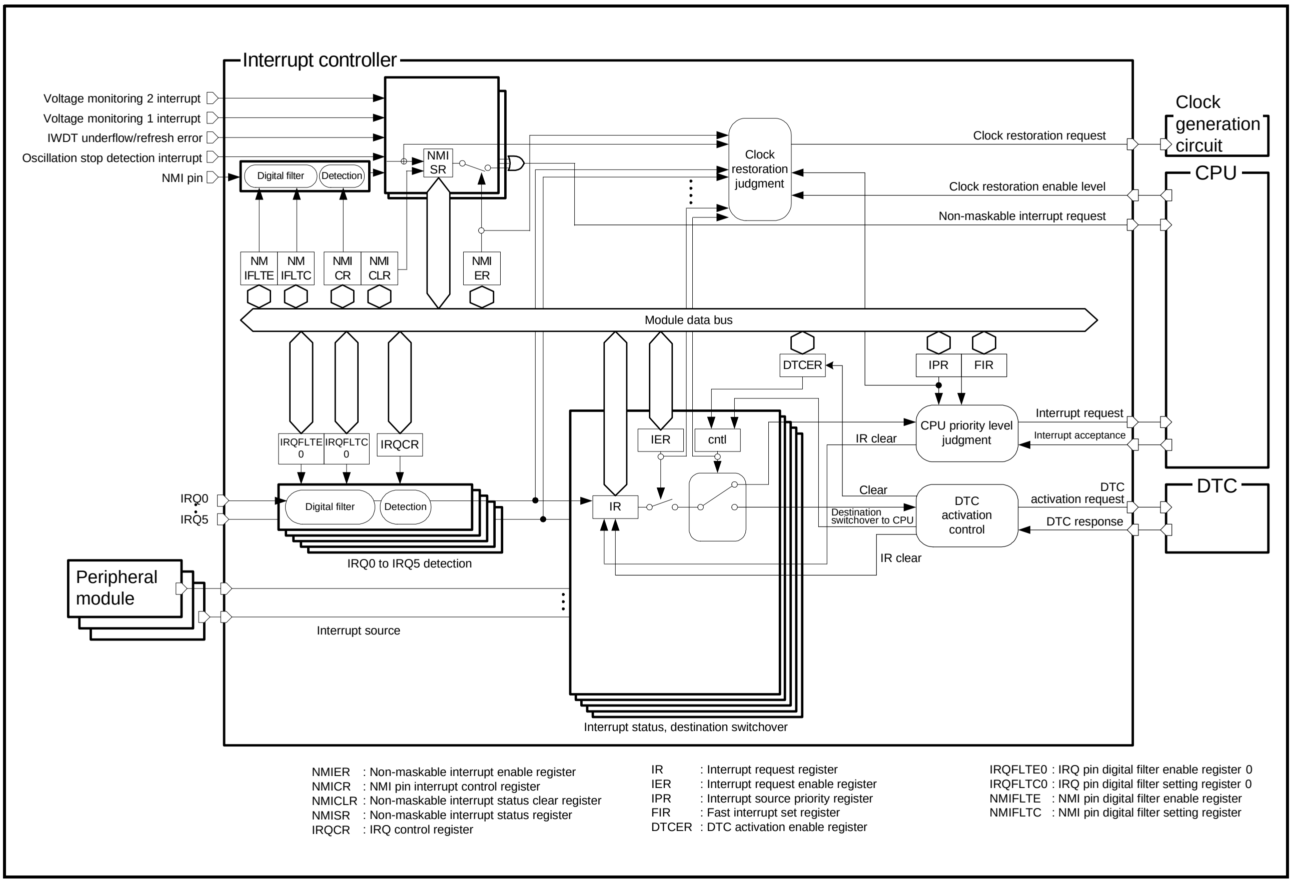


Figure 14.1 Block Diagram of Interrupt Controller

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 IRQ5	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 70FFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated 1: An interrupt request is generated	R/(W) *1
b7 to b1	—	Reserved	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 flag 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 (i = 0 to 5) pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits. 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.

(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 Bit (Interrupt Request Enable j) (j = 0 to 7)

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 (n = interrupt vector number). 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 = interrupt vector number)

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

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled)*1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	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.

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 (m = 02h to 1Fh, j = 0 to 7)).

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.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0 (m = 02h to 1Fh, j = 0 to 7)).

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 (n = interrupt vector number). When using the fast interrupt for returning from the software standby mode, see section 14.6.2, Return from Software Standby Mode.
If the setting of the IERm.IENj 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 71FFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC 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

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 IRQ Control Register i (IRQCRi) (i = 0 to 5)

Address(es): 0008 7500h to 0008 7505h

	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 (m = 02h to 1Fh, j = 0 to 7) 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 interrupt detection sensing method of IRQi pin.
For the external pin interrupt detection setting, see section 14.4.8, External Pin Interrupts.

14.2.8 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7510h

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

FLTENi Bit (IRQi Digital Filter Enable) (i = 0 to 5)

This bit enables the digital filter used for the IRQi pin.

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.

14.2.9 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
	—	—	—	—	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
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 5)

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

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.

14.2.10 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

14.2.11 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.

14.2.12 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.

14.2.13 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.

14.2.14 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.

14.2.15 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 4-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.
sstb return	“✓” in this column indicates usability as a request for return from software-standby mode.
IER	Name of the IER register and bit corresponding to the vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register 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	sstb Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	N/A	N/A	N/A	—	—	—
—	Reserved	18	0048h	—	N/A	N/A	N/A	—	—	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	—	—	—
—	Reserved	21	0054h	—	N/A	N/A	N/A	—	—	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	—	—	—
—	Reserved	26	0068h	—	N/A	N/A	N/A	—	—	—
ICU	SWINT	27	006Ch	Edge	✓	✓	N/A	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	✓	✓	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	N/A	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	✓	✓	N/A	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	✓	✓	N/A	IER03.IEN7	IPR007	DTCER031
CAC	FERRF	32	0080h	Level	✓	N/A	N/A	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	✓	N/A	N/A	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	✓	N/A	N/A	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	36	0090h	—	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	—	—	—
—	Reserved	38	0098h	—	N/A	N/A	N/A	—	—	—
—	Reserved	39	009Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	40	00A0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	41	00A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	42	00A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	43	00ACh	—	N/A	N/A	N/A	—	—	—
RSPI0	SPEI0	44	00B0h	Level	✓	N/A	N/A	IER05.IEN4	IPR044	—
	SPRI0	45	00B4h	Edge	✓	✓	N/A	IER05.IEN5		DTCER045
	SPTI0	46	00B8h	Edge	✓	✓	N/A	IER05.IEN6		DTCER046
	SPII0	47	00BCh	Level	✓	N/A	N/A	IER05.IEN7		—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	—	—	—

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	sstb Return	IER	IPR	DTCER
—	Reserved	51	00CCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	52	00D0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	53	00D4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	54	00D8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	55	00DCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	56	00E0h	—	N/A	N/A	N/A	—	—	—
DOC	DOPCF	57	00E4h	Level	✓	N/A	N/A	IER07.IEN1	IPR057	—
—	Reserved	58	00E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	59	00ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	60	00F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	61	00F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	62	00F8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	63	00FCh	—	N/A	N/A	N/A	—	—	—
ICU	IRQ0	64	0100h	Edge/Level	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/Level	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	Edge/Level	✓	✓	✓	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	Edge/Level	✓	✓	✓	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	Edge/Level	✓	✓	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/Level	✓	✓	✓	IER08.IEN5	IPR069	DTCER069
—	Reserved	70	0118h	—	N/A	N/A	N/A	—	—	—
—	Reserved	71	011Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	72	0120h	—	N/A	N/A	N/A	—	—	—
—	Reserved	73	0124h	—	N/A	N/A	N/A	—	—	—
—	Reserved	74	0128h	—	N/A	N/A	N/A	—	—	—
—	Reserved	75	012Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	76	0130h	—	N/A	N/A	N/A	—	—	—
—	Reserved	77	0134h	—	N/A	N/A	N/A	—	—	—
—	Reserved	78	0138h	—	N/A	N/A	N/A	—	—	—
—	Reserved	79	013Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	80	0140h	—	N/A	N/A	N/A	—	—	—
—	Reserved	81	0144h	—	N/A	N/A	N/A	—	—	—
—	Reserved	82	0148h	—	N/A	N/A	N/A	—	—	—
—	Reserved	83	014Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	84	0150h	—	N/A	N/A	N/A	—	—	—
—	Reserved	85	0154h	—	N/A	N/A	N/A	—	—	—
—	Reserved	86	0158h	—	N/A	N/A	N/A	—	—	—
—	Reserved	87	015Ch	—	N/A	N/A	N/A	—	—	—
LVD	LVD1	88	0160h	Edge	✓	N/A	✓	IER0B.IEN0	IPR088	—
	LVD2	89	0164h	Edge	✓	N/A	✓	IER0B.IEN1	IPR089	—
—	Reserved	90	0168h	—	N/A	N/A	N/A	—	—	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	92	0170h	—	N/A	N/A	N/A	—	—	—
—	Reserved	93	0174h	—	N/A	N/A	N/A	—	—	—
—	Reserved	94	0178h	—	N/A	N/A	N/A	—	—	—
—	Reserved	95	017Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	96	0180h	—	N/A	N/A	N/A	—	—	—
—	Reserved	97	0184h	—	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	100	0190h	—	N/A	N/A	N/A	—	—	—
—	Reserved	101	0194h	—	N/A	N/A	N/A	—	—	—

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	sstb Return	IER	IPR	DTCER
S12AD	S12ADI	102	0198h	Edge	✓	✓	N/A	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	✓	✓	N/A	IER0C.IEN7	IPR103	DTCER103
—	Reserved	104	01A0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	105	01A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	106	01A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	107	01ACh	—	N/A	N/A	N/A	—	—	—
CMPC0	CMPC0	108	01B0h	Edge	✓	✓	N/A	IER0D.IEN4	IPR108	DTCER108
CMPC1	CMPC1	109	01B4h	Edge	✓	✓	N/A	IER0D.IEN5	IPR109	DTCER109
CMPC2	CMPC2	110	01B8h	Edge	✓	✓	N/A	IER0D.IEN6	IPR110	DTCER110
—	Reserved	111	01BCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	112	01C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	113	01C4h	—	N/A	N/A	N/A	—	—	—
MTU0	TGIA0	114	01C8h	Edge	✓	✓	N/A	IER0E.IEN2	IPR114	DTCER114
	TGIB0	115	01CCh	Edge	✓	✓	N/A	IER0E.IEN3		DTCER115
	TGIC0	116	01D0h	Edge	✓	✓	N/A	IER0E.IEN4		DTCER116
	TGID0	117	01D4h	Edge	✓	✓	N/A	IER0E.IEN5		DTCER117
	TCIV0	118	01D8h	Edge	✓	N/A	N/A	IER0E.IEN6	IPR118	—
	TGIE0	119	01DCh	Edge	✓	N/A	N/A	IER0E.IEN7		—
	TGIF0	120	01E0h	Edge	✓	N/A	N/A	IER0F.IEN0		—
MTU1	TGIA1	121	01E4h	Edge	✓	✓	N/A	IER0F.IEN1	IPR121	DTCER121
	TGIB1	122	01E8h	Edge	✓	✓	N/A	IER0F.IEN2		DTCER122
	TCIV1	123	01ECh	Edge	✓	N/A	N/A	IER0F.IEN3	IPR123	—
	TCIU1	124	01F0h	Edge	✓	N/A	N/A	IER0F.IEN4		—
MTU2	TGIA2	125	01F4h	Edge	✓	✓	N/A	IER0F.IEN5	IPR125	DTCER125
	TGIB2	126	01F8h	Edge	✓	✓	N/A	IER0F.IEN6		DTCER126
	TCIV2	127	01FCh	Edge	✓	N/A	N/A	IER0F.IEN7	IPR127	—
	TCIU2	128	0200h	Edge	✓	N/A	N/A	IER10.IEN0		—
MTU3	TGIA3	129	0204h	Edge	✓	✓	N/A	IER10.IEN1	IPR129	DTCER129
	TGIB3	130	0208h	Edge	✓	✓	N/A	IER10.IEN2		DTCER130
	TGIC3	131	020Ch	Edge	✓	✓	N/A	IER10.IEN3		DTCER131
	TGID3	132	0210h	Edge	✓	✓	N/A	IER10.IEN4		DTCER132
	TCIV3	133	0214h	Edge	✓	N/A	N/A	IER10.IEN5	IPR133	—
MTU4	TGIA4	134	0218h	Edge	✓	✓	N/A	IER10.IEN6	IPR134	DTCER134
	TGIB4	135	021Ch	Edge	✓	✓	N/A	IER10.IEN7		DTCER135
	TGIC4	136	0220h	Edge	✓	✓	N/A	IER11.IEN0		DTCER136
	TGID4	137	0224h	Edge	✓	✓	N/A	IER11.IEN1		DTCER137
	TCIV4	138	0228h	Edge	✓	✓	N/A	IER11.IEN2	IPR138	DTCER138
MTU5	TGIU5	139	022Ch	Edge	✓	✓	N/A	IER11.IEN3	IPR139	DTCER139
	TGIV5	140	0230h	Edge	✓	✓	N/A	IER11.IEN4		DTCER140
	TGIW5	141	0234h	Edge	✓	✓	N/A	IER11.IEN5		DTCER141
—	Reserved	142	0238h	—	N/A	N/A	N/A	—	—	—
—	Reserved	143	023Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	144	0240h	—	N/A	N/A	N/A	—	—	—
—	Reserved	145	0244h	—	N/A	N/A	N/A	—	—	—
—	Reserved	146	0248h	—	N/A	N/A	N/A	—	—	—
—	Reserved	147	024Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	148	0250h	—	N/A	N/A	N/A	—	—	—
—	Reserved	149	0254h	—	N/A	N/A	N/A	—	—	—
—	Reserved	150	0258h	—	N/A	N/A	N/A	—	—	—
—	Reserved	151	025Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	152	0260h	—	N/A	N/A	N/A	—	—	—

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	sstb Return	IER	IPR	DTCER
—	Reserved	153	0264h	—	N/A	N/A	N/A	—	—	—
—	Reserved	154	0268h	—	N/A	N/A	N/A	—	—	—
—	Reserved	155	026Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	156	0270h	—	N/A	N/A	N/A	—	—	—
—	Reserved	157	0274h	—	N/A	N/A	N/A	—	—	—
—	Reserved	158	0278h	—	N/A	N/A	N/A	—	—	—
—	Reserved	159	027Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	160	0280h	—	N/A	N/A	N/A	—	—	—
—	Reserved	161	0284h	—	N/A	N/A	N/A	—	—	—
—	Reserved	162	0288h	—	N/A	N/A	N/A	—	—	—
—	Reserved	163	028Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	164	0290h	—	N/A	N/A	N/A	—	—	—
—	Reserved	165	0294h	—	N/A	N/A	N/A	—	—	—
—	Reserved	166	0298h	—	N/A	N/A	N/A	—	—	—
—	Reserved	167	029Ch	—	N/A	N/A	N/A	—	—	—
POE	OEI1	168	02A0h	Level	✓	N/A	N/A	IER15.IEN0	IPR168	—
—	Reserved	169	02A4h	—	N/A	N/A	N/A	—	—	—
POE	OEI3	170	02A8h	Level	✓	N/A	N/A	IER15.IEN2	IPR168	—
	OEI4	171	02ACh	Level	✓	N/A	N/A	IER15.IEN3		—
—	Reserved	172	02B0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	173	02B4h	—	N/A	N/A	N/A	—	—	—
TMR0	CMIA0	174	02B8h	Edge	✓	✓	N/A	IER15.IEN6	IPR174	DTCER174
	CMIB0	175	02BCh	Edge	✓	✓	N/A	IER15.IEN7		DTCER175
	OVI0	176	02C0h	Edge	✓	N/A	N/A	IER16.IEN0		—
TMR1	CMIA1	177	02C4h	Edge	✓	✓	N/A	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	✓	✓	N/A	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	✓	N/A	N/A	IER16.IEN3		—
TMR2	CMIA2	180	02D0h	Edge	✓	✓	N/A	IER16.IEN4	IPR180	DTCER180
	CMIB2	181	02D4h	Edge	✓	✓	N/A	IER16.IEN5		DTCER181
	OVI2	182	02D8h	Edge	✓	N/A	N/A	IER16.IEN6		—
TMR3	CMIA3	183	02DCh	Edge	✓	✓	N/A	IER16.IEN7	IPR183	DTCER183
	CMIB3	184	02E0h	Edge	✓	✓	N/A	IER17.IEN0		DTCER184
	OVI3	185	02E4h	Edge	✓	N/A	N/A	IER17.IEN1		—
—	Reserved	186	02E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	187	02ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	188	02F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	189	02F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	190	02F8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	191	02FCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	192	0300h	—	N/A	N/A	N/A	—	—	—
—	Reserved	193	0304h	—	N/A	N/A	N/A	—	—	—
—	Reserved	194	0308h	—	N/A	N/A	N/A	—	—	—
—	Reserved	195	030Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	196	0310h	—	N/A	N/A	N/A	—	—	—
—	Reserved	197	0314h	—	N/A	N/A	N/A	—	—	—
—	Reserved	198	0318h	—	N/A	N/A	N/A	—	—	—
—	Reserved	199	031Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	200	0320h	—	N/A	N/A	N/A	—	—	—
—	Reserved	201	0324h	—	N/A	N/A	N/A	—	—	—
—	Reserved	202	0328h	—	N/A	N/A	N/A	—	—	—
—	Reserved	203	032Ch	—	N/A	N/A	N/A	—	—	—

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	sstb Return	IER	IPR	DTCER
—	Reserved	204	0330h	—	N/A	N/A	N/A	—	—	—
—	Reserved	205	0334h	—	N/A	N/A	N/A	—	—	—
—	Reserved	206	0338h	—	N/A	N/A	N/A	—	—	—
—	Reserved	207	033Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	208	0340h	—	N/A	N/A	N/A	—	—	—
—	Reserved	209	0344h	—	N/A	N/A	N/A	—	—	—
—	Reserved	210	0348h	—	N/A	N/A	N/A	—	—	—
—	Reserved	211	034Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	212	0350h	—	N/A	N/A	N/A	—	—	—
—	Reserved	213	0354h	—	N/A	N/A	N/A	—	—	—
—	Reserved	214	0358h	—	N/A	N/A	N/A	—	—	—
—	Reserved	215	035Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	216	0360h	—	N/A	N/A	N/A	—	—	—
—	Reserved	217	0364h	—	N/A	N/A	N/A	—	—	—
SCI1	ERI1	218	0368h	Level	✓	N/A	N/A	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	✓	✓	N/A	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	✓	✓	N/A	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	✓	N/A	N/A	IER1B.IEN5		—
SCI5	ERI5	222	0378h	Level	✓	N/A	N/A	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	✓	✓	N/A	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	✓	✓	N/A	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	✓	N/A	N/A	IER1C.IEN1		—
—	Reserved	226	0388h	—	N/A	N/A	N/A	—	—	—
—	Reserved	227	038Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	228	0390h	—	N/A	N/A	N/A	—	—	—
—	Reserved	229	0394h	—	N/A	N/A	N/A	—	—	—
—	Reserved	230	0398h	—	N/A	N/A	N/A	—	—	—
—	Reserved	231	039Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	232	03A0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	233	03A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	234	03A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	235	03ACh	—	N/A	N/A	N/A	—	—	—
—	Reserved	236	03B0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	237	03B4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	238	03B8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	239	03BCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	240	03C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	241	03C4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	242	03C8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	243	03CCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	244	03D0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	245	03D4h	—	N/A	N/A	N/A	—	—	—
RIIC0	EEI0	246	03D8h	Level	✓	N/A	N/A	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	✓	✓	N/A	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	✓	✓	N/A	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	✓	N/A	N/A	IER1F.IEN1	IPR249	—
—	Reserved	250	03E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	251	03ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	252	03F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	253	03F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	254	03F8h	—	N/A	N/A	N/A	—	—	—

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	sstb Return	IER	IPR	DTCER
—	Reserved	255	03FCh	—	N/A	N/A	N/A	—	—	—

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.

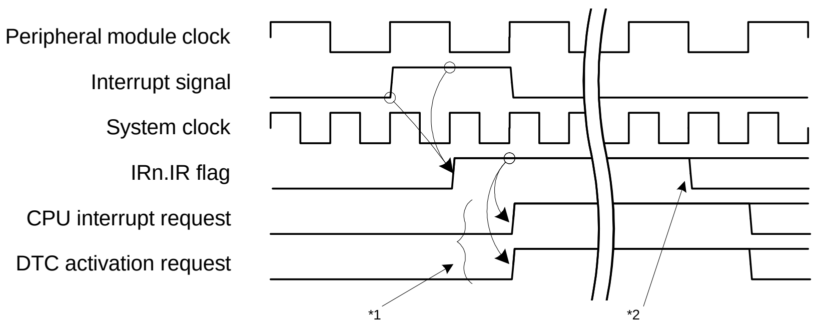
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt or DTC activation)
- Determining priority

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 5) 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 (n = interrupt vector number) 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 DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DTC transfer settings and transfer count. For details, see Table 14.4, Operation at DTC Activation.



Note 1. One of the following requests is issued: CPU interrupt request, DTC 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, see Table 14.4, Operation at DTC Activation.

Figure 14.2 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 14.3 to Figure 14.5 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 between issuance of continuous interrupt requests.

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 14.3 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 25, Serial Communications Interface (SCIg), section 26, I^2C-bus Interface (RIICa), and section 27, Serial Peripheral Interface (RSPIa).

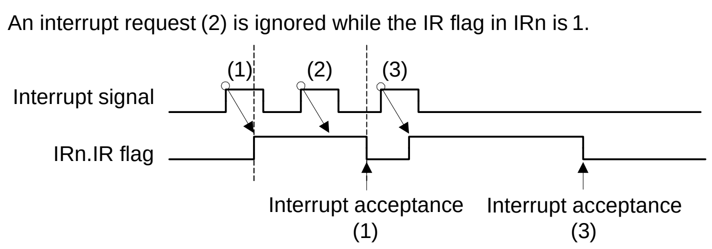


Figure 14.3 Timing for IRn.IR Flag Re-Setting

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 14.4 shows operation when the interrupt is disabled.

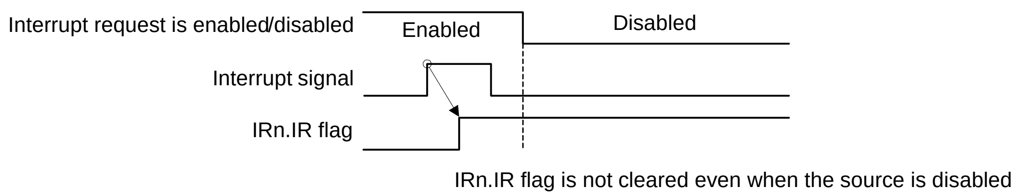


Figure 14.4 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 14.5 shows the operation of the interrupt status flag (IR flag) in IRn (n = interrupt vector number) 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.

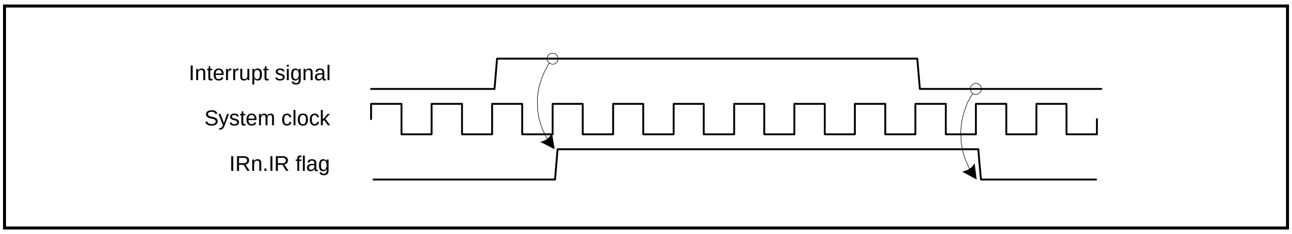


Figure 14.5 IRn.IR Flag Operation for Level Detection Interrupts

Figure 14.6 shows the procedure for handling level detection interrupts.

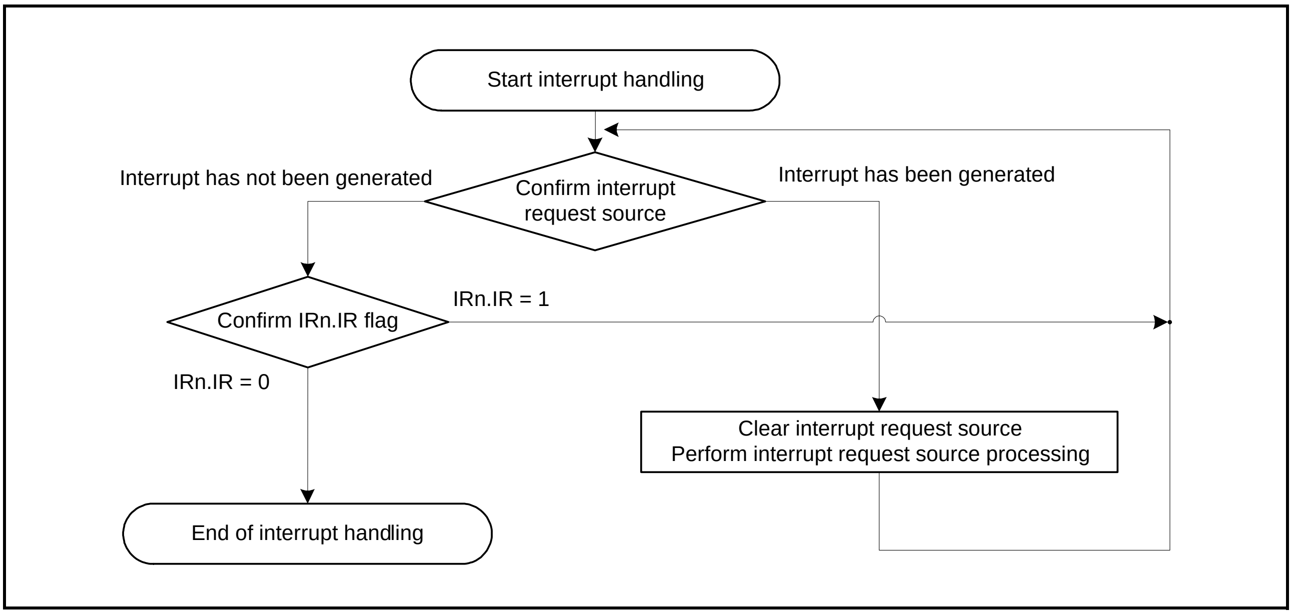


Figure 14.6 Procedure for Handling Level Detection Interrupts

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 (m = 02h to 1Fh, j = 0 to 7)

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag (n = interrupt vector number) 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 25, Serial Communications Interface (SCIg), section 26, I^2C-bus Interface (RIICa), and section 27, Serial Peripheral Interface (RSPIa).

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 "✓" in Table 14.3.
If the DTC is selected as the destination for requests from an IRQi pin (i = 0 to 5), 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) DTC Activation

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7) is 0.

1. Set the DTC activation enable bit in the DTC activation enable register (DTCERn.DTCE (n = interrupt vector number)) for the pertinent source to 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).

(2) CPU Interrupt Request

If the interrupt request destination is the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7) to 1 while the DTC activation settings described above are in place.

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

Table 14.4 Operation at DTC Activation

Interrupt Request Destination	DISEL *1	Remaining Number of Transfer Operations	Operation per Request	IR*2	Interrupt Request Destination after Transfer
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 DTCERn.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	Cleared on interrupt acceptance by the CPU	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.

Note 1. DISEL for the DTC is set by the DTC.MRB.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC activation request) that is generated again will be ignored.
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 DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit (n = interrupt vector number) has not been cleared) after the settings described under (1) 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 (1) 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 (n = interrupt vector number) have no effect. An interrupt source with a smaller vector number takes precedence.

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 IPR[3:0] bits in IPRn (n = interrupt vector number). 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 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 5) 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 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 14.7 shows an example of digital filter operation.

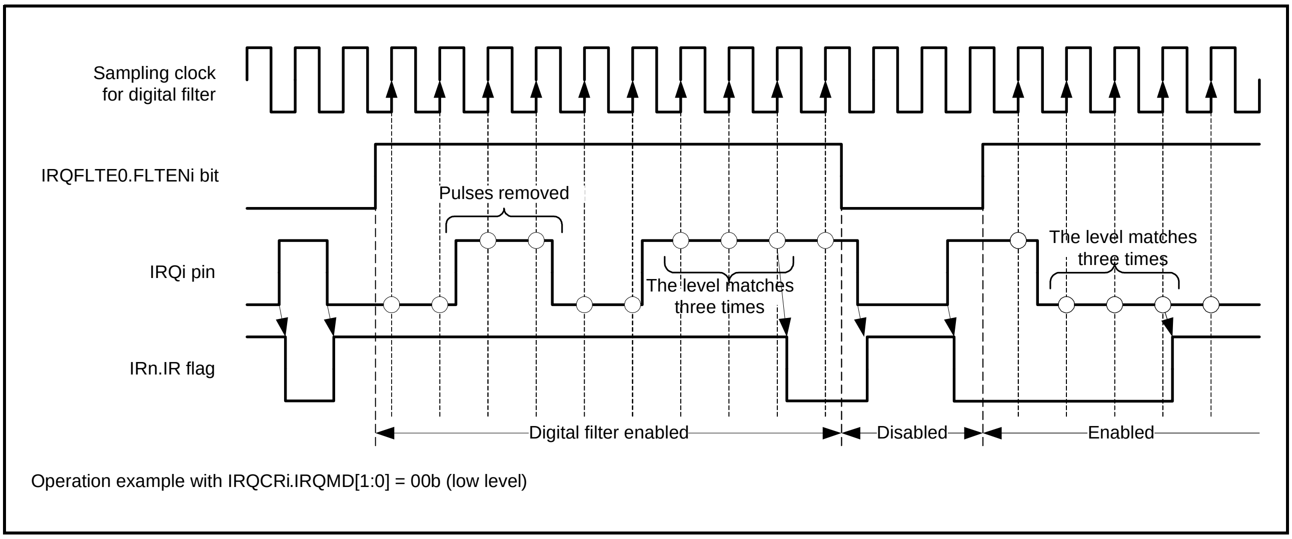


Figure 14.7 Digital Filter Operation Example

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 (m = 02h to 1Fh, j = 0 to 7) to 0 (interrupt request disabled).
2. Clear the IRQFLTE0.FLTENi bit (i = 0 to 5) to 0 (digital filter disabled).*1
3. Set the digital filter sampling clock with the IRQFLTC0.FCLKSELi[1:0] bits.*1
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 (n = interrupt vector number) to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).*1
8. If the interrupt is to be used for DTC activation, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if the setting not is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

Note 1. To use the digital filter function, settings must be made beforehand.

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 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, before ending the handler.

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).*1
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.*1
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).*1
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

Note 1. To use the digital filter function, settings must be made beforehand.

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, deep sleep 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 or Deep 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 (m = 02h to 1Fh, j = 0 to 7) 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 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 (m = 02h to 1Fh, j = 0 to 7) 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 (n = interrupt vector number)) 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 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, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 2	• Connected to peripheral modules (modules other than those connected to internal peripheral buses 1, 3, and 4) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 3	• Connected to peripheral modules (CMPC) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (MTU3) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 6	• Connected to the flash control module • Operates in synchronization with the FlashIF clock (FCLK)

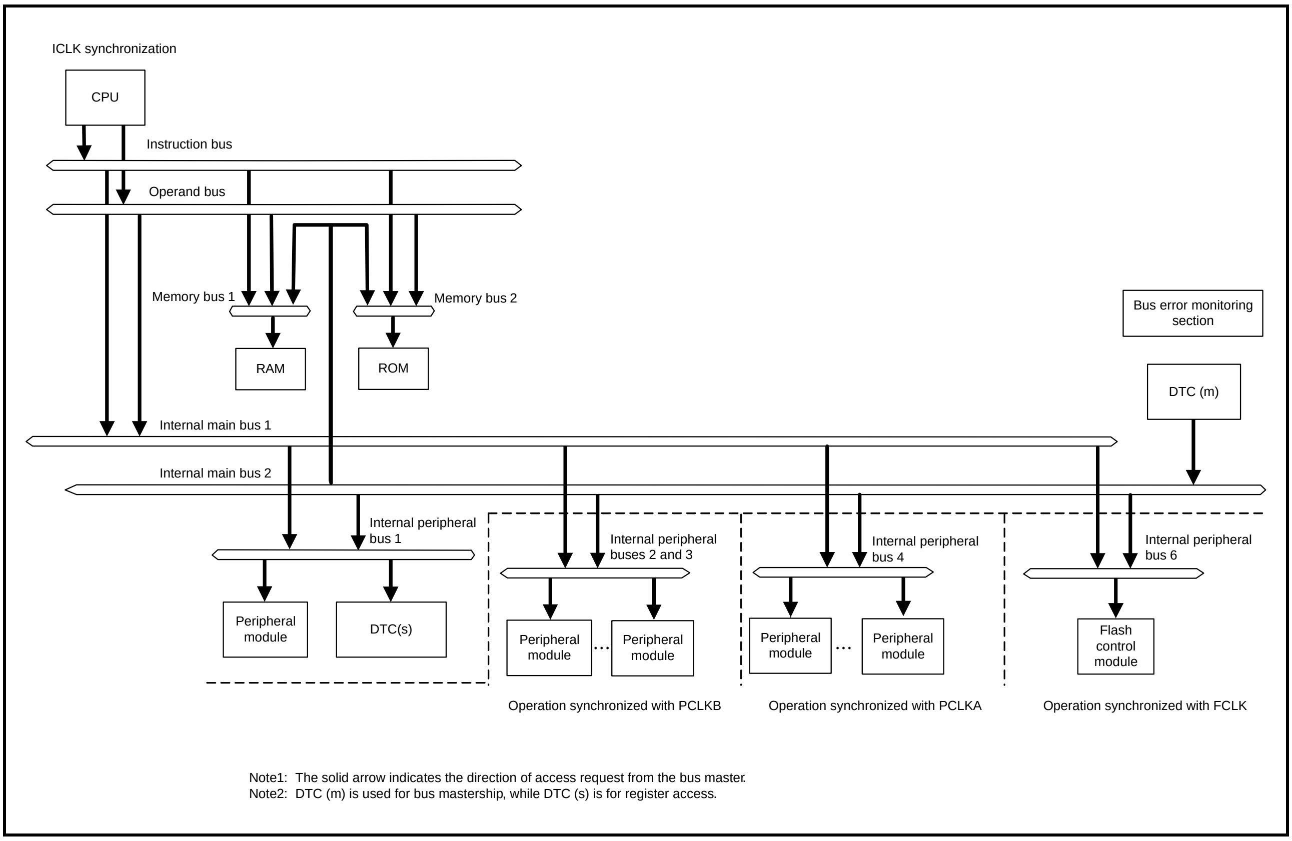


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 27FFh	Memory bus 1	RAM
0000 2800h to 0000 3FFFh		Reserved area
0000 4000h to 0000 4A7Fh		RAM
0000 4A80h to 0007 FFFFh		Reserved area
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
000C 0000h to 000D FFFFh	Internal peripheral bus 4	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Flash control module
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. The instruction bus is 64 bits while the operand bus is 32 bits.
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.

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. The memory buses are 64 bits. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of the buses can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (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 (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.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 4), 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	Internal main buses	Bus Master
High ↑	2	DTC
Low	1	CPU

Note: The above applies when the priority order of the buses is fixed.
The priority order of internal main bus 1 and another bus (internal main bus 2) can be toggled by the bus priority control register (BUSPRI) (round-robin method).

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, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, and 4
Internal peripheral bus 3	CMPC
Internal peripheral bus 4	MTU3
Internal peripheral bus 6	Flash control module

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 4, 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]), internal peripheral bus 4 priority control bits (BUSPRI.BPHB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted (round-robin method).
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 15.2).

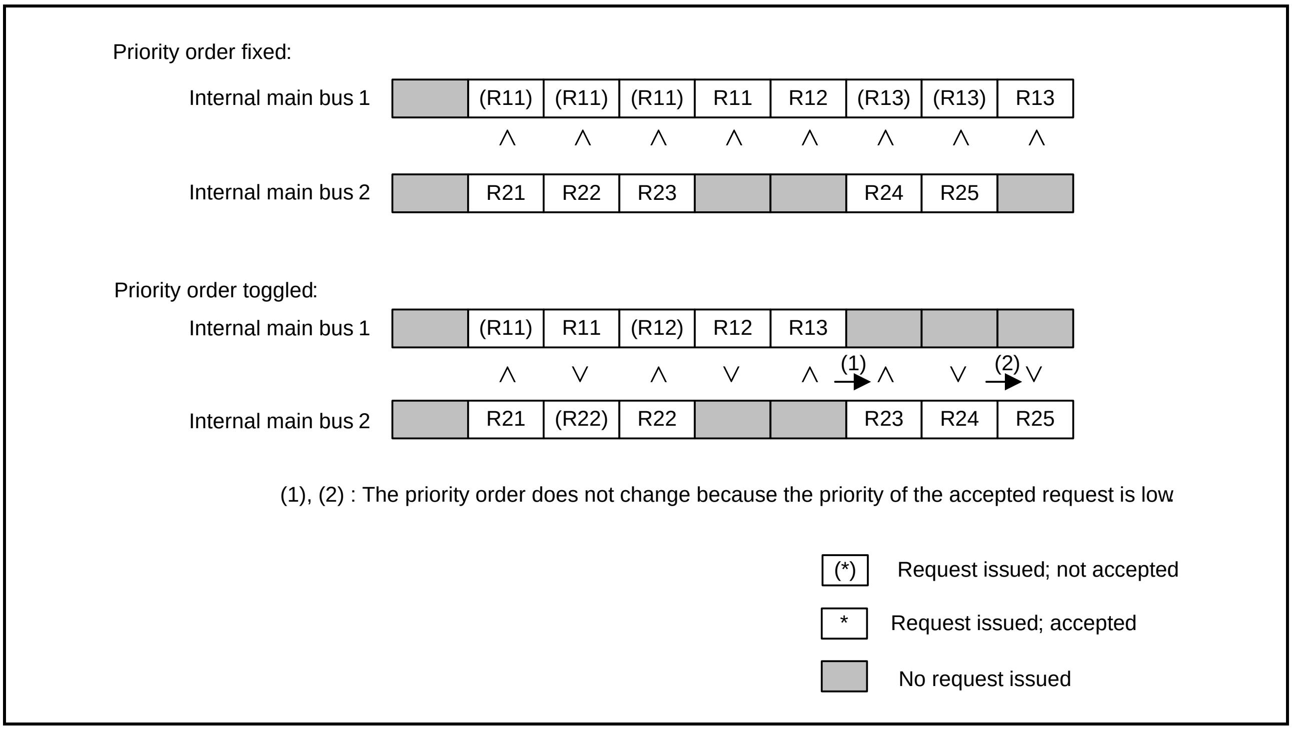


Figure 15.2 Priority Order between Internal Peripheral Bus Accesses

15.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When 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).

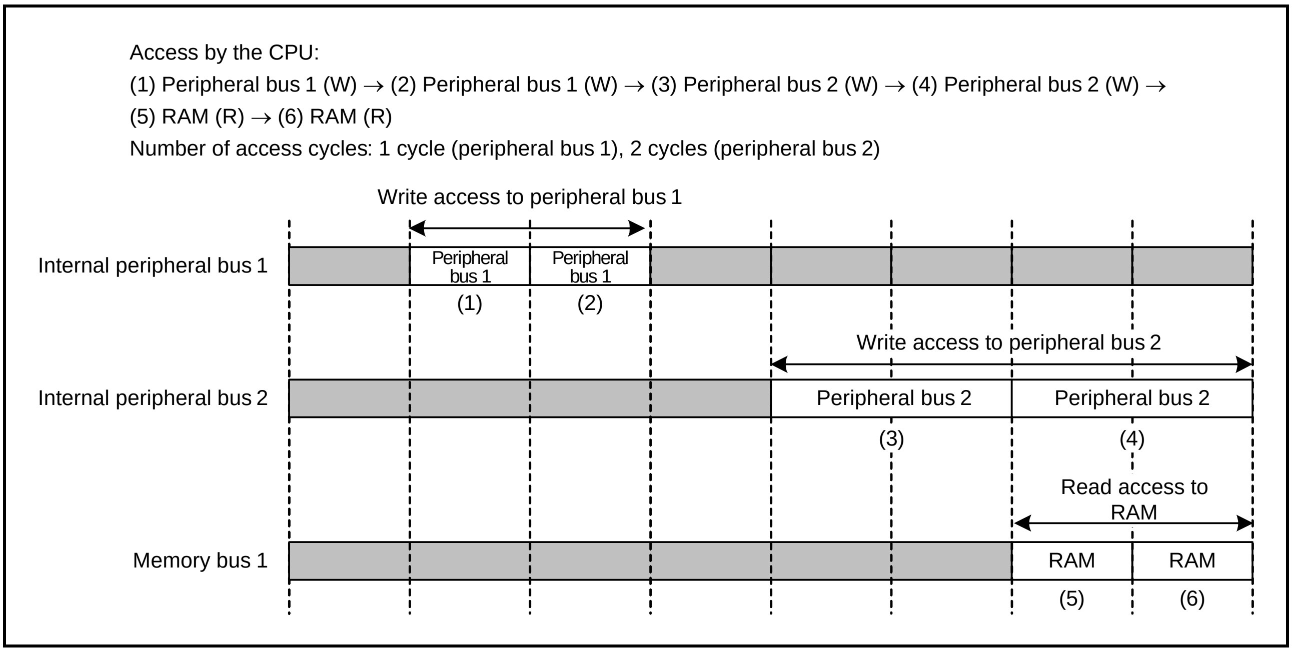


Figure 15.3 Write Buffer Function

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 DTC is able to handle transfer between peripheral buses at the same time.

An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DTC simultaneously employs internal main bus 2 for access to peripheral buses during access to RAM and ROM by the CPU.

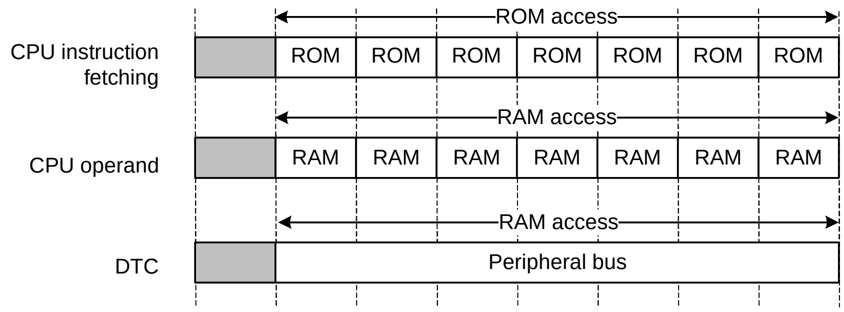


Figure 15.4 Example of Parallel Operations

15.2.7 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

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

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

15.3 Register Descriptions

15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

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

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

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

STSCLR Bit (Status Clear)

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

15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

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

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1,*2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not clear the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

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

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

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

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

15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

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

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

15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BPFB[1:0]		BPHB[1:0]		BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (RAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (ROM) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Bus 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b9, b8	BPHB[1:0]	Internal Peripheral Bus 4 Priority Control	b9 b8 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC 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 internal main bus 1.
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 internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

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

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

BPGB[1:0] Bits (Internal Peripheral Bus 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPHB[1:0] Bits (Internal Peripheral Bus 4 Priority Control)

These bits specify the priority order for internal peripheral bus 4.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

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

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

15.4 Bus Error Monitoring Section

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

15.4.1 Types of Bus Error

There are two types of bus error: illegal address access and timeout.
Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

15.4.1.1 Illegal Address Access

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

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

15.4.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKA cycles.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles. In this MCU, a timeout error does not occur.

15.4.2 Operations When a Bus Error Occurs

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

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

15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the types of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected on the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until BERSRn is cleared.

Table 15.5 Types of Bus Errors

Address	Type of Area	Type of Error
		Illegal Address Access
0000 0000h to 0007 FFFFh	Memory bus 1	—
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	—
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	Δ
000A 0000h to 000B FFFFh	Internal peripheral bus 3	Δ
000C 0000h to 000D FFFFh	Internal peripheral bus 4	Δ
000E 0000h to 000F FFFFh	Reserved area	—
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Δ
0100 0000h to 0FFF FFFFh	Reserved area	—
1000 0000h to 7FFF FFFFh	Reserved area	○
8000 0000h to FFFF FFFFh	Memory bus 2	—

—: A bus error does not result.
Δ: A bus error may or may not result.
○: A bus error results.
Note: The capacity of the RAM and ROM differs depending on the product. For details, see section 33, RAM, section 34, Flash Memory.

15.5 Interrupt

15.5.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 15.6 Interrupt Source

Name	Interrupt Source	DTC Activation
BUSERR	Illegal address access error or timeout	Not possible

16. Memory-Protection Unit (MPU)

16.1 Overview

The RXv2 CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).
Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.
The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.
Table 16.1 lists the specifications of the memory-protection unit, and Figure 16.1 shows a block diagram of the memory-protection unit.

Table 16.1 Specifications of Memory Protection

Specifictaions	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

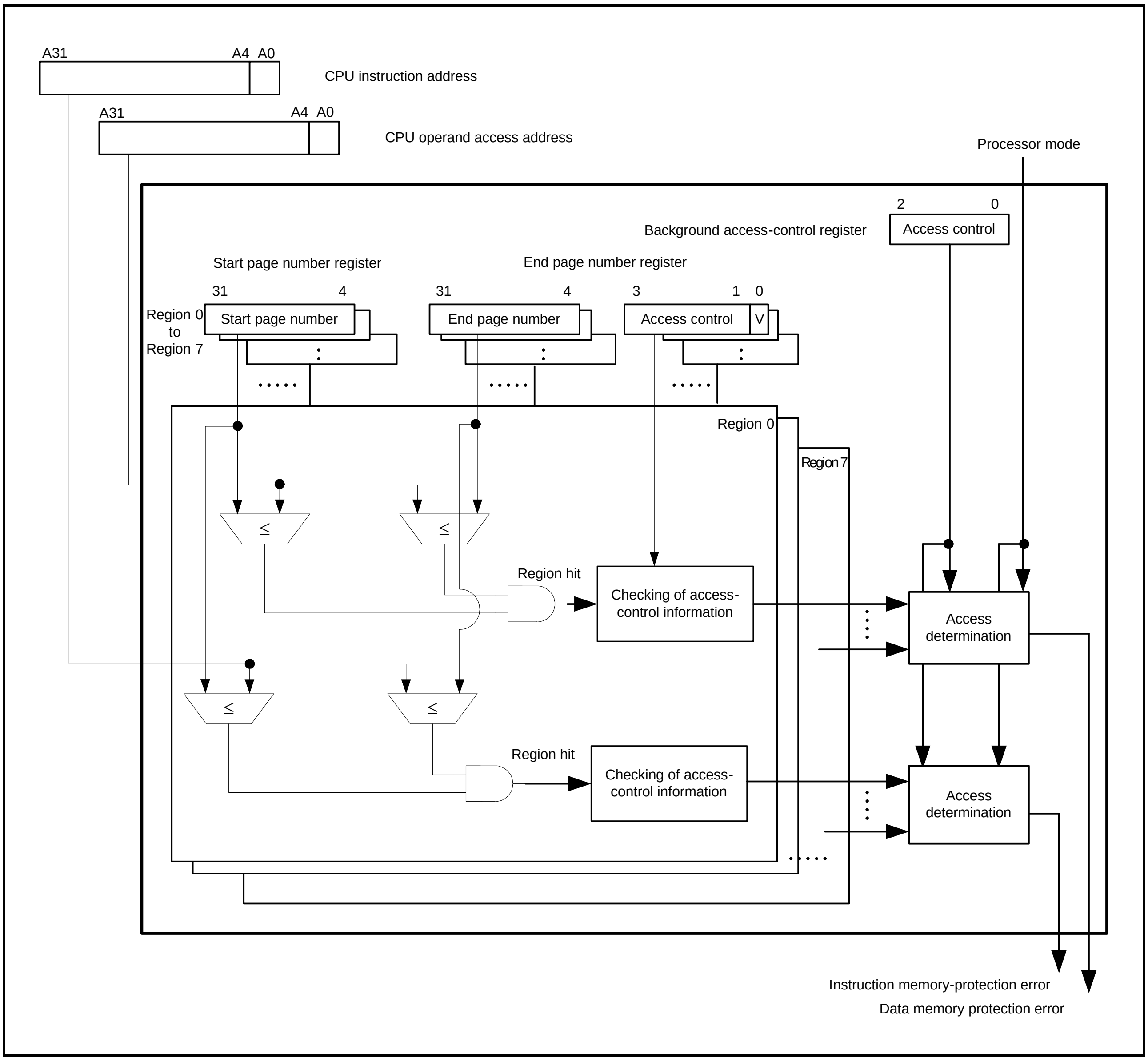


Figure 16.1 Block Diagram of the Memory-Protection Unit

16.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

16.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.
The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

16.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

16.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

16.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

16.2 Register Descriptions

16.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h,
RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[27:0]												—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should always be 0	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

16.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch, REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[27:0]												UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.
This bit is cleared to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.
Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

16.2.3 Memory-Protection Enable Register (MPEN)

Address(es): 0008 6500h

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

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

Bit	Symbol	Bit Name	Function	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled 0: The memory protection is disabled	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE or RTFI) that shifts operation to the user mode.

16.2.4 Background Access Control Register (MPBAC)

Address(es): 0008 6504h

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

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

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should always be 0	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

16.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address(es): 0008 6508h

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

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

Bit	Symbol	Bit Name	Function	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DMPER and IMPER bits in MPESTS are cleared to 0	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generation bit (DMPER), and the instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

16.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): 0008 650Ch

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

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

Bit	Symbol	Bit Name	Function	R/W
b0	IMPER	Instruction Memory-Protection Error Generation	0: No instruction memory-protection error was generated 1: Instruction memory-protection error was generated	R
b1	DMPER	Data Memory-Protection Error Generation	0: No data memory-protection error was generated 1: Data memory-protection error was generated	R
b2	DRW	Data Read/Write	0: Data were read 1: Data were written	R
b31 to b3	—	Reserved	The read value is 0. The write value should always be 0	R/W

IMPER Bit (Instruction Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by instruction execution.
Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DMPER Bit (Data Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by operand access.
Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DRW Bit (Data Read/Write)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the DMPER bit is 1.
Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

16.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): 0008 6514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

16.2.8 Region Search Address Register (MPSA)

Address(es): 0008 6520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

16.2.9 Region Search Operation Register (MPOPS)

Address(es): 0008 6524h

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

Bit	Symbol	Bit Name	Function	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD).

Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

16.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): 0008 6526h

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

Bit	Symbol	Bit Name	Function	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done 1: All access-controlled areas are invalidated	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is cleared to 0, all settings other than background access-control settings are invalid.

16.2.11 Instruction-Hit Region Register (MHITI)

Address(es): 0008 6528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7 1: Instruction memory-protection error was generated in region 7 b22 0: Instruction memory-protection error was not generated in region 6 1: Instruction memory-protection error was generated in region 6 b21 0: Instruction memory-protection error was not generated in region 5 1: Instruction memory-protection error was generated in region 5 b20 0: Instruction memory-protection error was not generated in region 4 1: Instruction memory-protection error was generated in region 4 b19 0: Instruction memory-protection error was not generated in region 3 1: Instruction memory-protection error was generated in region 3 b18 0: Instruction memory-protection error was not generated in region 2 1: Instruction memory-protection error was generated in region 2 b17 0: Instruction memory-protection error was not generated in region 1 1: Instruction memory-protection error was generated in region 1 b16 0: Instruction memory-protection error was not generated in region 0 1: Instruction memory-protection error was generated in region 0	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.
If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

16.2.12 Data-Hit Region Register (MHITD)

Address(es): 0008 652Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7 1: A data memory-protection error or search hit was generated in region 7 b22 0: Neither a data memory-protection error nor a search hit was generated in region 6 1: A data memory-protection error or search hit was generated in region 6 b21 0: Neither a data memory-protection error nor a search hit was generated in region 5 1: A data memory-protection error or search hit was generated in region 5 b20 0: Neither a data memory-protection error nor a search hit was generated in region 4 1: A data memory-protection error or search hit was generated in region 4 b19 0: Neither a data memory-protection error nor a search hit was generated in region 3 1: A data memory-protection error or search hit was generated in region 3 b18 0: Neither a data memory-protection error nor a search hit was generated in region 2 1: A data memory-protection error or search hit was generated in region 2 b17 0: Neither a data memory-protection error nor a search hit was generated in region 1 1: A data memory-protection error or search hit was generated in region 1 b16 0: Neither a data memory-protection error nor a search hit was generated in region 0 1: A data memory-protection error or search hit was generated in region 0	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.

When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is cleared to 0000 0000h.

16.3 Functions

16.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.
Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.
An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

16.3.2 Region Search

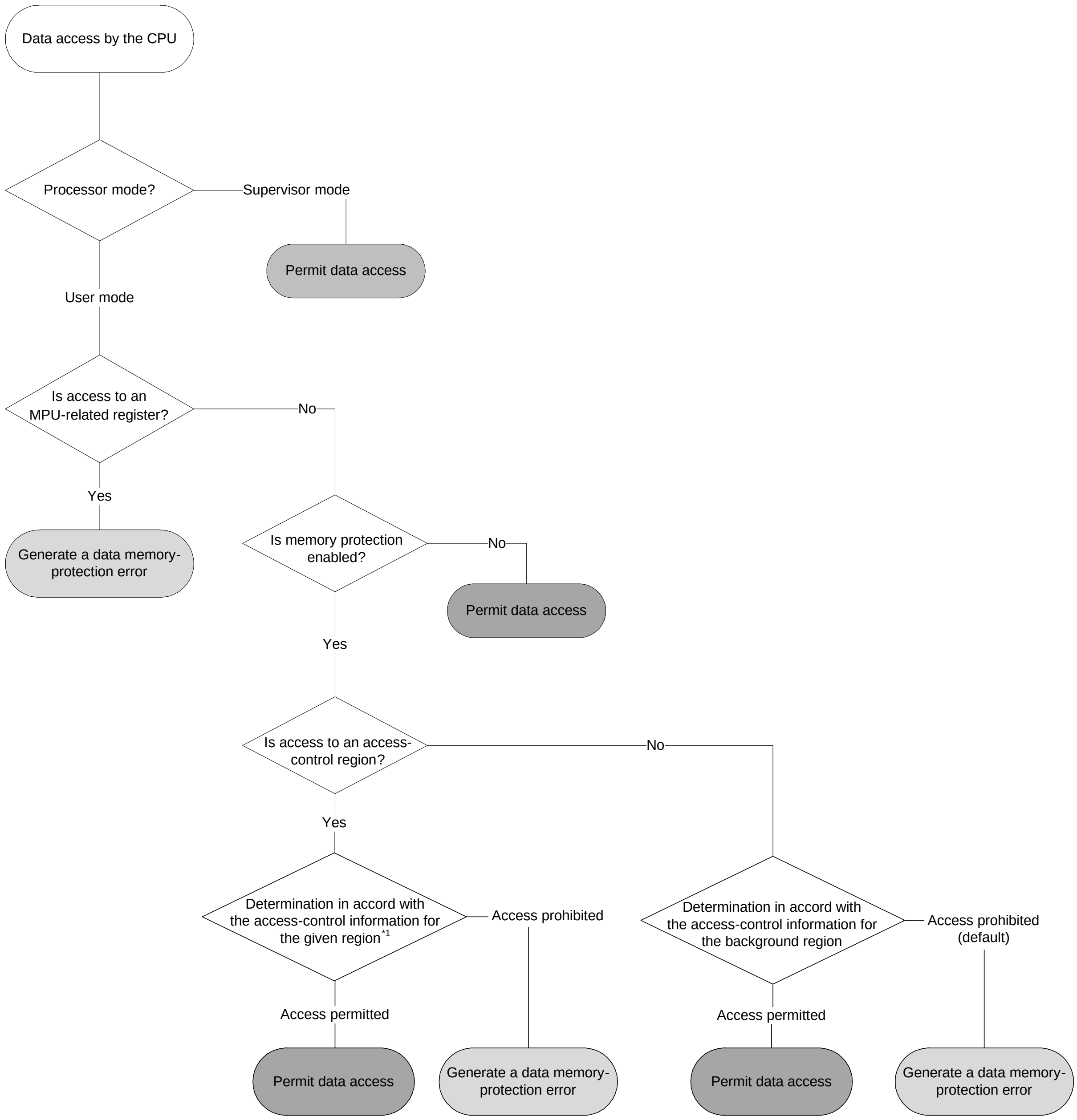
Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOP) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

16.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

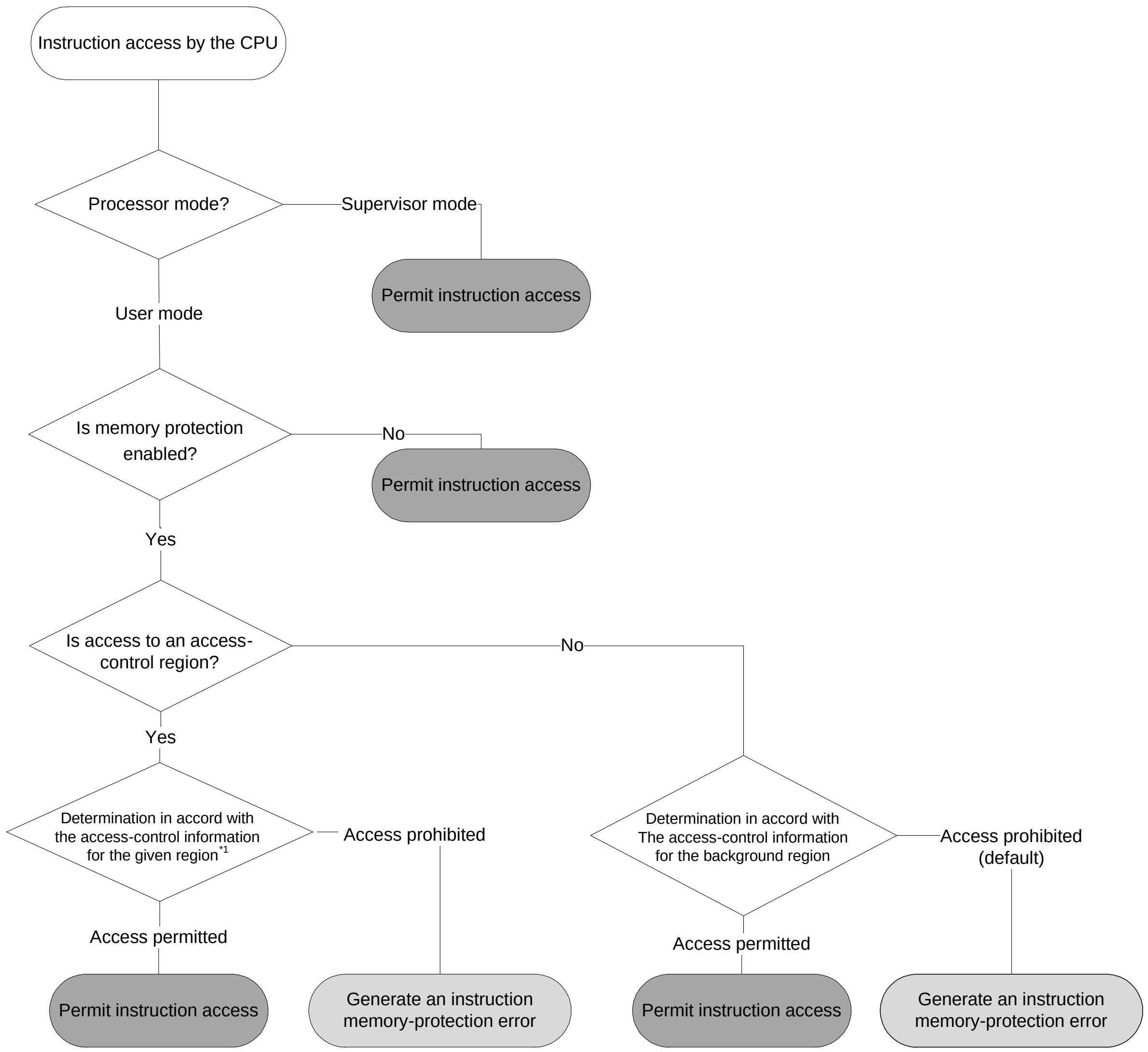
16.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 16.2 shows the flow of determination in the case of data access and Figure 16.3 shows the flow of determination in the case of instruction access.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are different for the given regions (this includes the background region).

Figure 16.2 Flow of Determination for Data Access



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are not the same for the given regions (this includes the background region).

Figure 16.3 Flow of Determination for Instruction Access

16.4 Procedures for Using Memory Protection

16.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.
Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.
Settings for the background access-control region are made in the background access-control register (MPBAC).

16.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

16.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, please be sure to read any of these registers and check that the settings have been made and execute an operation using the value read before the transition to user mode.
Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

16.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 13, Exception Handling.
To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generation (IMPER) and data memory-protection error generation (DMPER) bits in the memory-protection error status register (MPESTS) from within the exception-processing routine.
After confirming the type of error, clear the memory-protection error status register (MPESTS) by writing 1 to the status clearing (MPE) bit in the memory-protection error status clearing register (MPECLR).

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical "or" of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITD.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical "or" of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

17. Data Transfer Controller (DTCa)

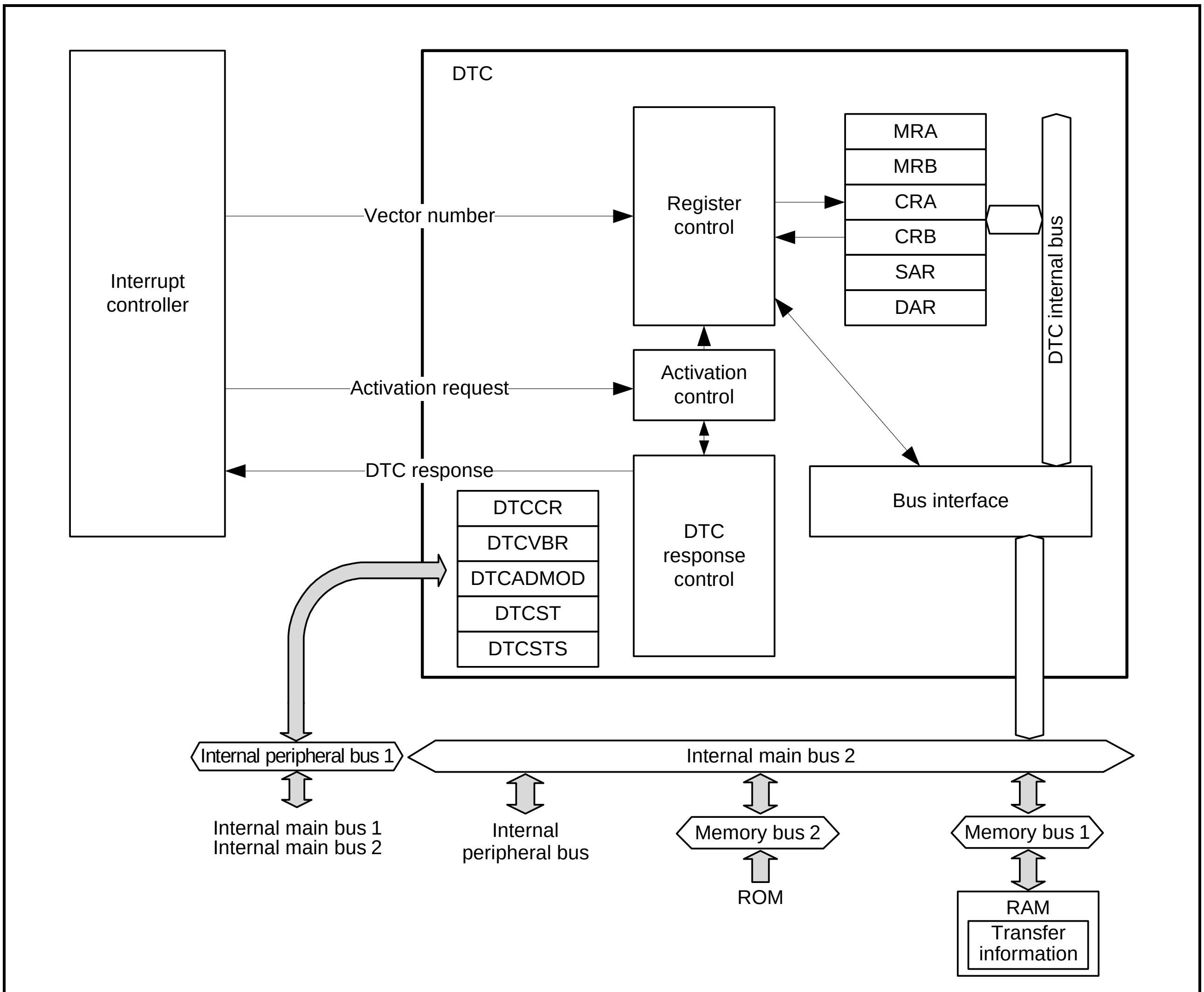
This MCU incorporates a data transfer controller (DTC).
The DTC is activated by an interrupt request to perform data transfers.

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 modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer corresponding to the interrupt source is possible (transferred by the DTC activation request from the ICU). • Multiple data can be transferred on a single activation source (chain transfer). • Either “executed when the counter is 0” or “always executed” can be selected for chain transfer.
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Read skip	Transfer information read skip can be executed.
Write-back skip	When “fixed” is selected for transfer source address or transfer destination address, write-back skip is executed.
Low power consumption function	Module stop state can be set.



MRA:	DTC mode register A	DTCCR:	DTC control register
MRB:	DTC mode register B	DTCVBR:	DTC vector base register
CRA:	DTC transfer count register A	DTCADMOD:	DTC address mode register
CRB:	DTC transfer count register B	DTCST:	DTC module start register
SAR:	DTC transfer source register	DTCSTS:	DTC status register
DAR:	DTC transfer destination register		

Figure 17.1 DTC Block Diagram

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 these DTC internal registers are placed in the RAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents are written back to the RAM area as transfer information.

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 (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register cannot be accessed directly from the CPU.

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 MRA.SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer. (–1 when SZ[1:0] bits = 00b, –2 when SZ[1:0] bits = 01b, –4 when MRA.SZ[1:0] bits = 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated when specified data transfer is completed. 1: An interrupt request to the CPU is generated each time DTC data transfer is performed.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed continuously. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

MRB register cannot be accessed directly from the CPU.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 17.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 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 register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

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 register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

17.2.5 DTC Transfer Count Register A (CRA)

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

• Normal transfer mode

	CRA															
	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

• Repeat transfer mode/block transfer mode

	CRAH								CRAL							
	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

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

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA register cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

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 register 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 register cannot be accessed directly from the CPU.

17.2.7 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous activation process.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.

17.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Bit Name	Description	R/W
b9 to b0	DTC Vector Base Address (Lower 10 bits)	These bits are read as 0. The write value should be 0.	R
b31 to b10	DTC Vector Base Address (Upper 22 bits)	Writing to the upper 4 bits (b31 to b28) is ignored, and the address of this register is extended by the value specified by b27.	R/W

DTCVBR register 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): DTC.DTCADMOD 0008 2408h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set	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 register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

17.2.10 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, deep sleep mode, or software standby mode. For details on transitions to the module stop state, deep sleep mode, and software standby mode, refer to section 17.8, Low Power Consumption Function, and section 11, Low Power Consumption.

17.2.11 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC-Activating Vector Number Monitoring	These bits indicate the vector number for the activation source when DTC transfer is in progress. The value is only valid if DTC transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activation source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer).

For the correspondence between the DTC activation sources and the vector addresses, refer to section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

ACT Flag (DTC Active Flag)

This flag indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC is completed in response to a transfer request.

17.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt as an activation source for the DTC.
For the correspondence between the DTC activation sources and the vector addresses, refer to section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For activation by software, refer to section 14.2.5, Software Interrupt Activation Register (SWINTR) in section 14, Interrupt Controller (ICUb).
Once the DTC has accepted an activation request, it does not accept another activation request until transfer for that single request is completed, regardless of the priority of the requests. When multiple activation requests are generated during DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple activation requests are generated while the DTC module start bit (DTCST.DTCST) is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the activation source is set to 0 at the start of data transfer.

17.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each activation source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. In the RAM area, the start address of the transfer information (n) with vector number n should be 4n added to the base address in the vector table.
Transfer information can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 17.2 shows the relationship between the DTC vector table and transfer information.
Figure 17.3 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 17.9.2, Allocating Transfer Information.

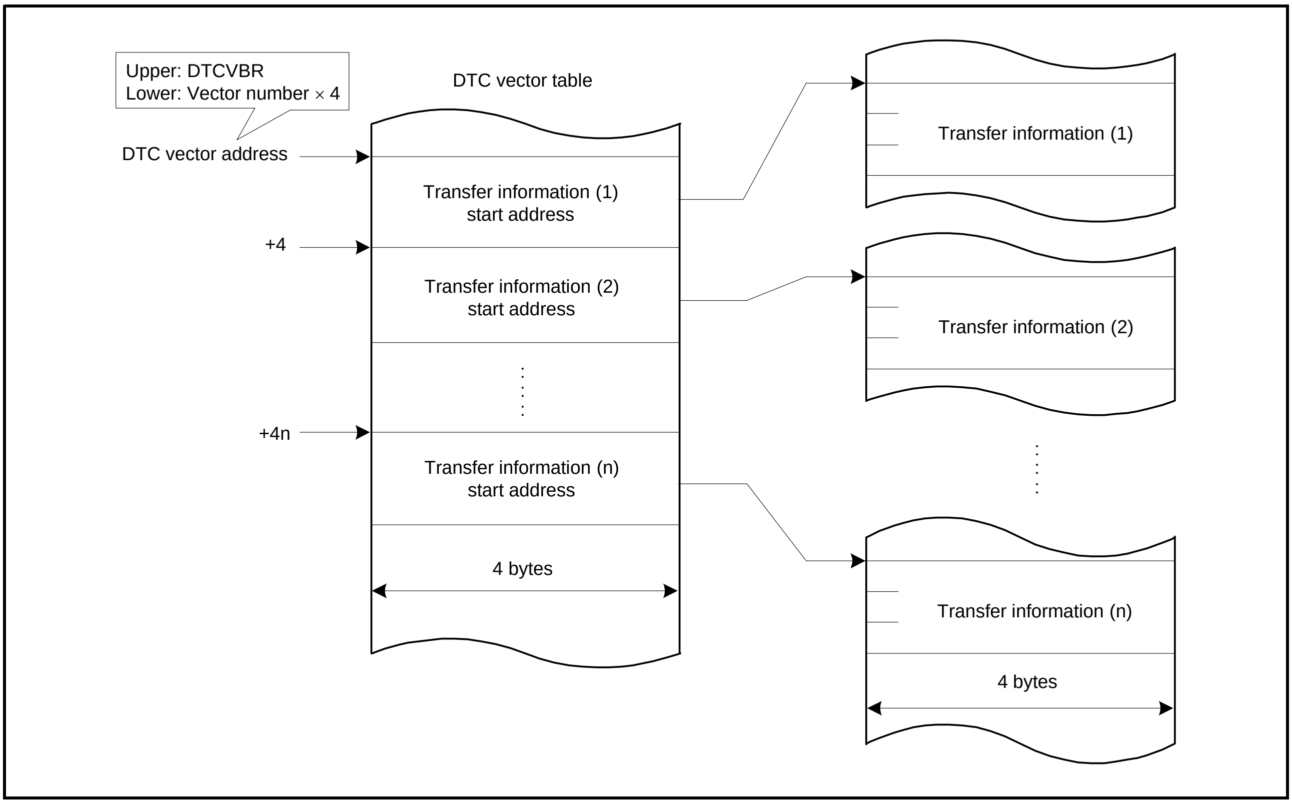


Figure 17.2 DTC Vector Table and Transfer Information

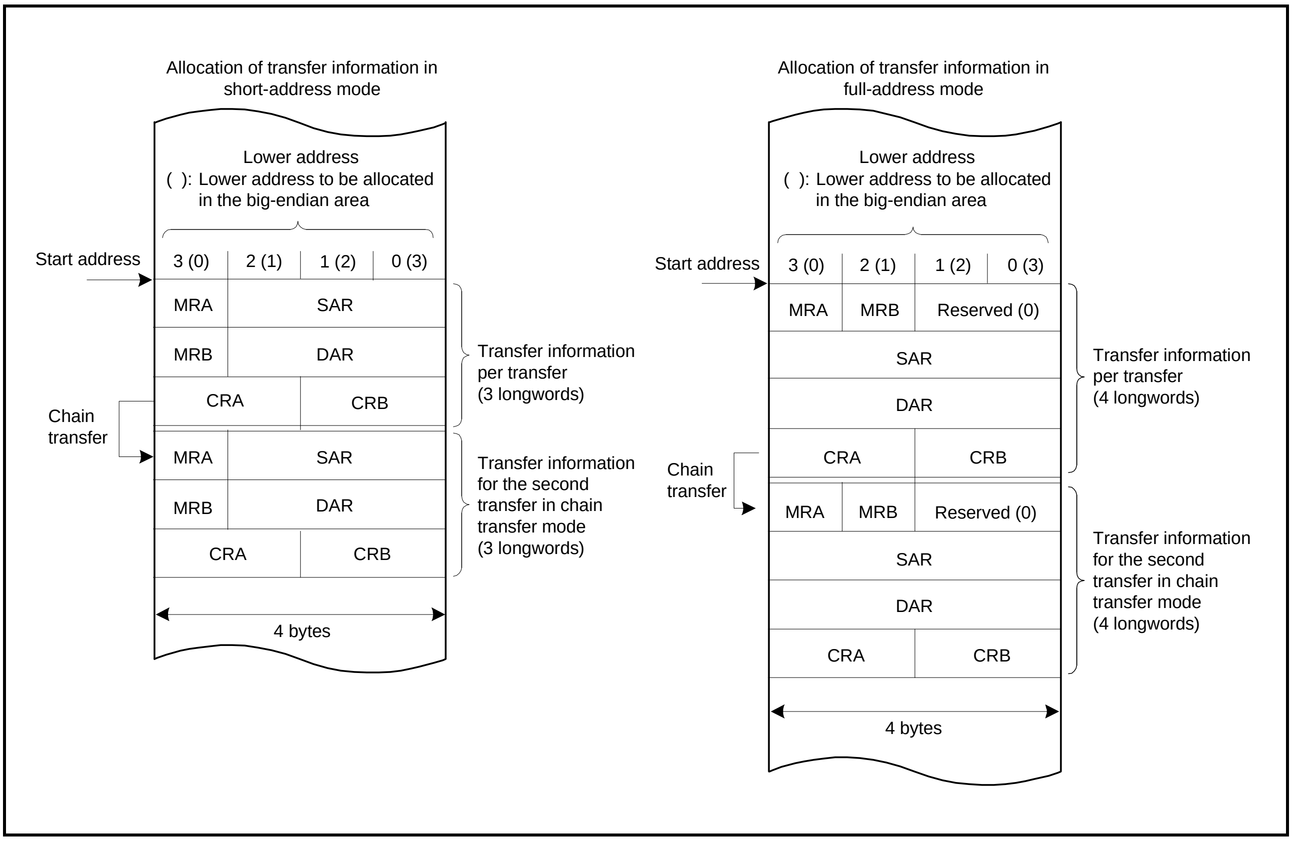


Figure 17.3 Allocation of Transfer Information in the RAM Area

17.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.
When the DTC is activated, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the transfer information store address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Storing transfer information in the RAM area allows data transfer of arbitrary number of channels.
There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.
The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.
Table 17.2 lists transfer modes of the DTC.

Table 17.2 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single activation source. Setting the MRB.CHNS bit also enables chain transfer when specified data transfer is completed.
Figure 17.4 shows the operation flowchart of the DTC. Table 17.3 lists chain transfer conditions.

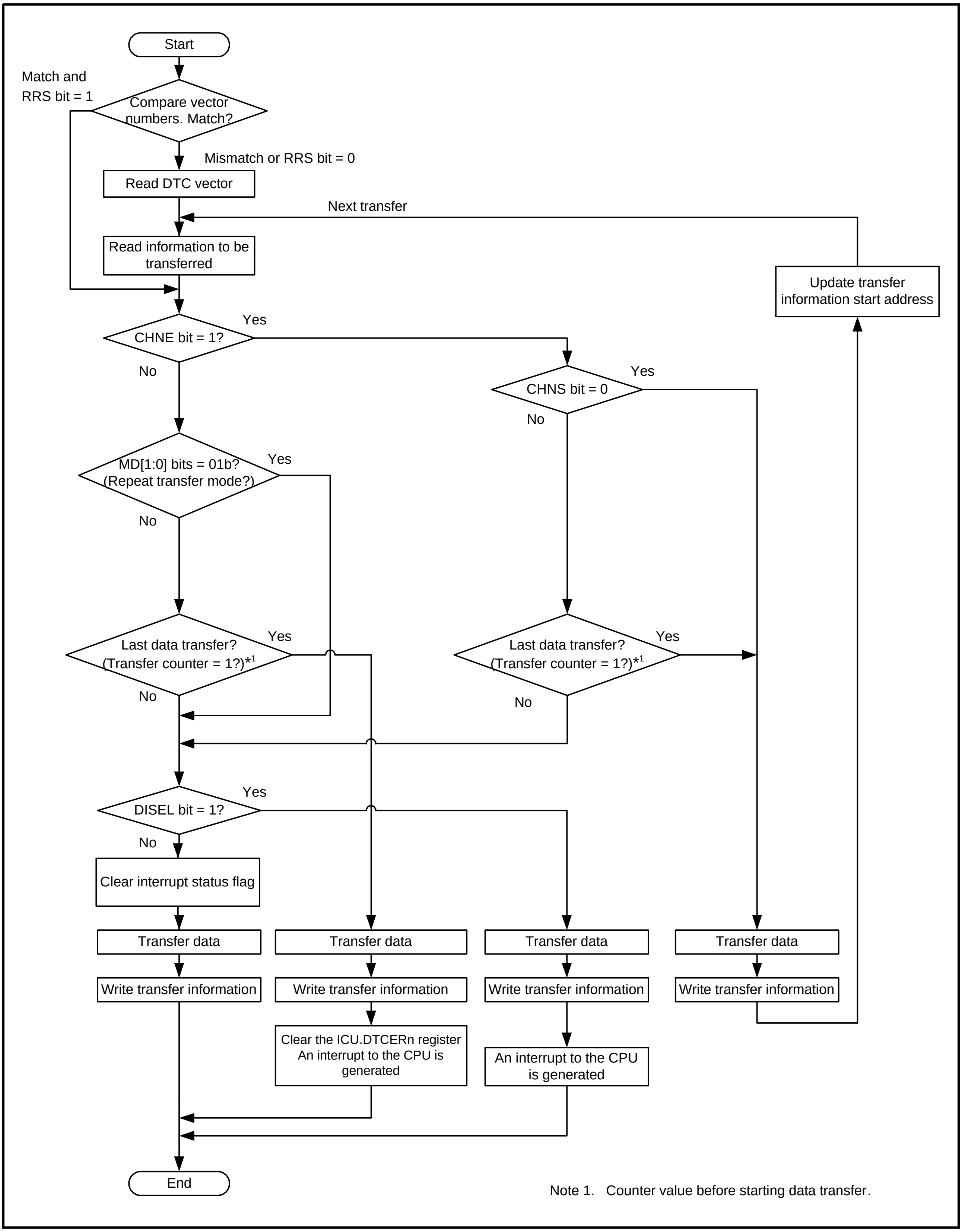


Figure 17.4 Operation Flowchart of the DTC

Table 17.3 Chain Transfer Conditions

First Transfer				Second Transfer*3				DTC Transfer
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	
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 Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by the setting of the DTCCR.RRS bit.
When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer was chain transfer, the vector address and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 17.13 shows an example of transfer information read skip.
To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. The retained vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

17.4.2 Transfer Information Write-Back Skip Function

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

The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are skipped.

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

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	SAR Register	DAR Register
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

17.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source. The transfer count can be set to 1 to 65536.
Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.
This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.
Table 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 Information
SAR	Transfer source address	Increment/decrement/fixed*1
DAR	Transfer destination address	Increment/decrement/fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

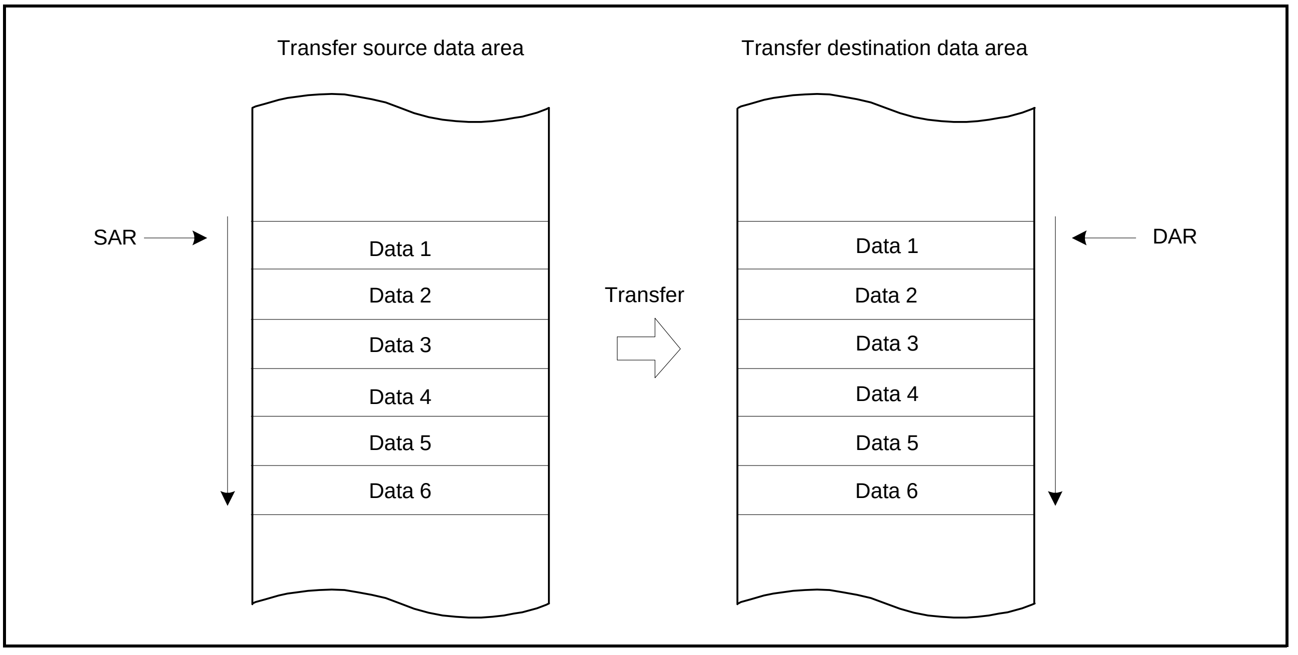


Figure 17.5 Memory Map of Normal Transfer Mode

17.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
Table 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 Information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment/decrement/fixed*1	(When the MRB.DTS bit is 0) Increment/decrement/fixed*1 (When the MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	(When the MRB.DTS bit is 0) DAR register initial value (When the MRB.DTS bit is 1) Increment/decrement/fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

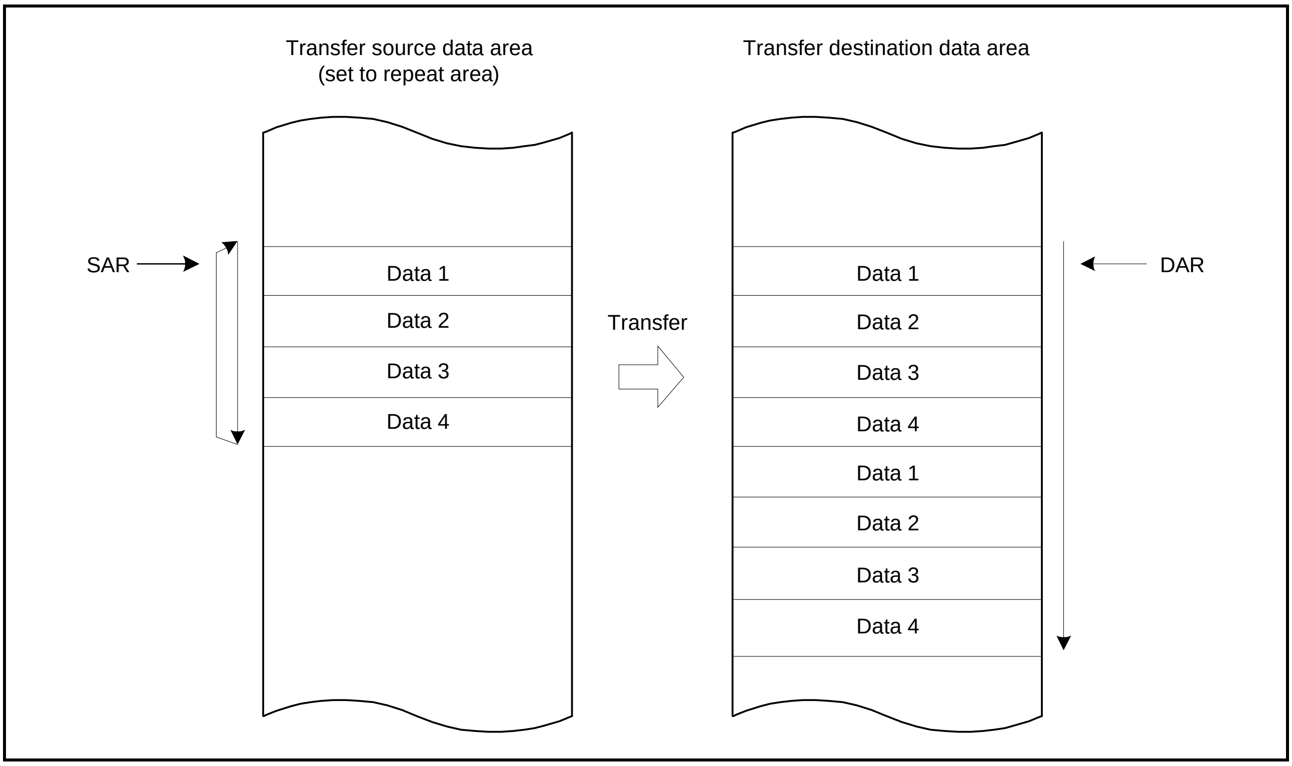


Figure 17.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

17.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single activation source.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit = 1 or the DAR register when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 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 Information
SAR	Transfer source address	(When MRB.DTS bit is 0) Increment/decrement/fixed*1 (When MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	(When MRB.DTS bit is 0) DAR register initial value (When MRB.DTS bit is 1) Increment/decrement/fixed*1
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

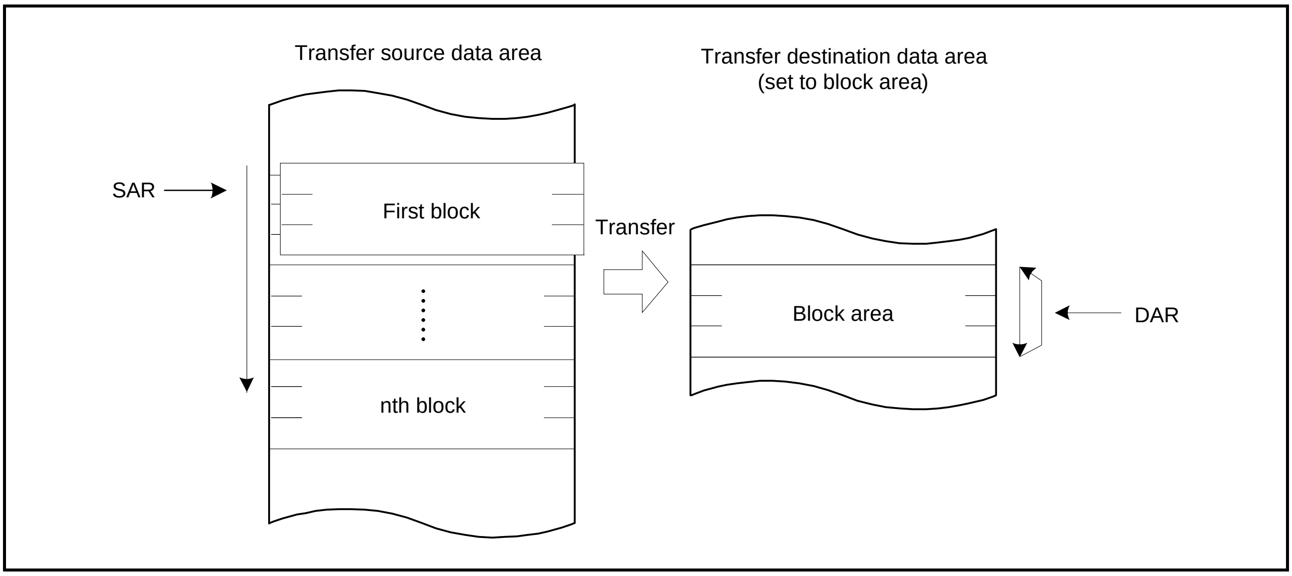


Figure 17.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

17.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source.
If the MRB.CHNE and CHNS bits are set to 1 and 0, respectively, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the MRB.DISEL bit to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed), and data transfer has no effect on the interrupt status flag of the activation source.
The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer.
Figure 17.8 shows chain transfer operation.

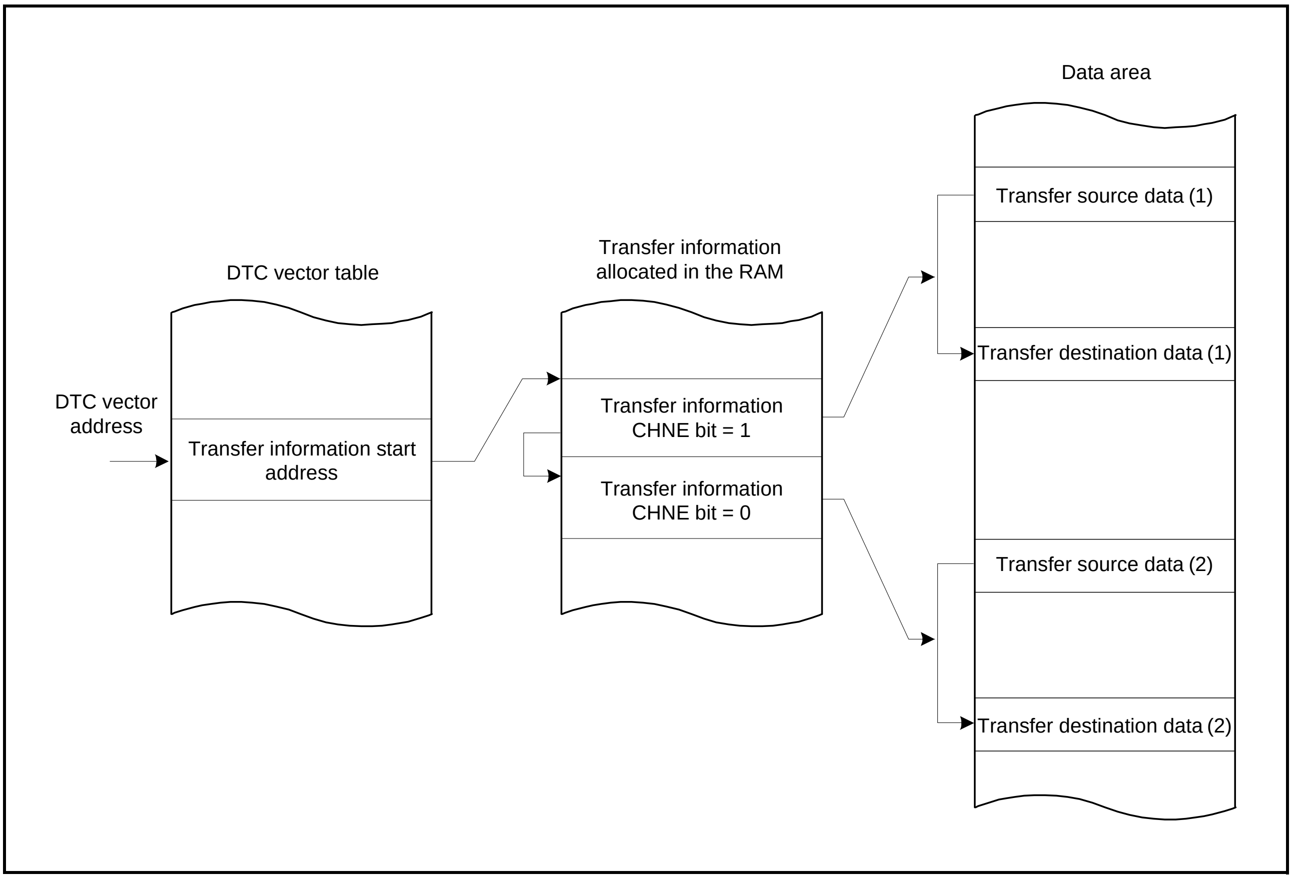


Figure 17.8 Chain Transfer Operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of specified data transfer. In repeat transfer mode, chain transfer is performed after completion of specified data transfer.
For details on chain transfer conditions, refer to Table 17.3, Chain Transfer Conditions.

17.4.7 Operation Timing

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

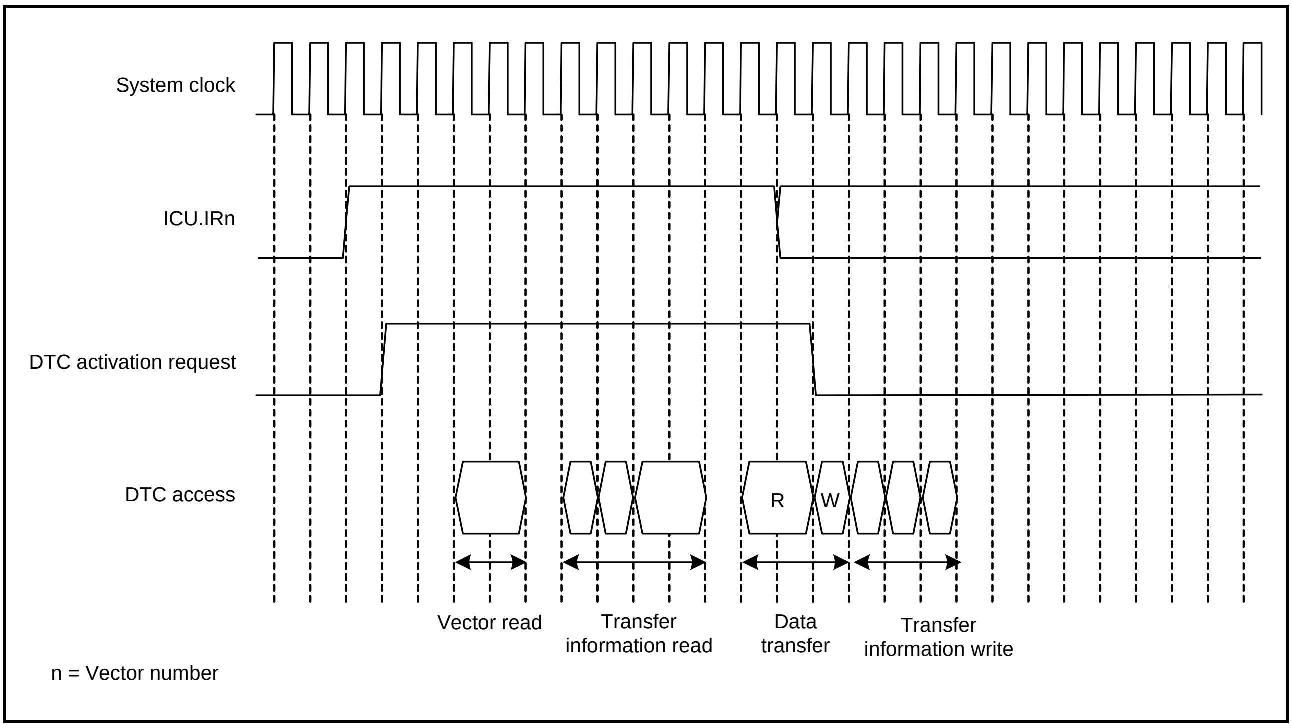


Figure 17.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

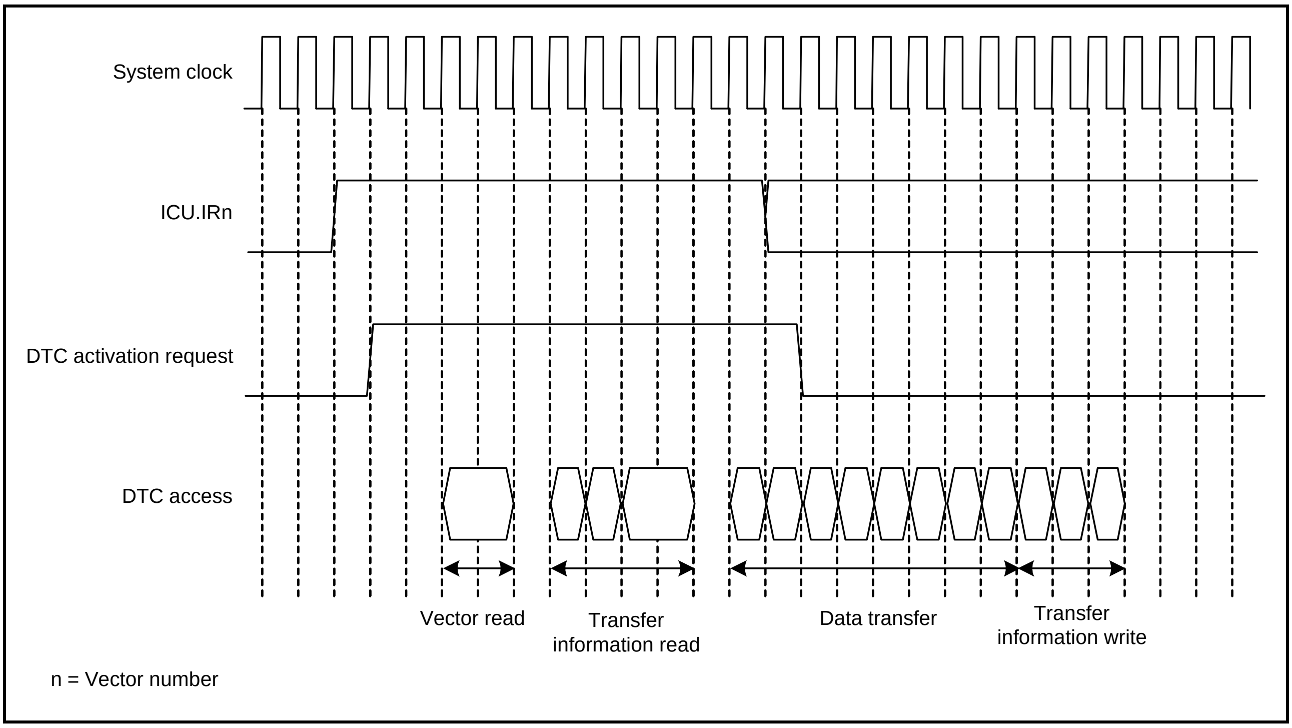


Figure 17.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

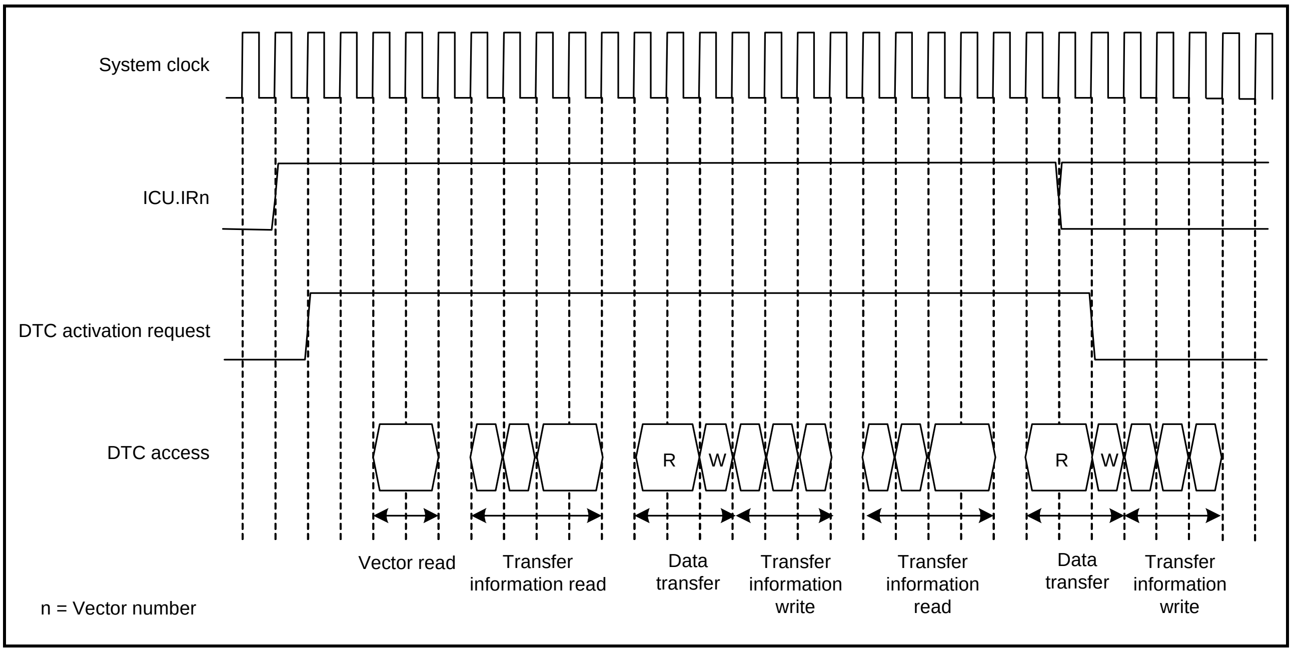


Figure 17.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

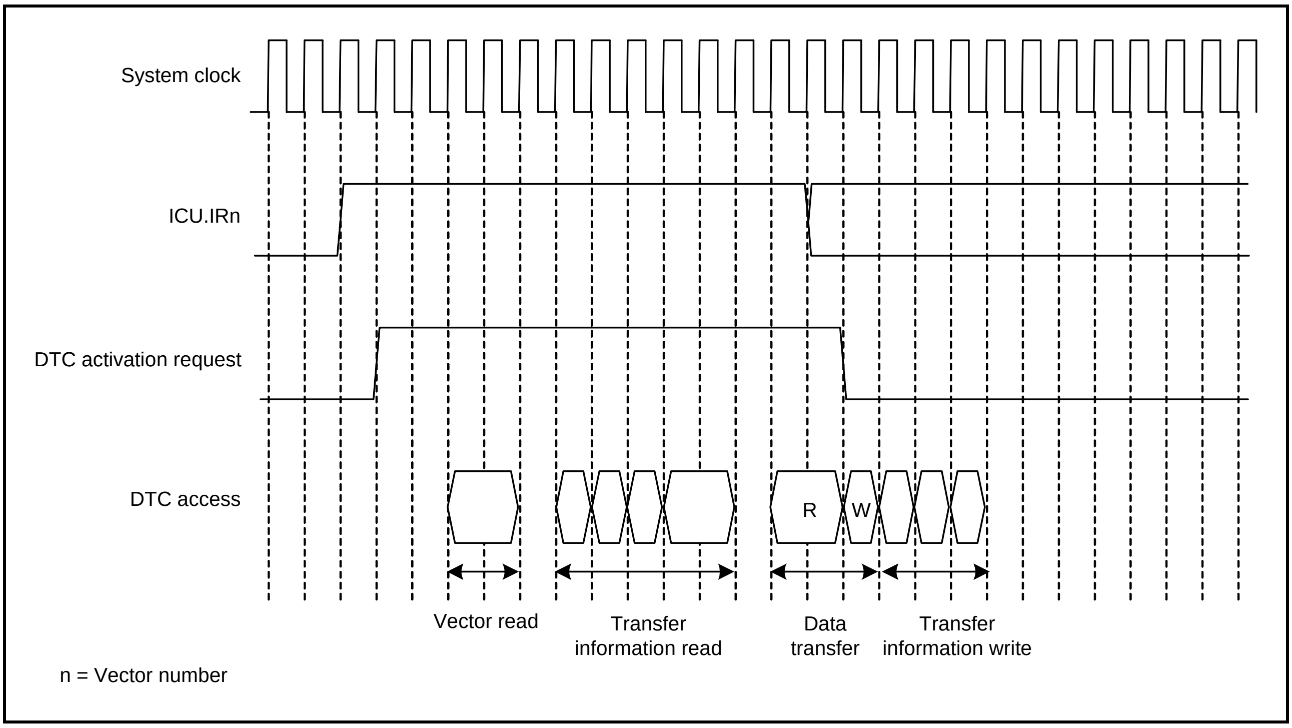
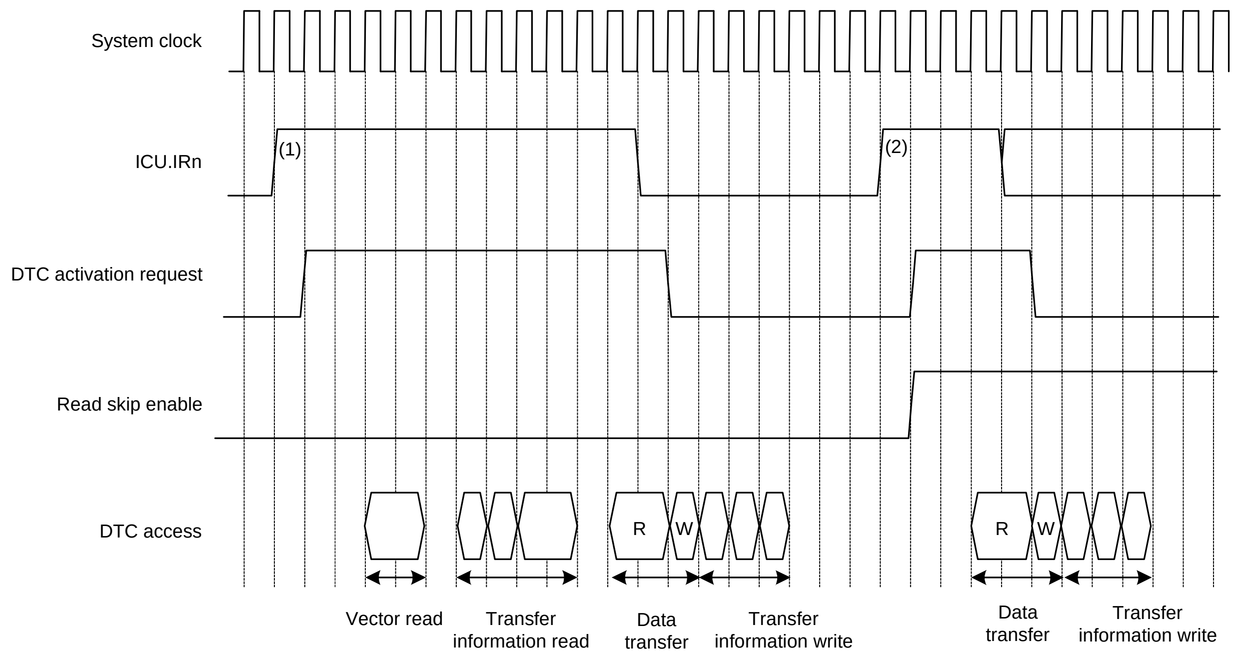


Figure 17.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

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

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

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 Information Read			Transfer Information Write			Data Transfer Read	Data Transfer Write	Internal Operation	
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed mode
Note 5. When SAR or DAR is set to address-fixed mode
Note 6. When SAR and DAR are set to address-fixed mode
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 33, RAM, section 34, Flash Memory, and section 5, I/O Registers.)

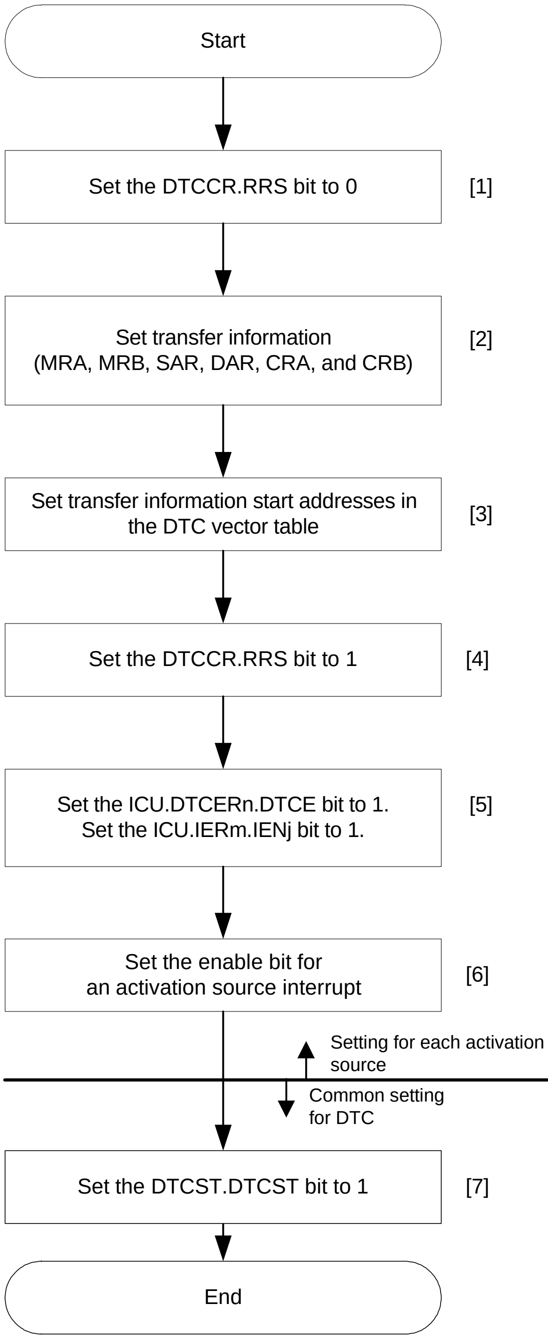
17.4.9 DTC Bus Mastership Release Timing

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

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.



Set the ICU.IERm.IENj bit corresponding to the activation source interrupt to 0 and provide the following settings.

[1] Setting the DTCCR.RRS bit to 0 resets the transfer information read skip flag. After that, transfer information read is not skipped while the DTC is activated. When transfer information is updated, be sure to make this setting.

[2] Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. For setting transfer information, refer to section 17.2, Register Descriptions. For how to allocate transfer information, refer to section 17.3.1, Allocating Transfer Information and DTC Vector Table.

[3] Set transfer information start addresses in the DTC vector table. For how to set the DTC vector table, refer to section 17.3.1, Allocating Transfer Information and DTC Vector Table.

[4] Setting the DTCCR.RRS bit to 1 enables skipping of the second and the subsequent transfer information read cycles for continuous DTC activation due to the same interrupt source. The RRS bit can be set to 1, but this setting during DTC transfer becomes valid from the next transfer.

[5] Set the ICU.DTCERn.DTCE bit (n = interrupt vector number) that corresponds to the DTC activation interrupt to 1. Set the ICU.IERm.IENj bit to 1. For the correspondence between interrupt sources and DTCERn register, refer to the Interrupt Vector Table.

[6] Set the enable bit for an activation source interrupt to 1. When a source interrupt is generated, the DTC is activated. For the setting of the interrupt source enable bit, refer to the setting of the module that is to be an activation source.

[7] Set the DTC module start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.

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 reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

In the MRA register, select a fixed source address (MRA.SM[1:0] bits = 00b), normal transfer mode (MRA.MD[1:0] bits = 00b), and byte-sized transfer (MRA.SZ[1:0] bits = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] bits = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

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

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 rounds of data transfer have been completed and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

17.6.2 Chain Transfer When the Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Repeating this chain transfer enables transfers to be repeated 256 times or more.
The following shows an example of configuring a 128-Kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 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, the CRA register = 0000h (65,536 times), the MRB.CHNE bit = 1 (chain transfer is enabled), the MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0), and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
2. Prepare the upper 8-bit address of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM). For example, when setting the input buffer to 20 0000h to 21 FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination. At this time, set the MRB.CHNE bit = 0 (chain transfer is disabled) and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed). When setting the input buffer mentioned above to 20 0000h to 21 FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

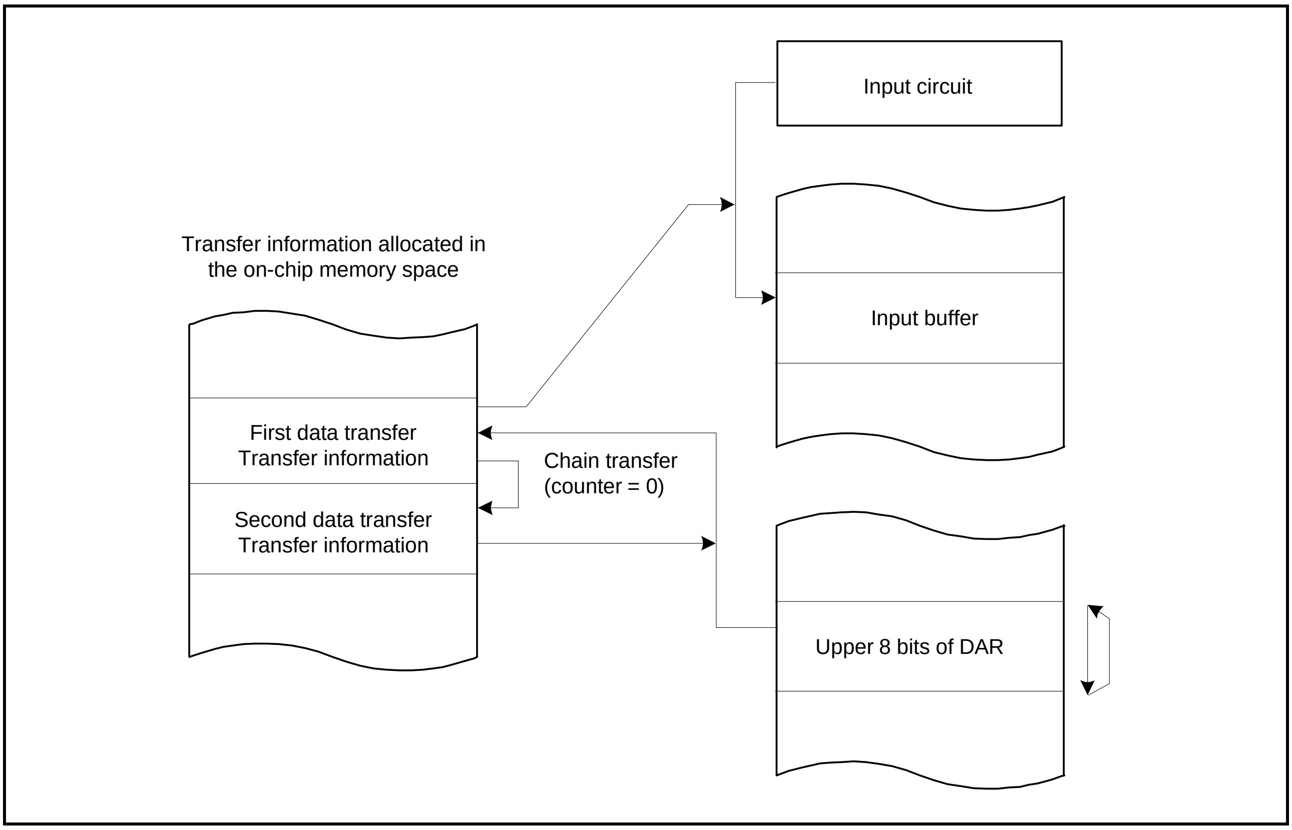


Figure 17.15 Chain Transfer When the Counter = 0

17.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC activation source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

17.8 Low Power Consumption Function

Before making a transition to the module stop state, deep sleep mode, or software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If DTC transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) Deep Sleep Mode

Make settings according to the procedure under section 11.6.2.1, Entry to Deep Sleep 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 deep sleep 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 deep sleep mode.

(3) Software Standby Mode

Make settings according to the procedure under section 11.6.3.1, Entry 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, refer to section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DTC transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in deep sleep mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

17.9 Usage Notes

17.9.1 Transfer Information Start Address

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

17.9.2 Allocating Transfer Information

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.

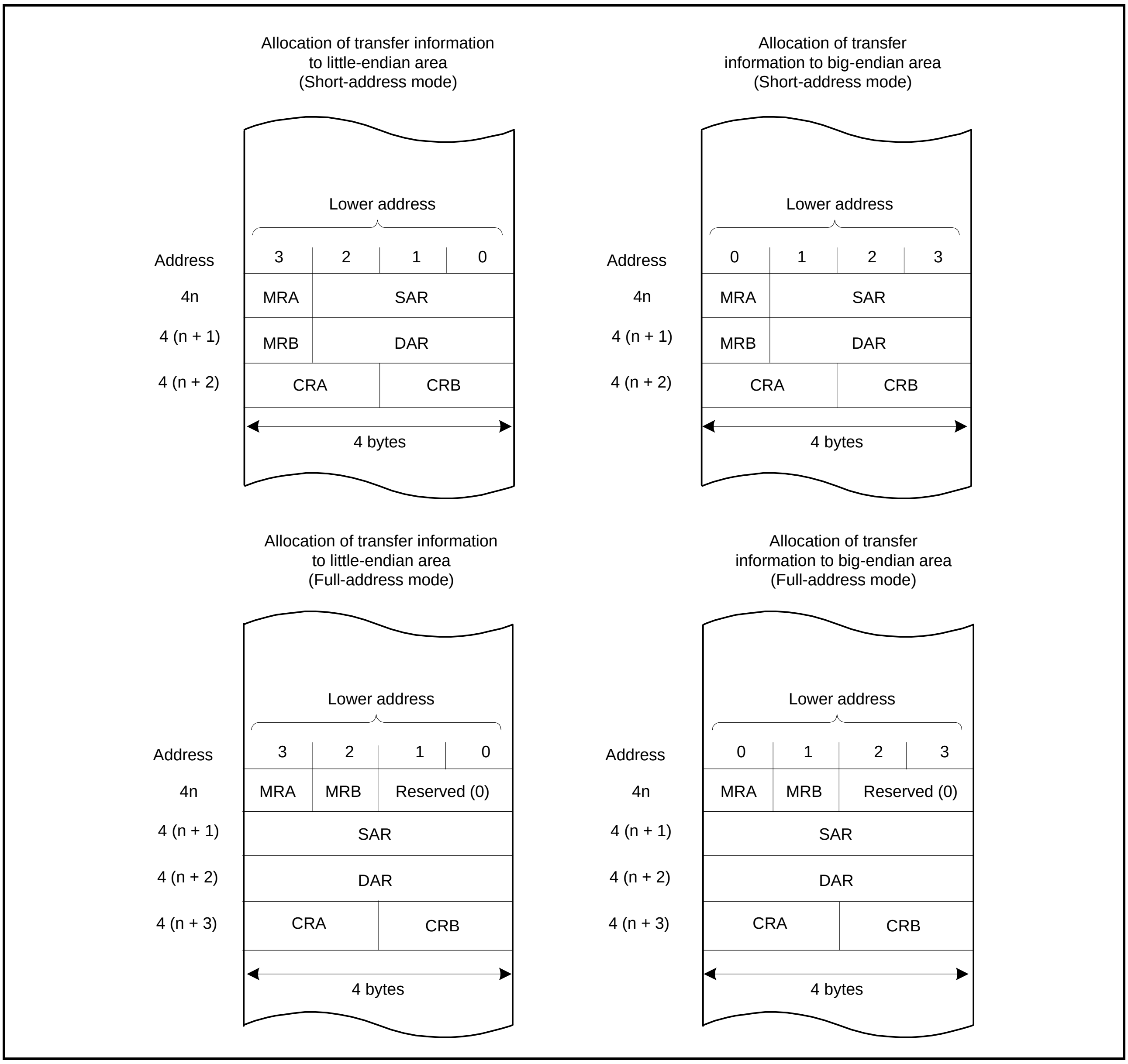


Figure 17.16 Allocation of Transfer Information

18. I/O Ports

18.1 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 on-chip peripheral modules.
Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, the drive capacity control register (DSCR) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.
For details on the PMR register, see section 19, Multi-Function Pin Controller (MPC).
The configuration of the I/O ports differs depending on the package. Table 18.1 lists the specifications of I/O ports, and Table 18.2 list the port functions.

Table 18.1 Specifications of I/O Ports

Port	Package 64 Pins	Number of Pin	Package 52 Pins	Number of Pin	Package 48 Pins	Number of Pin
PORT0	P00 to P02	3	P02	1	None	None
PORT1	P10, P11	2	P10, P11	2	P10, P11	2
PORT2	P22 to P24	3	P22 to P24	3	P22 to P24	3
PORT3	P30 to P33, P36, P37	6	P33, P36, P37	3	P36, P37	2
PORT4	P40 to P47	8	P40 to P47	8	P40 to P47	8
PORT7	P70 to P76	7	P70 to P76	7	P70 to P76	7
PORT9	P91 to P94	4	P93, P94	2	P93, P94	2
PORTA	PA2 to PA5	4	PA2, PA3, PA5	3	PA2, PA3	2
PORTB	PB0 to PB7	8	PB0 to PB7	8	PB0 to PB6	7
PORTD	PD3 to PD7	5	PD3 to PD6	4	PD3 to PD6	4
PORTE	PE2	1	PE2	1	PE2	1
	Total of Pins	51	Total of Pins	42	Total of Pins	38

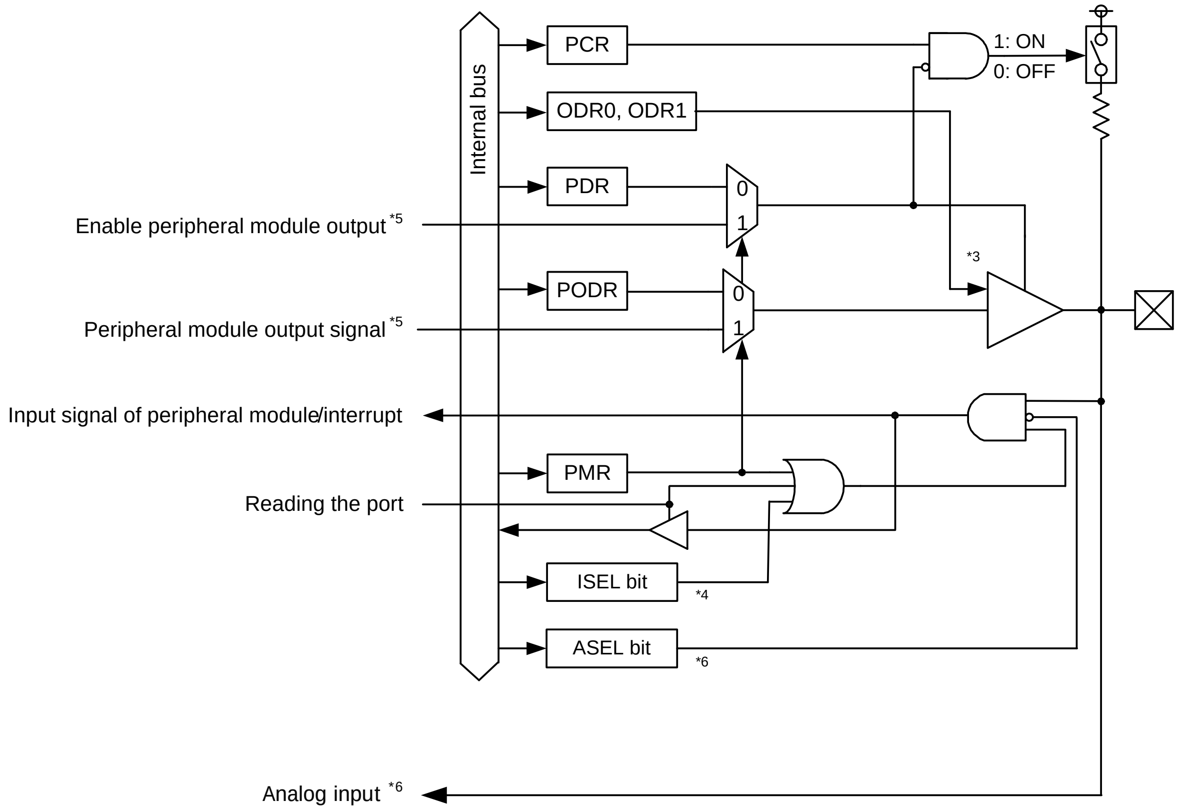
Table 18.2 Port Functions

Port	Pin	Input Pull-up	Open Drain Output	Drive Capacity Switching	High current pin	5-V Tolerant
PORT0	P00, P01, P02	○	○	○	—	—
PORT1	P10, P11	○	○	○	—	—
PORT2	P22	○	○	○	—	—
	P23, P24	○	○	○	—	—
PORT3	P30 to P33, P36, P37	○	○	○	—	—
PORT4	P40 to P47	○	—	Fixed to normal output	—	—
PORT7	P70	○	○	○	—	—
	P71 to P76	○	○	Fixed to high drive output	○	—
PORT9	P91, P92, P94	○	○	○	—	—
	P93	○	○	○	—	—
PORTA	PA2, PA3, PA5	○	○	○	—	—
	PA4	○	○	○	—	—
PORTB	PB0, PB3	○	○	○	—	—
	PB1, PB2	○	○	Fixed to high drive output	—	○
	PB5	○	○	Fixed to high drive output	○	—
	PB4, PB6, PB7	○	○	○	—	—
PORTD	PD3	○	○	Fixed to high drive output	○	—
	PD4 to PD7	○	○	○	—	—
PORTE	PE2	—	—	—	—	—

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

18.2 I/O Port Configuration

Port 0: P00*1, P01*1, P02*2
Port 1: P10, P11
Port 2: P22 to P24
Port 3: P30 to P32*1, P33*2
Port 4: P40 to P47
Port 7: P70 to P76
Port 9: P91*1, P92*1, P93, P94
Port A: PA2, PA3, PA4*1, PA5*2
Port B: PB0 to PB6, PB7*2
Port D: PD3 to PD6, PD7*2



Note 1. Not provided on the 52-/48-pin package
Note 2. Not provided on the 48-pin package
Note 3. Control signal for N-channel open-drain output
Note 4. An external interrupt function is multiplexed on this pin
Note 5. A peripheral module output is multiplexed on this pin
Note 6. An analog input function is multiplexed on this pin.

Figure 18.1 I/O Port Configuration (1)

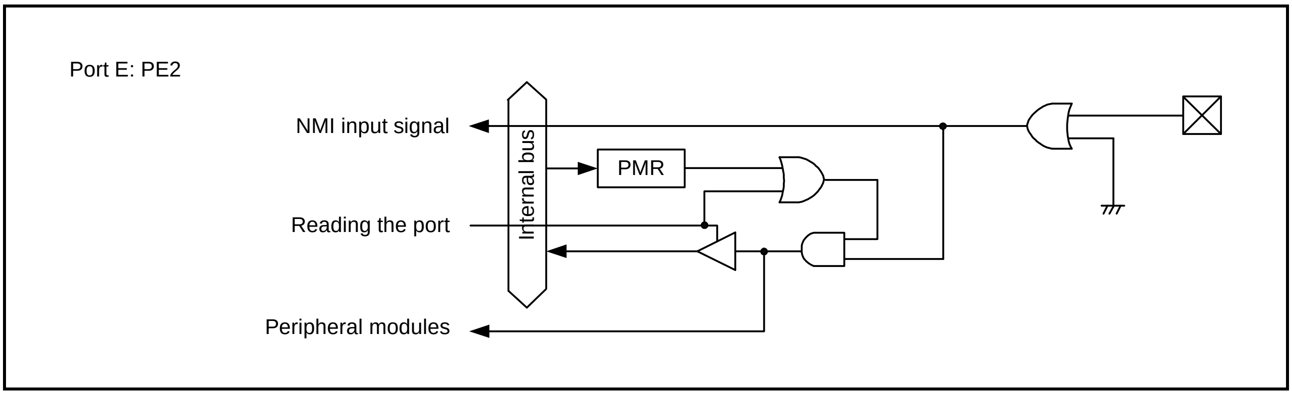


Figure 18.2 I/O Port Configuration (2)

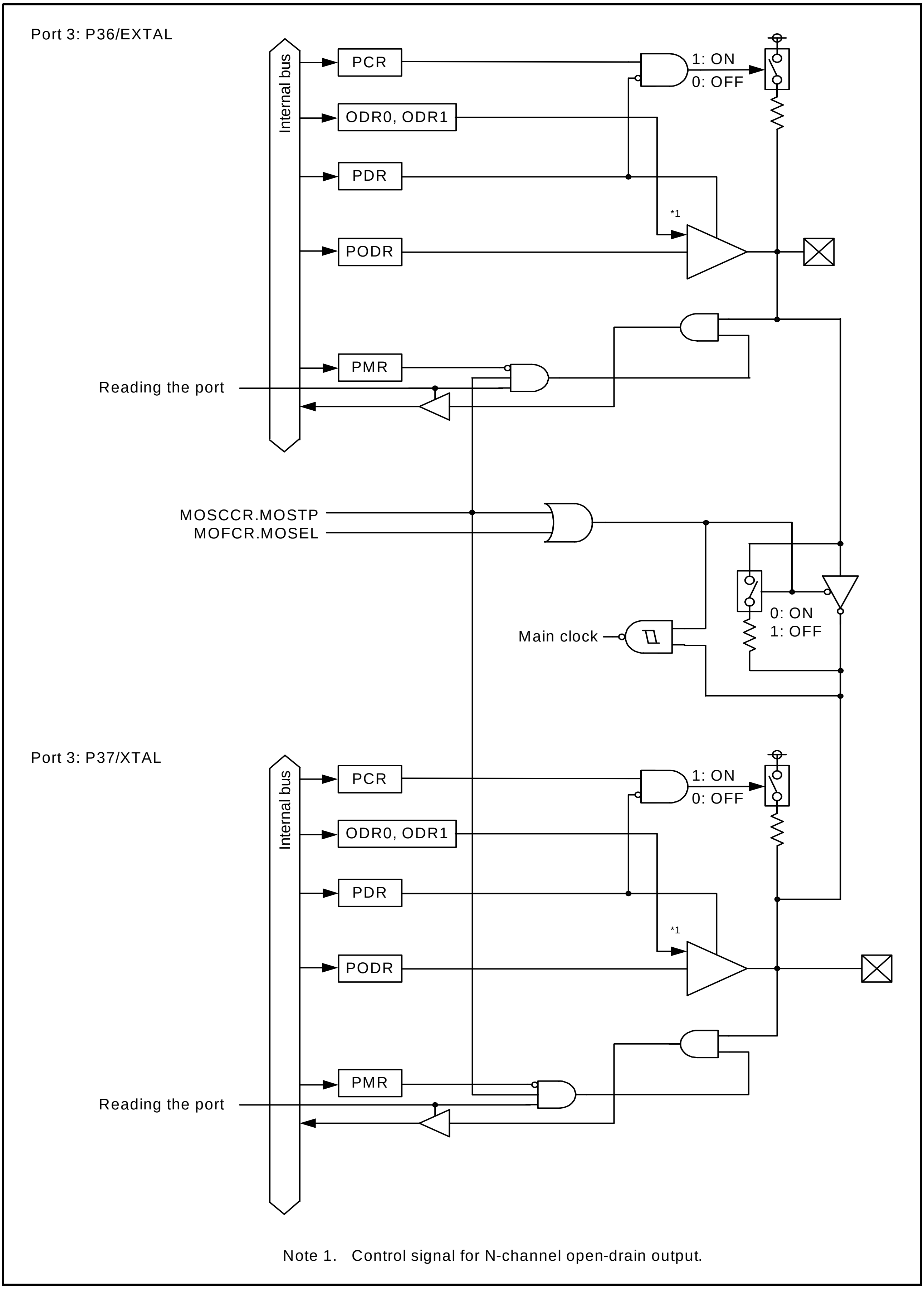


Figure 18.3 I/O Port Configuration (3)

18.3 Register Descriptions

18.3.1 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, PORT7.PDR 0008 C007h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTD.PDR 0008 C00Dh

	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 4, 7, 9, A, B, D

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.

Write 1 (output) to each bit of PDR corresponding to port m that does not exist.

Each bit of PDR corresponding to port m that does not exist is reserved. Make settings according to the description in section 18.4, Initialization of the Port Direction Register (PDR).

The PORTE.PDR.B2 bit is reserved, because the PE2 pin is input only. A reserved bit is read as 0. The write value should be 0.

18.3.2 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, PORT7.PODR 0008 C027h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTD.PODR 0008 C02Dh

	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 4, 7, 9, A, B, D

PODR holds the data to be output from the pins used for general output ports.
Write 1 (output) to each bit of PDR corresponding to port m that does not exist.

18.3.3 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, PORT7.PIDR 0008 C047h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh

	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 4, 7, 9, A, B, D, E

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 PE2 bit.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

18.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT7.PMR 0008 C067h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh

	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 3, 7, 9, A, B, D, E

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.
Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.
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.

18.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT7.ODR0 0008 C08Eh, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTD.ODR0 0008 C09Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm1 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 0 to 3, 7, 9, A, B, D

Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.
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.

18.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT7.ODR1 0008 C08Fh, PORT9.ODR1 0008 C093h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTD.ODR1 0008 C09Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	B1	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 2, 3, 7, 9, A, B, D

Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.
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.

18.3.7 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, PORT7.PCR 0008 C0C7h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTD.PCR 0008 C0CDh

	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 4, 7, 9, A, B, D

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 used 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 the PCR register.

The pull-up resistor is also disabled in the reset state.

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.

18.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT3.DSCR 0008 C0E3h, PORT7.DSCR 0008 C0E7h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTD.DSCR 0008 C0EDh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 0 to 3, 7, 9, A, B, D

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.

18.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 18.3 to Table 18.5.

- The blank columns in Table 18.3 to Table 18.5 indicate the bits corresponding to the pins listed in Table 18.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.
- The columns other than the blank columns in Table 18.3 to Table 18.5 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 18.3 to Table 18.5.
  When setting a value to a reserved bit, access in byte units.

Table 18.3 PDR Register Settings in 64-Pin Packages

Port Symbol	PDR Register							
	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1			
PORT1	1	1	1	1	1	1		
PORT2	1	1	1				1	1
PORT3	0	0	1	1				
PORT4								
PORT7	1							
PORT9	1	1	1					1
PORTA	1	1					1	1
PORTB								
PORTD						1	1	1

Table 18.4 PDR Register Settings in 52-Pin Packages

Port Symbol	PDR Register							
	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1		1	1
PORT1	1	1	1	1	1	1		
PORT2	1	1	1				1	1
PORT3	0	0	1	1		1	1	1
PORT4								
PORT7	1							
PORT9	1	1	1			1	1	1
PORTA	1	1		1			1	1
PORTB								
PORTD	1					1	1	1

Table 18.5 PDR Register Settings in 48-Pin Packages

Port Symbol	PDR Register							
	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1
PORT1	1	1	1	1	1	1		
PORT2	1	1	1				1	1
PORT3	0	0	1	1	1	1	1	1
PORT4								
PORT7	1							
PORT9	1	1	1			1	1	1
PORTA	1	1	1	1			1	1
PORTB	1							
PORTD	1					1	1	1

18.5 Handling of Unused Pins

The configuration of unused pins is listed in Table 18.6.

Table 18.6 Unused Pin Configuration

Pin Name	Description
MD	(Always used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
PE2/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 4, 7, 9, A, B, D.
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 4, 7, 9, A, B, D. When the external clock is input to the EXTAL pin, leave this pin open.
Ports 0 to 4, 7, 9 Ports A, B, D	• 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

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.

19. Multi-Function Pin Controller (MPC)

19.1 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 19.1 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 19.1 Allocation of Pin Functions to Multiple Pins (1 / 4)

Module/Function	Channel	Pin Functions	Allocation on Port	Package		
				64-pin	52-pin	48-pin
Interrupt	NMI	NMI (input)	PE2	○	○	○
Interrupt	IRQ0	IRQ0 (input)	P10	○	○	○
			P93	○	○	○
	IRQ1	IRQ1 (input)	P11	○	○	○
			P94	○	○	○
	IRQ2	IRQ2 (input)	P00	○	×	×
			P22	○	○	○
			PB1	○	○	○
			PD4	○	○	○
	IRQ3	IRQ3 (input)	P24	○	○	○
			PB4	○	○	○
			PD5	○	○	○
	IRQ4	IRQ4 (input)	P01	○	×	×
			P23	○	○	○
			PA2	○	○	○
	IRQ5	IRQ5 (input)	P02	○	○	×
			P70	○	○	○
			PB6	○	○	○
			PD6	○	○	○
Multi-function timer unit 3	MTU0	MTIOC0A (input/output)	P31	○	×	×
			PB3	○	○	○
		MTIOC0B (input/output)	P30	○	×	×
			P93	○	○	○
			PB2	○	○	○
		MTIOC0C (input/output)	P94	○	○	○
			PB1	○	○	○
		MTIOC0D (input/output)	PB0	○	○	○
	MTU1	MTIOC1A (input/output)	PA5	○	○	×
		MTIOC1B (input/output)	PA4	○	×	×
	MTU2	MTIOC2A (input/output)	PA3	○	○	○
		MTIOC2B (input/output)	PA2	○	○	○
	MTU3	MTIOC3A (input/output)	P11	○	○	○
			P33	○	○	×
		MTIOC3B (input/output)	P71	○	○	○
		MTIOC3C (input/output)	P32	○	×	×
		MTIOC3D (input/output)	P74	○	○	○

Table 19.1 Allocation of Pin Functions to Multiple Pins (2 / 4)

Module/Function	Channel	Pin Functions	Allocation Port	Package 64-pin	52-pin	48-pin
Multi-function timer unit 3	MTU4	MTIOC4A (input/output)	P72	○	○	○
		MTIOC4B (input/output)	P73	○	○	○
		MTIOC4C (input/output)	P75	○	○	○
		MTIOC4D (input/output)	P76	○	○	○
	MTU5	MTIC5U (input)	P24	○	○	○
		MTIC5V (input)	P23	○	○	○
		MTIC5W (input)	P22	○	○	○
	MTU	MTCLKA (input)	P33	○	○	×
		MTCLKB (input)	P32	○	×	×
		MTCLKC (input)	P11	○	○	○
			P31	○	×	×
		MTCLKD (input)	P10	○	○	○
			P30	○	×	×
		ADSM0 (output)	PB2	○	○	○
8-bit timer	TMR0	TMO0 (output)	PD3	○	○	○
		TMCI0 (input)	PD4	○	○	○
		TMRI0 (input)	PD5	○	○	○
	TMR1	TMO1 (output)	P94	○	○	○
			PD6	○	○	○
		TMCI1 (input)	P92	○	×	×
		TMRI1 (input)	P93	○	○	○
			PD7	○	×	×
	TMR2	TMO2 (output)	P23	○	○	○
		TMCI2 (input)	P24	○	○	○
		TMRI2 (input)	P22	○	○	○
	TMR3	TMO3 (output)	P11	○	○	○
		TMCI3 (input)	PA5	○	○	×
		TMRI3 (input)	P10	○	○	○
Port output enable 3	POE0	POE0# (input)	P70	○	○	○
	POE8	POE8# (input)	PB4	○	○	○
	POE10	POE10# (input)	PE2	○	○	○
Serial communications interface	SCI1	RXD1 (input) / SMISO1 (input/ output) / SSCL1 (input/output)	PD5	○	○	○
		TXD1 (output) / SMOSI1 (input/ output) / SSDA1 (input/output)	PD3	○	○	○
		SCK1 (input/output)	PD4	○	○	○
		CTS1# (input) / RTS1# (output) / SS1# (input)	P02	○	○	×
			PD6	○	○	○
	SCI5	RXD5 (input) / SMISO5 (input/ output) / SSCL5 (input/output)	PB1	○	○	○
			PB6	○	○	○
		TXD5 (input) / SMOSI5 (input/ output) / SSDA5 (input/output)	PB2	○	○	○
			PB5	○	○	○
		SCK5 (input/output)	P93	○	○	○
			PB3	○	○	○
			PB7	○	○	×

Table 19.1 Allocation of Pin Functions to Multiple Pins (3 / 4)

Module/Function	Channel	Pin Functions	Allocation Port	Package 64-pin	52-pin	48-pin
Serial communications interface	SCI5	CTS5# (input) / RTS5# (output) / SS5# (input)	PA2	○	○	○
I2C bus interface		SCL0 (input/output)	PB1	○	○	○
		SDA0 (input/output)	PB2	○	○	○
Serial peripheral interface		RSPCKA (input/output)	P24	○	○	○
			P93	○	○	○
			PA4	○	×	×
			PB3	○	○	○
		MOSIA (input/output)	P23	○	○	○
			PB0	○	○	○
		MISOA (input/output)	P22	○	○	○
			P94	○	○	○
			PA5	○	○	×
		SSLA0 (input/output)	P30	○	×	×
			PA3	○	○	○
			PD6	○	○	○
		SSLA1 (output)	P31	○	×	×
			PA2	○	○	○
			PD7	○	×	×
		SSLA2 (output)	P32	○	×	×
			P92	○	×	×
		SSLA3 (output)	P33	○	○	×
			P91	○	×	×
12-bit A/D converter		AN000 (input)	P40	○	○	○
		AN001 (input)	P41	○	○	○
		AN002 (input)	P42	○	○	○
		AN003 (input)	P43	○	○	○
		AN004 (input)	P44	○	○	○
		AN005 (input)	P45	○	○	○
		AN006 (input)	P46	○	○	○
		AN007 (input)	P47	○	○	○
		AN016 (input)	P11	○	○	○
		AN017 (input)	P10	○	○	○
		ADTRG0# (input)	PA4	○	×	×
		ADST0 (output)	P02	○	○	×
			PD6	○	○	○
Clock frequency accuracy measurement circuit		CACREF (input)	P01	○	×	×
			P23	○	○	○
			PB3	○	○	○

Table 19.1 Allocation of Pin Functions to Multiple Pins (4 / 4)

Module/Function	Channel	Pin Functions	Allocation Port	Package 64-pin	52-pin	48-pin
Comparator		CMPC00 (input)	P40	○	○	○
		CMPC01 (input)	P43	○	○	○
		CMPC02 (input)	P46	○	○	○
		CMPC10 (input)	P41	○	○	○
		CMPC11 (input)	P44	○	○	○
		CMPC12 (input)	P47	○	○	○
		CMPC20 (input)	P42	○	○	○
		CMPC21 (input)	P45	○	○	○
		CMPC22 (input)	P47	○	○	○
		COMP0 (output)	P24	○	○	○
		COMP1 (output)	P23	○	○	○
		COMP2 (output)	P22	○	○	○
		CVREFC0 (input)	P11	○	○	○
		CVREFC1 (input)	P10	○	○	○

19.2 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.

19.2.1 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.

19.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 2)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P00: IRQ2 input switch (64-pin) P01: IRQ4 input switch (64-pin) P02: IRQ5 input switch (64-/52-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The Pmn pin function control register (PmnPFS) selects the pin function.
Bits PSEL[4:0] select the peripheral function assigned to each port pin.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ISEL bit to which IRQn is not specified is reserved.

Table 19.2 Register Settings for Input/Output Pin Function in 64-/52-pin

	Register/Pin	
	P01PFS	**P02PFS**
PSEL[4:0] Settings	**P01*1**	**P02**
00000b (Initial value)	Hi-Z	
00111b	CACREF	—
01001b	—	ADST0
01010b	—	CTS1#/RTS1#/SS1#

—: Do not specify this value.
Note 1. Available for 64-pin package only.

19.2.3 P1n Pin Function Select Register (P1nPFS) (n = 0, 1)

Address(es): P10PFS 0008 C148h, P11PFS 0008 C149h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.3.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P10:IRQ0 (64/52/48-pin) P11:IRQ1 (64/52/48-pin)	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin 1: Used as analog pin P10:AN017, CVREF1 (64/52/48-pin) P11:AN016, CVREF0 (64/52/48-pin)	R/W

Table 19.3 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin	
	P10PFS	**P11PFS**
PSEL[4:0] Settings	**P10**	**P11**
00000b (Initial value)	Hi-Z	
00001b	—	MTIOC3A
00010b	MTCLKD	MTCLKC
00101b	TMRI3	TMO3

—: Do not specify this value.

19.2.4 P2n Pin Function Select Register (P2nPFS) (n = 2 to 4)

Address(es): P22PFS 0008 C152h, P23PFS 0008 C153h, P24PFS 0008 C154h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.4.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P22:IRQ2 (64/52/48-pin) P23:IRQ4 (64/52/48-pin) P24:IRQ3 (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.4 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin		
	P22PFS	**P23PFS**	**P24PFS**
PSEL[4:0] Settings	**P22**	**P23**	**P24**
00000b (Initial value)	Hi-Z		
00001b	MTIC5W	MTIC5V	MTIC5U
00101b	TMRI2	TMO2	TMCI2
00111b	—	CACREF	—
01101b	MISOA	MOSIA	RSPCKA
11110b	COMP2	COMP1	COMP0

—: Do not specify this value.

19.2.5 P3n Pin Function Select Register (P3nPFS) (n = 0 to 3)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.5.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 19.5 Register Settings for Input/Output Pin Function in 64-/52-pin

	Register/Pin			
	P30PFS	P31PFS	P32PFS	P33PFS
PSEL[4:0] Settings	P30 *1	P31 *1	P32 *1	P33
00000b (initial value)	Hi-Z			
00001b	MTIOC0B	MTIOC0A	MTIOC3C	MTIOC3A
00010b	MTCLKD	MTCLKC	MTCLKB	MTCLKA
01101b	SSLA0	SSLA1	SSLA2	SSLA3

Note 1. Available for 64-pin package only.

19.2.6 P4n Pin Function Select Register (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

The ASEL bit is set when a pin is used as an analog pin. The pin state cannot be read at this point.

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 Input Function Select	0: Not used as an analog pin 1: Used as an analog pin P40: AN000 (64/52/48-pin) P41: AN001 (64/52/48-pin) P42: AN002 (64/52/48-pin) P43: AN003 (64/52/48-pin) P44: AN004 (64/52/48-pin) P45: AN005 (64/52/48-pin) P46: AN006 (64/52/48-pin) P47: AN007 (64/52/48-pin)	R/W

19.2.7 P7n Pin Function Select Register (P7nPFS) (n = 0 to 6)

Address(es): P70PFS 0008 C178h, P71PFS 0008 C179h, P72PFS 0008 C17Ah, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.6.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P70: IRQ5 (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.6 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin						
	P70PFS	**P71PFS**	**P72PFS**	**P73PFS**	**P74PFS**	**P75PFS**	**P76PFS**
PSEL[4:0] Settings	**P70**	**P71**	**P72**	**P73**	**P74**	**P75**	**P76**
00000b (initial value)	Hi-Z						
00001b	—	MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
00111b	POE0#	—	—	—	—	—	—

—: Do not specify this value.

19.2.8 P9n Pin Function Select Register (P9nPFS) (n = 1 to 4)

Address(es): P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh, P94PFS 0008 C18Ch

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.7.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P93: IRQ0 (64/52/48-pin) P94: IRQ1 (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.7 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin			
	P91PFS	P92PFS	P93PFS	P94PFS
PSEL[4:0] Settings	P91*1	P92*1	P93	P94
00000b (initial value)	Hi-Z			
00001b	—	—	MTIOC0B	MTIOC0C
00101b	—	TMCI1	TMRI1	TMO1
01010b	—	—	SCK5	—
01101b	SSLA3	SSLA2	RSPCKA	MISOA

—: Do not specify this value.

Note 1. Available for 64-pin package only.

19.2.9 PAn Pin Function Select Register (PAnPFS) (n = 2 to 5)

Address(es): PA2PFS 0008 C192h, PA3PFS 0008 C193h, PA4PFS 0008 C194h, PA5PFS 0008 C195h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.8.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PA2: IRQ4 input switch (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.8 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin			
	PA2PFS	**PA3PFS**	**PA4PFS**	**PA5PFS**
PSEL[4:0] Settings	**PA2**	**PA3**	**PA4*1**	**PA5*2**
00000b (initial value)	Hi-Z			
00001b	MTIOC2B	MTIOC2A	MTIOC1B	MTIOC1A
00101b	—	—	—	TMCI3
01001b	—	—	ADTRG0#	—
01010b	CTS5#/RTS5#/SS5#	—	—	—
01101b	SSLA1	SSLA0	RSPCKA	MISOA

—: Do not specify this value.
Note 1. Available for 64-pin package only.
Note 2. Available for 64-/52-pin package only.

19.2.10 PBn Pin Function Select Register (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[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.9.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB1:IRQ2 (64/52/48-pin) PB4:IRQ3 (64/52/48-pin) PB6:IRQ5 (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.9 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

PSEL[4:0] Settings	Register/Pin							
	PB0PFS	**PB1PFS**	**PB2PFS**	**PB3PFS**	**PB4PFS**	**PB5PFS**	**PB6PFS**	**PB7PFS**
	PB0	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7*1**
00000b (initial value)	Hi-Z							
00001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	—	—
00111b	—	—	—	CACREF	POE8#	—	—	—
01001b	—	—	ADSM0	—	—	—	—	—
01010b	—	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	—	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5	SCK5
01101b	MOSIA	—	—	RSPCKA	—	—	—	—
01111b	—	SCL0	SDA0	—	—	—	—	—

—: Do not specify this value.
Note 1. Available for 64-/52-pin package only.

19.2.11 PDn Pin Function Select Register (PDnPFS) (n = 3 to 7)

Address(es): PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.10.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PD4:IRQ2 (64/52/48-pin) PD5:IRQ3 (64/52/48-pin) PD6:IRQ5 (64/52/48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 19.10 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin				
	PD3PFS	**PD4PFS**	**PD5PFS**	**PD6PFS**	**PD7PFS**
PSEL[4:0] Settings	**PD3**	**PD4**	**PD5**	**PD6**	**PD7*1**
00000b (initial value)	Hi-Z				
00101b	TMO0	TMCI0	TMRI0	TMO1	TMRI1
01001b	—	—	—	ADST0	—
01010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	CTS1#/RTS1#/SS1#	—
01101b	—	—	—	SSLA0	SSLA1

—: Do not specify this value.
Note 1. Available for 64-pin package only.

19.2.12 PEn Pin Function Select Register (PE2PFS)

Address(es): PE2PFS 0008 C1B2h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 19.11.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 19.11 Register Settings for Input/Output Pin Function in 64-/52-/48-pin

	Register/Pin
	PE2PFS
PSEL[4:0] Settings	**PE2**
00000b (initial value)	Hi-Z
00111b	POE10#

The priority is given to NMI pin interrupt operation when the NMIER.NMIEN=1

19.3 Usage Notes

19.3.1 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 Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 4, 7, 9, A, B, D, E; n = 0 to 7)

(4) Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.

(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.

(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

19.3.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is cleared to 0. If a Pmn pin function control register is set while the PMR register of corresponding pin is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.

(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.

(3) Do not assign a single function to multiple pins through the MPC settings.

(4) Analog input functions for the A/D converter are multiplexed with input pins of ports 1 and 4. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, i.e. configuring the pin as a general-purpose input, and setting the PmnPFS.ASEL bit to 1.

(5) Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 19.12.
The pin state is readable when the PmnPFS.ASEL bit is 0.
If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bj bit is 0.

Table 19.12 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 19.2 to Table 19.11)	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x	x	Register settings are not required.
Analog inputs	0*2	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (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 2. Setting PORT4 is not required.

Note: The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- 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.

19.3.3 Note on Using Analog Functions

When an analog function is in use, configure the pin as a general-purpose input by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, and then set the ASEL bit in the Pmn pin function control register (PmnPFS) to 1.

19.3.4 Note on PB1 Pin Input Level

PB1 input level is specified to TTL when SCL is selected in the PB1PFS.PSEL bit and SMBus is selected in the ICMR3.SMBS bit in RIIC. At this time, input levels of the PB1 port read and the IRQ2 also become TTL.

20. Multi-Function Timer Pulse Unit (MTU3c)

20.1 Overview

This MCU has an on-chip multi-function timer pulse unit (MTU3c), consisting of six 16-bit timer channels.
Table 20.1 shows the specifications of the MTU and Table 20.2 lists the functions of the MTU. Figure 20.1 shows a block diagram of the MTU.

Table 20.1 MTU Specifications

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU2, 10 clocks for MTU5, and four clocks for MTU1 & MTU2 (LWA = 1))
Operating frequency	Up to 40 MHz
Available operations	[MTU0 to MTU4] • Waveform output on compare match • Input capture function (noise filter setting available) • Counter-clearing operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing on compare match or input capture • Simultaneous input and output to registers in synchronization with counter operations • Up to 12-phase PWM output in combination with synchronous operation
	[MTU0, MTU3, MTU4] • Buffer operation specifiable
	[MTU1, MTU2] • Phase counting mode can be specified independently • 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) • Cascade connection operation available
	[MTU3, MTU4] • Through interlocked operation of MTU3/4, the positive and negative signals in six phases can be output in complementary PWM and reset PWM operation. • In complementary PWM mode, transfer of values from buffer registers to temporary registers on peaks and troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD) • Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] • Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU5] • Capable of operation as a dead-time compensation counter
Interrupt skipping function	• In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	28 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D converter start triggers can be generated
	A/D converter start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 20.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5
Count clock		PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/2 PCLK/4 PCLK/8 PCLK/16 PCLK/32 PCLK/64 PCLK/256 PCLK/1024 MTIOC1A
External clock for phase counting mode		—	MTCLKA MTCLKB	MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TRGALW TRGBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRA/BLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	✓	✓	✓	✓	✓	✓	—
	1 output	✓	✓	✓	✓	✓	✓	—
	Toggle output	✓	✓	✓	✓	✓	✓	—
Input capture function		✓	✓	✓	✓*1	✓	✓	✓
Synchronous operation		✓	✓	✓	—	✓	✓	—
PWM mode 1		✓	✓	✓	—	✓	✓	—
PWM mode 2		✓	✓	✓	—	—	—	—
Complementary PWM mode		—	—	—	—	✓	✓	—
Reset-synchronized PWM mode		—	—	—	—	✓	✓	—
AC synchronous motor drive mode		✓	—	—	—	✓	✓	—
Phase counting mode		—	✓	✓	✓	—	—	—
Buffer operation		✓	—	—	—	✓	✓	—
Dead time compensation counter function		—	—	—	—	—	—	✓
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRA/BLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/underflow (only in complementary PWM mode)	TGR compare match or input capture
A/D converter start trigger		TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—
Interrupt sources		Seven sources •Compare match or input capture 0A •Compare match or input capture 0B •Compare match or input capture 0C •Compare match or input capture 0D •Compare match0E •Compare match0F •Overflow	Four sources •Compare match or input capture 1A •Compare match or input capture 1B •Overflow •Underflow	Four sources •Compare match or input capture 2A •Compare match or input capture 2B •Overflow •Underflow	Four sources •Compare match or input capture 1A •Compare match or input capture 1B •Overflow •Underflow	Five sources •Compare match or input capture 3A •Compare match or input capture 3B •Compare match or input capture 3C •Compare match or input capture 3D •Overflow	Five sources •Compare match or input capture 4A •Compare match or input capture 4B •Compare match or input capture 4C •Compare match or input capture 4D •Overflow or underflow (only in complementary PWM mode)	Three sources •Compare match or input capture 5U •Compare match or input capture 5V •Compare match or input capture 5W

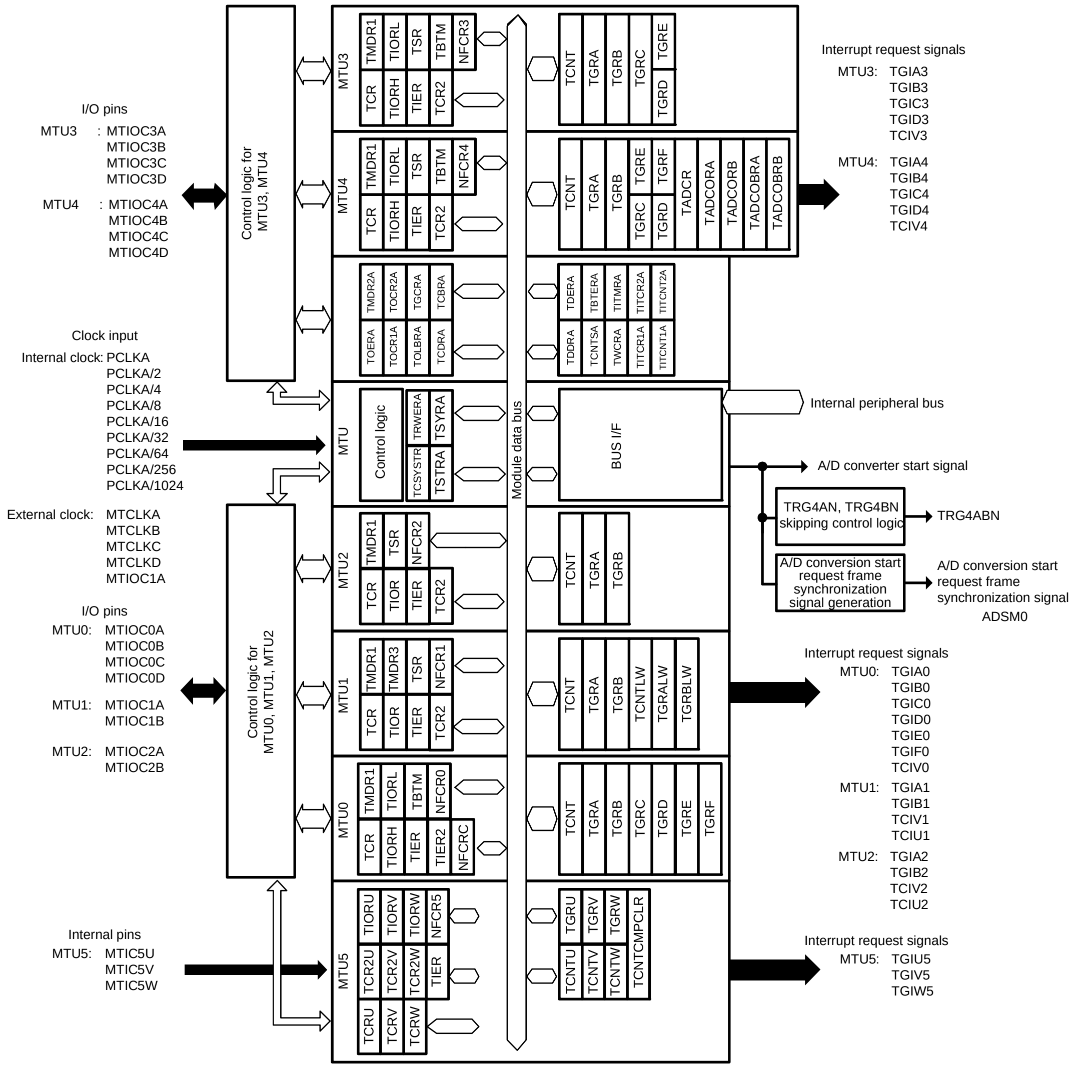
Table 20.2 MTU Functions (2/2)

Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5
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	—
Interrupt skipping 1	—	—	—	—	•Skips TGRA compare match interrupts	•Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	•Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	MSTPCRA.MSTPA9*2						

✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/input capture event.

Note 2. For details on the module stop function, refer to section 11, Low Power Consumption.



TCR: Timer control register
TCR2: Timer control register 2
TCRU: Timer control register U
TCR2U: Timer control register 2U
TCRV: Timer control register V
TCR2V: Timer control register 2V
TCRW: Timer control register W
TCR2W: Timer control register 2W
TMDR1: Timer mode register 1
TMDR2A: Timer mode register 2
TMDR3: Timer mode register 3
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIORU: Timer I/O control register U
TIORV: Timer I/O control register V
TIORW: Timer I/O control register W
TIER: Timer interrupt enable register
TIER2: Timer interrupt enable register 2
TSR: Timer status register
TBTM: Timer buffer operation transfer mode register

TCNT: Timer counter
TCNTLW: Timer longword counter
TGRA: Timer general register A
TGRALW: Timer longword general register A
TGRB: Timer general register B
TGRBLW: Timer longword general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TSTRA: Timer start register
TSYRA: Timer synchronous register
TCSYSTR: Timer counter synchronous start register
TRWERA: Timer read/write enable register
TOERA: Timer output master enable register
TOCR1A: Timer output control register 1
TOCR2A: Timer output control register 2
TOLBRA: Timer output level buffer register
TGCRA: Timer gate control register A
TCNTSA: Timer subcounter
TCDRA: Timer cycle data register
TCBRA: Timer cycle buffer register

TDDRA: Timer dead time data register
TDERA: Timer dead time enable register
TBTERA: Timer buffer transfer set register
TWCRA: Timer waveform control register
TICCR: Timer input capture control register
TADCR: Timer A/D converter start request control register
TADCORA: Timer A/D converter start request cycle set register A
TADCORB: Timer A/D converter start request cycle set register B
TADCOBRA: Timer A/D converter start request cycle set buffer register A
TADCOBRB: Timer A/D converter start request cycle set buffer register B
TITMRA: Timer interrupt skipping mode register
TITCR1A: Timer interrupt skipping set register 1
TITCR2A: Timer interrupt skipping set register 2
TITCNT1A: Timer interrupt skipping counter 1
TITCNT2A: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 20.1 Block Diagram of MTU (MTU0 to MTU5)

Table 20.3 shows the configuration of pins for the MTU.

Table 20.3 Pin Configuration of the MTU

Channel	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 (MTU1 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
	ADSM0	Output	A/D conversion start request frame synchronization signal 0 output pin
MTU0	MTIOC0A	I/O	MTU0 TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0 TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0 TGRC input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	MTU0 TGRD input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1 TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1 TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2 TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2 TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3 TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3 TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3 TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3 TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4 TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4 TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4 TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4 TGRD input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5 TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5 TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5 TGRW input capture input/external pulse input pin

20.2 Register Descriptions

20.2.1 Timer Control Register (TCR)

- MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR

Address(es): MTU0.TCR 000C 1300h, MTU1.TCR 000C 1380h, MTU2.TCR 000C 1400h, MTU3.TCR 000C 1200h, MTU4.TCR 000C 1201h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 20.6 to Table 20.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	Refer to Table 20.4 and Table 20.5.	R/W

x: Don’t care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock. The clock source can be selected independently for each channel. Refer to Table 20.6 to Table 20.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the input clock edge, including the MTIOC1A pin. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLKA/4 at both edges = PCLKA/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 PCLKA/2 or slower. When PCLKA/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 Source Select)

These bits select the TCNT counter clearing source. Refer to Table 20.4 and Table 20.5 for details.

Table 20.4 CCLR[2:0] (MTU0, MTU3, MTU4)

	Bit 7	Bit 6	Bit 5	
Channel	CCLR[2]	CCLR[1]	CCLR[0]	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 TSYRA.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 20.5 CCLR[2:0] (MTU1 and MTU2)

	Bit 7	Bit 6	Bit 5	
Channel	Reserved*2	CCLR[1]	CCLR[0]	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 TSYRA.SYNC bit to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

- MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

Address(es): MTU5.TCRU 000C 1484h, MTU5.TCRV 000C 1494h, MTU5.TCRW 000C 14A4h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPSC[1:0]	Time Prescaler Select	Refer to Table 20.10.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock. Refer to Table 20.10 for details.

20.2.2 Timer Control Register 2 (TCR2)

- MTU0.TCR2, MTU3.TCR2, MTU4.TCR2

Address(es): MTU0.TCR2 000C 1328h, MTU3.TCR2 000C 124Ch, MTU4.TCR2 000C 124Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TCR2, MTU2.TCR2

Address(es): MTU1.TCR2 000C 1394h, MTU2.TCR2 000C 140Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 20.6 to Table 20.9.	R/W
b4, b3	PCB[1:0]	Phase Counting Mode Function Expansion Control	Functional Expansion Control for Phase Counting Modes 2, 3, and 5	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. The MTU has a total of eight TCR2 registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock. The clock source can be selected independently for each channel. Refer to Table 20.6 to Table 20.9 for details.

PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. Refer to section 20.3.6, Phase Counting Mode.

• MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

Address(es): MTU5.TCR2U 000C 1485h, MTU5.TCR2V 000C 1495h, MTU5.TCR2W 000C 14A5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CKEG[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 20.10.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Counts at the rising edge. 0 1: Counts at the falling edge. 1 x: Counts at both edges.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock. Refer to Table 20.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock signal output from the MTIOC1A pin.

Table 20.6 TPSC[2:0], TPSC2[2:0] (MTU0)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

Table 20.7 TPSC[2:0], TPSC2[2:0] (MTU1)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKA/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when MTU1 is in phase counting mode.

Table 20.8 TPSC[2:0], TPSC2[2:0] (MTU2)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKA/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when the TU2 is in phase counting mode.

Table 20.9 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU3 MTU4	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	Internal clock: counts on PCLKA/256
	0	0	0	1	0	1	Internal clock: counts on PCLKA/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Table 20.10 TPSC[1:0], TPSC2[2:0] (MTU5)

	TCR2 register			TCR register		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	Description
MTU5	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	1	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	Setting prohibited
	1	1	1	x	x	Internal clock: counts on MTIOC1A pin input

x: Don't care

Note: Bits 7 to 2 of the TCR register are reserved for MTU5. These bits are read as 0. The write value should be 0.

20.2.3 Timer Mode Register 1 (TMDR1)

- MTU0.TMDR1

Address(es): MTU0.TMDR1 000C 1301h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TMDR1, MTU2.TMDR1

Address(es): MTU1.TMDR1 000C 1381h, MTU2.TMDR1 000C 1401h

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

- MTU3.TMDR1, MTU4.TMDR1

Address(es): MTU3.TMDR1 000C 1202h, MTU4.TMDR1 000C 1203h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 20.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of five TMDR1 registers, one each for MTU0 to MTU4. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 20.11 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4)

Bit 3 MD[3]	Bit 2 MD[2]	Bit 1 MD[1]	Bit 0 MD[0]	Description	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4
0	0	0	0	Normal mode	✓	✓	✓		✓	✓
0	0	0	1	Setting prohibited						
0	0	1	0	PWM mode 1	✓	✓	✓		✓	✓
0	0	1	1	PWM mode 2	✓	✓	✓			
0	1	0	0	Phase counting mode 1		✓	✓	✓		
0	1	0	1	Phase counting mode 2		✓	✓	✓		
0	1	1	0	Phase counting mode 3		✓	✓	✓		
0	1	1	1	Phase counting mode 4		✓	✓	✓		
1	0	0	0	Reset-synchronized PWM mode*1					✓	
1	0	0	1	Phase counting mode 5		✓	✓	✓		
1	0	1	x	Setting prohibited						
1	1	0	0	Setting prohibited						
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1					✓	
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1					✓	
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1					✓	

x: Don't care

Note: Only set the corresponding operating mode listed above for each channel.

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3.
When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and automatically conform to the MTU3 setting, respectively. MTU4 should be set to the initial values (normal mode).

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.
If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.
In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA bit of MTU3.TMDR1. The BFA bit of MTU4.TMDR1 should be set to 0.
In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 20.47 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.
If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.
In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFB bit of MTU3.TMDR1. The BFB bit of MTU4.TMDR1 should be set to 0.
In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 20.47 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU0 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

20.2.4 Timer Mode Register 2 (TMDR2A)

Address(es): MTU.TMDR2A 000C 1270h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DRS	Double Buffer Select	0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMDR2A specifies the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). TMDR2A value should be specified only while TCNT operation is stopped.

DRS Bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

20.2.5 Timer Mode Register 3 (TMDR3)

Address(es): MTU1.TMDR3 000C 1391h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LWA	MTU1/MTU2 Combination Longword Access Control	0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
b1	PHCKSEL	External Input Phase Clock Select	0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in Table 20.12.

LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2. When LWA is set to 0, MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW cannot be accessed. This bit is read as 0. When LWA is set to 1, MTU1.TCNT, MTU2.TCNT, MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB cannot be accessed. This bit is read as 0. MTU1 and MTU2 operate together while LWA is 1, so the settings of the timer control registers (TCR and TCR2) and timer I/O control register (TIOR) in MTU1 are enabled, and the settings of the control registers in MTU2 are disabled. Furthermore, MTU2 input capture and compare match are also disabled.
When changing the value of the LWA bit, initialize the counters and general registers for MTU1 and MTU2 in advance.

PHCKSEL Bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects either the A- or B-phase signal from the external clock. Refer to Table 20.51, Clock Input Pins in Phase Counting Mode for details.

Table 20.12 Setting and Combination of the TMDR3 Register

	TMDR3.LWA = 0		TMDR3.LWA = 1	
Register	Symbol	Access mode	Symbol	Access mode
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNT_1_LW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRA_1_LW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRB_1_LW	Longword
General register B in MTU2	MTU2.TGRB	Word		

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

20.2.6 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 000C 1302h, MTU1.TIOR 000C 1382h, MTU2.TIOR 000C 1402h, MTU3.TIORH 000C 1204h, MTU4.TIORH 000C 1206h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A*1	Refer to the following tables. MTU0.TIORH: Table 20.21 MTU1.TIOR: Table 20.23 MTU2.TIOR: Table 20.24 MTU3.TIORH: Table 20.25 MTU4.TIORH: Table 20.27	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 20.13 MTU1.TIOR: Table 20.15 MTU2.TIOR: Table 20.16 MTU3.TIORH: Table 20.17 MTU4.TIORH: Table 20.19	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 000C 1303h, MTU3.TIORL 000C 1205h, MTU4.TIORL 000C 1207h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C*1	Refer to the following tables. MTU0.TIORL: Table 20.22 MTU3.TIORL: Table 20.26 MTU4.TIORL: Table 20.28	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 20.14 MTU3.TIORL: Table 20.18 MTU4.TIORL: Table 20.20	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 000C 1486h, MTU5.TIORV 000C 1496h, MTU5.TIORW 000C 14A6h

	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	Refer to the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 20.29	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TIOR register controls the TGR register. The MTU has a total of 11 TIOR registers, two each for MTU0, MTU3, and MTU4, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.
Note that TIOR is affected by the TMDR1 setting.
The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTRA 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 20.13 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	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 the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT.

x: Don't care

Table 20.14 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	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 the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT.

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.15 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU1.TGRB Function	MTIOC1B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register

x: Don't care

Table 20.16 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	**IOB[2]**	**IOB[1]**	**IOB[0]**	**MTU2.TGRB Function**	**MTIOC2B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.17 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	**IOB[2]**	**IOB[1]**	**IOB[0]**	**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 20.18 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	**IOD[2]**	**IOD[1]**	**IOD[0]**	**MTU3.TGRD Function**	**MTIOC3D Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.19 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	**IOB[2]**	**IOB[1]**	**IOB[0]**	**MTU4.TGRB Function**	**MTIOC4B Pin Function**
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.20 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU4.TGRD Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.21 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	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 the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT.

x: Don't care

Table 20.22 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	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 the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT.

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.23 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	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 20.24 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	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 20.25 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	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 20.26 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	**IOC[2]**	**IOC[1]**	**IOC[0]**	**MTU3.TGRC Function**	**MTIOC3C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.27 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	**IOA[2]**	**IOA[1]**	**IOA[0]**	**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 20.28 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	**IOC[2]**	**IOC[1]**	**IOC[0]**	**MTU4.TGRC Function**	**MTIOC4C Pin Function**
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 20.29 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register*1	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Setting prohibited
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

x: Don't care

Note 1. Set the IOC[4:0] bits to 19h, 1Ah, 1Bh, 1Dh, 1Eh, or 1Fh only when using external pulse width measurement or only when using dead time compensation linked with MTU3 and MTU4. For details, refer to section 20.3.11, External Pulse Width Measurement and section 20.3.12, Dead Time Compensation.

20.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 000C 14B6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

20.2.8 Timer Interrupt Enable Register (TIER)

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 000C 1384h, MTU2.TIER 000C 1404h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU0.TIER, MTU3.TIER

Address(es): MTU0.TIER 000C 1304h, MTU3.TIER 000C 1208h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER

Address(es): MTU4.TIER 000C 1209h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Converter Start Request Enable 2	0: A/D converter start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTUn.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

n = 4

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of seven TIER registers, two for MTU0 and one each for MTU1 to MTU5.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (TGIn) (n = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).
In 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 MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4).
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.

- MTU0.TIER2

Address(es): MTU0.TIER2 000C 1324h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TTGE2	A/D Converter Start Request Enable 2	0: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRn (n = E, F).

TTGE2 Bit (A/D Converter Start Request Enable 2)

Each bit enables or disables A/D converter start requests by compare match between MTU0.TCNT and MTU0.TGRE.

- MTU5.TIER

Address(es): MTU5.TIER 000C 14B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5n Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGIn5) (n = U, V, W).

20.2.9 Timer Status Register (TSR)

- MTU1.TSR, MTU2.TSR

Address(es): MTU1.TSR 000C 1385h, MTU2.TSR 000C 1405h

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

- MTU3.TSR, MTU4.TSR

Address(es): MTU3.TSR 000C 122Ch, MTU4.TSR 000C 122Dh

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined.The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR indicates the states of each of the channels. The MTU has a total of four TSR registers, one each for MTU1 to MTU4.

TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4.

20.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM

Address(es): MTU0.TBTM 000C 1326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM

Address(es): MTU3.TBTM 000C 1238h, MTU4.TBTM 000C 1239h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TTSA	Timing Select A	0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
b1	TTSB	Timing Select B	0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
b2	TTSE	Timing Select E	0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared, 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

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of three TBTM registers, one each for MTU0, MTU3, and MTU4.

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 a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

20.2.11 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 000C 1390h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the TMTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

20.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 000C 1306h, MTU1.TCNT 000C 1386h, MTU2.TCNT 000C 1406h, MTU3.TCNT 000C 1210h, MTU4.TCNT 000C 1212h, MTU5.TCNTU 000C 1480h, MTU5.TCNTV 000C 1490h, MTU5.TCNTW 000C 14A0h

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

Note: TCNT must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNT is a 16-bit readable/writable counter. The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTUV, and TCNTUW) for MTU5. The TCNT counters in MTU0 to MTU4 are initialized to 0000h by a reset. MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW are initialized to 0000h by a reset.
In MTU0 to MTU4, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16- bit units.
The MTU1.TCNT and MTU2.TCNT counters are read as 0000h when TMDR3.LWA is 1. Refer to section 20.2.5, Timer Mode Register 3 (TMDR3) for details.

20.2.13 Timer Longword Counter (TCNTLW)

Address(es): MTU1.TCNTLW 000C 13A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA is 1. The TCNTLW counter is initialized to 0000 0000h by a reset. This counter is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 20.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

20.2.14 Timer General Register (TGR)

Address(es): MTU0.TGRA 000C 1308h, MTU0.TGRB 000C 130Ah, MTU0.TGRC 000C 130Ch, MTU0.TGRD 000C 130Eh, MTU0.TGRE 000C 1320h, MTU0.TGRF 000C 1322h,
MTU1.TGRA 000C 1388h, MTU1.TGRB 000C 138Ah,
MTU2.TGRA 000C 1408h, MTU2.TGRB 000C 140Ah,
MTU3.TGRA 000C 1218h, MTU3.TGRB 000C 121Ah, MTU3.TGRC 000C 1224h, MTU3.TGRD 000C 1226h, MTU3.TGRE 000C 1272h
MTU4.TGRA 000C 121Ch, MTU4.TGRB 000C 121Eh, MTU4.TGRC 000C 1228h, MTU4.TGRD 000C 122Ah, MTU4.TGRE 000C 1274h,
MTU4.TGRF 000C 1276h,
MTU5.TGRU 000C 1482h, MTU5.TGRV 000C 1492h, MTU5.TGRW 000C 14A2h

	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: TGR must not be accessed in 8 bits; it should be accessed in 16 bits. The initial value of TGR is FFFFh.

The TGR register is 16-bit readable/writable register. The MTU has a total of 24 TGR registers, six for MTU0, two each for MTU1 and MTU2, five for MTU3, six for MTU4, and three for MTU5.
The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, 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. The TGRF register 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.
The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as 0000h when TMDR3.LWA is 1. Refer to section 20.2.5, Timer Mode Register 3 (TMDR3) for details.

20.2.15 Timer Longword General Registers (TGRALW, TGRBLW)

Address(es): MTU1.TGRALW 000C 13A4h, MTU1.TGRBLW 000C 13A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

The TGRnLW register (n = A, B) is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRn and MTU2.TGRn. Such operation is only effective when TMDR3.LWA is 1.
The TGRnLW register is initialized to FFFF FFFFh by a reset, but it is read as 0000 0000h when TMDR3.LWA is 0.
Refer to section 20.2.5, Timer Mode Register 3 (TMDR3) for details.
The TGRnLW register functions as an output compare or input capture register when TMDR3.LWA is 1.
This register can only be used in 32-bit phase counting mode.

20.2.16 Timer Start Registers (TSTRA, TSTR)

- MTU.TSTRA (for MTU0, MTU1, MTU2, MTU3, and MTU4

Address(es): MTU.TSTRA 000C 1280h

	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 counting is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
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 counting is stopped 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4.
TSTR starts or stops TCNT operation in MTU5.
Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4)

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.

- MTU5.TSTR

Address(es): MTU5.TSTR 000C 14B4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

20.2.17 Timer Synchronous Register (TSYRA)

Address(es): MTU.TSYRA 000C 1281h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

TSYRA 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 Operation n) (n = 0, 1, 2, 3, 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

20.2.18 Timer Counter Synchronous Start Register (TCSYSTR)

Address(es): MTU.TCSYSTR 000C 1282h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	—	—
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
b3	SCH4	Synchronous Start 4	0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/(W)*1
b4	SCH3	Synchronous Start 3	0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/(W)*1
b5	SCH2	Synchronous Start 2	0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/(W)*1
b6	SCH1	Synchronous Start 1	0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/(W)*1
b7	SCH0	Synchronous Start 0	0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/(W)*1

Note 1. Only 1 can be written to this bit, and doing so sets the flag.
TCSYSTR is automatically cleared after 1 is written to.

TCSYSTR specifies synchronous start of the counters.

SCH4 Bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 Bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 Bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 Bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 Bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

20.2.19 Timer Read/Write Enable Register (TRWERA)

Address(es): MTU.TRWERA 000C 1284h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1
- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)

24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)

20.2.20 Timer Output Master Enable Register (TOERA)

Address(es): MTU.TOERA 000C 120Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 18, I/O Ports.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after clearing the CST3 and CST4 bits in MTU.TSTRA to 0 (refer to Figure 20.42 and Figure 20.45).

20.2.21 Timer Output Control Register 1 (TOCR1A)

Address(es): MTU.TOCR1A 000C 120Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0*4	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*1, *3	Refer to Table 20.30.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 20.31.	R/W
b2	TOCS	TOC Select	0: TOCR1j setting is selected (j = A) 1: TOCR2j setting is selected	R/W
b3	TOCL	TOC Register Write Protection*2	0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Clearing the TOCR1j.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1j.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A enables or disables PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS Bit (TOC Select)

This bit selects either the TOCR1j or TOCR2j (j = A) 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 TOCR1j (j = A).

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Table 20.30 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	**Initial Output**	**Active Level**	**Up-Counting**	**Down-Counting**
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 20.31 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	**Initial Output**	**Active Level**	**Up-Counting**	**Down-Counting**
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 20.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

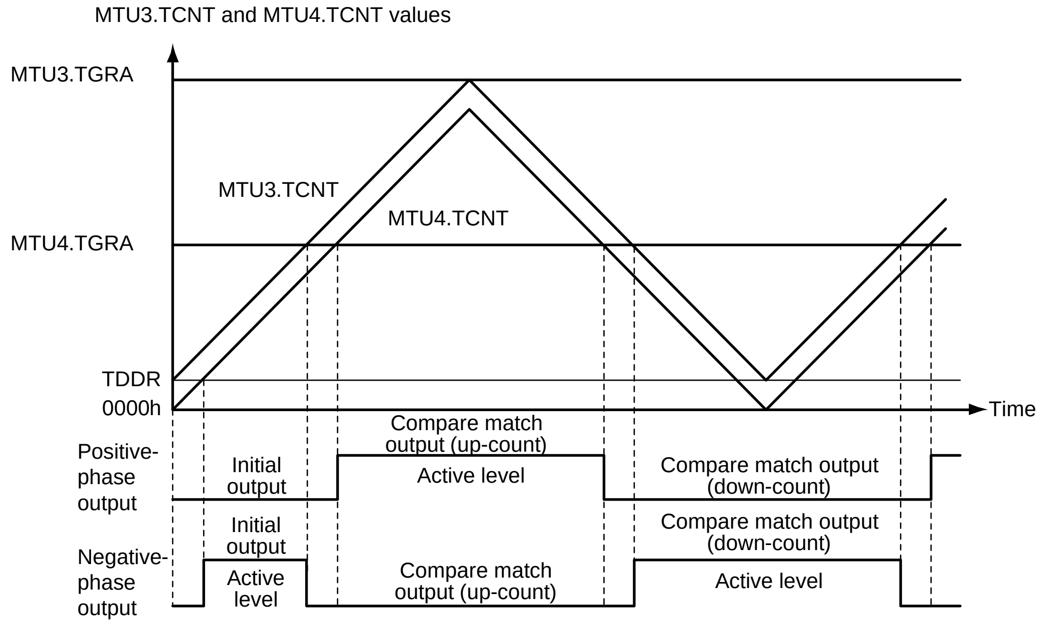


Figure 20.2 Example of Output in Complementary PWM Mode

20.2.22 Timer Output Control Register 2 (TOCR2A)

Address(es): MTU.TOCR2A 000C 120Fh

	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. Refer to Table 20.32.	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. Refer to Table 20.33.	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. Refer to Table 20.34.	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. Refer to Table 20.35.	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. Refer to Table 20.36.	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. Refer to Table 20.37.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRj to TOCR2j. Refer to Table 20.38 for details.	R/W

j = A

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A controls inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode. The initial output is selected while the counter is stopped.

Table 20.32 MTIOCmB Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 3

Table 20.33 MTIOCmD Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 3

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 20.34 MTIOCmA Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 4

Table 20.35 MTIOCmC Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 4

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 20.36 MTIOCmB Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 4

Table 20.37 MTIOCmD Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 4

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 20.38 Setting of TOCR2j.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF[1]	**BF[0]**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.
0	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRj) to TOCR2j when MTUm.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2j at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest and trough of the MTUn.TCNT count.	Setting prohibited

n = 4; m = 3; j = A

20.2.23 Timer Output Level Buffer Register (TOLBRA)

Address(es): MTU.TOLBRA 000C 1236h

	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 TOCR2j.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2j.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2j.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2j.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2j.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2j.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

j = A

TOLBRA is buffer register for TOCR2A and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 20.3 shows an example of the PWM output level setting procedure in buffer operation.

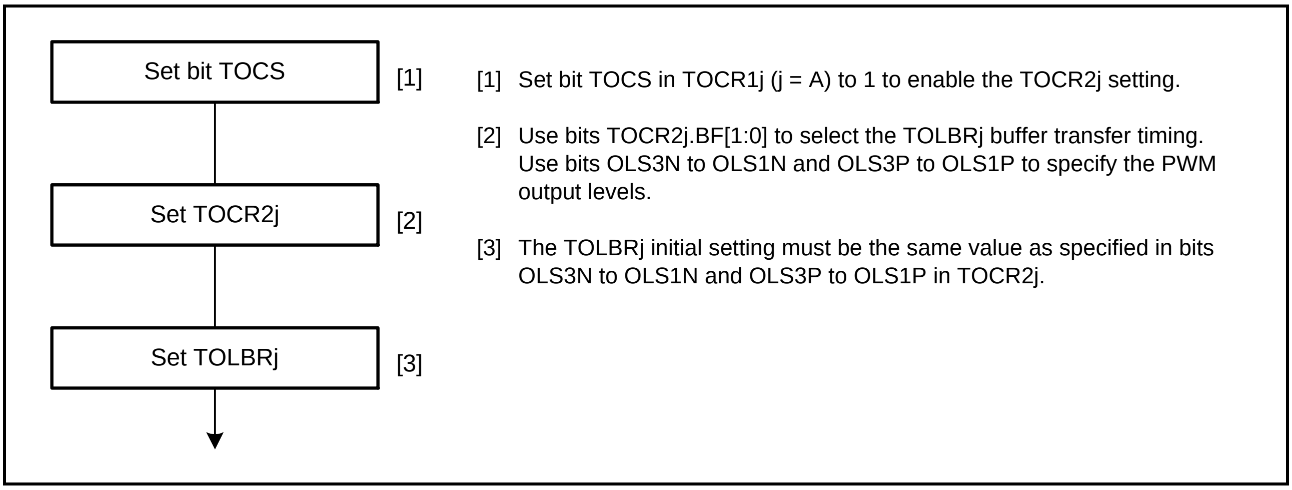


Figure 20.3 Example of PWM Output Level Setting Procedure in Buffer Operation

20.2.24 Timer Gate Control Register A (TGCRA)

Address(es): MTU.TGCRA 000C 120Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 20.39.	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 (TGCRA's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 20.39 for details.

FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA.
When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.

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 TGCRA effective or ineffective.

Table 20.39 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

20.2.25 Timer Subcounter (TCNTSA)

Address(es): MTU.TCNTSA 000C 1220h

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

Note: TCNTSA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNTSA is a 16-bit read-only counter used only in complementary PWM mode.
The initial value of TCNTSA after a reset is 0000h.

20.2.26 Timer Cycle Data Register (TCDRA)

Address(es): MTU.TCDRA 000C 1214h

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

Note: TCDRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCDRA is a 16-bit readable/writable register used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDRA value. The TCDRA register is constantly compared with the TCNTSA counter in complementary PWM mode, respectively. When a match occurs, the TCNTSA counter switches the count direction (down-count to up-count).
The initial value of TCDRA after a reset is FFFFh.

20.2.27 Timer Cycle Buffer Register (TCBRA)

Address(es): MTU.TCBRA 000C 1222h

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

Note: TCBRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCBRA is a 16-bit readable/writable register, used only in complementary PWM mode, that function as buffer register for TCDRA. The TCBRA value is transferred to TCDRA with the transfer timing set in TMDR1. The initial value of TCBRA after a reset is FFFFh.

20.2.28 Timer Dead Time Data Register (TDDRA)

Address(es): MTU.TDDRA 000C 1216h

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

Note: TDDRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TDDRA is a 16-bit readable/writable register, 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 TDDRA value is loaded into the MTU3.TCNT counter and the count operation starts. The initial value of TDDRA after a reset is FFFFh.

20.2.29 Timer Dead Time Enable Register (TDERA)

Address(es): MTU.TDERA 000C 1234h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA must be set to 1 or a larger value.

TDERA controls dead time generation in complementary PWM mode. The MTU has one TDER for MTU3. TDERA should be modified only while TCNT stops.

TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

20.2.30 Timer Buffer Transfer Set Register (TBTERA)

Address(es): MTU.TBTERA 000C 1232h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting	These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, refer to Table 20.40.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA

TBTERA enables or disables transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 20.40 Setting of TBTERA.BTE[1:0] Bits

Bit 1	Bit 0	
BTE[1]	**BTE[0]**	**Description**
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, refer to section 20.3.8, 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 (TITCR1A) or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A 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 (TBTERA) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

20.2.31 Timer Waveform Control Register (TWCRA)

Address(es): MTU.TWCRA 000C 1260h

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

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Waveform Retain Enable	0: Initial values specified in TOCR1A and TOCR2A are output 1: Initial output is inhibited	R/(W)
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE*1	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)

Note 1. Do not set to 1 when complementary PWM mode 1 is not selected.

TWCRA 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 TWCRA should be modified only while TCNT stops.

WRE Bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A are 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, refer to Figure 20.47.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE after reading CCE = 0

20.2.32 Noise Filter Control Register n (NFCRn) (n = 0 to 4, C)

- MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4

Address(es): MTU0.NFCR0 000C 1290h, MTU1.NFCR1 000C 1291h, MTU2.NFCR2 000C 1292h, MTU3.NFCR3 000C 1293h, MTU4.NFCR4 000C 1294h

	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: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in the NFCR1 and NFCR2 registers. These bits are read as 0 and writing to them has no effect.

The NFCRn register sets the noise filter function of external clock pins common to each channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

- MTU0.NFCRC

Address(es): MTU0.NFCRC 000C 1299h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCSC[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTCLKA pin is disabled. 1: The noise filter for the MTCLKA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTCLKB pin is disabled. 1: The noise filter for the MTCLKB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable	0: The noise filter for the MTCLKC pin is disabled. 1: The noise filter for the MTCLKC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTCLKD pin is disabled. 1: The noise filter for the MTCLKD pin is enabled.	R/W
b5, b4	NFCSC[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/2 1 0: PCLKA/8 1 1: PCLKA/32	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCSC[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the input capture function.

20.2.33 Noise Filter Control Register 5 (NFCR5)

Address(es): MTU5.NFCR5 000C 1295h

	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 MTIOC5U pin is disabled. 1: The noise filter for the MTIOC5U pin is enabled.	R/W
b1	NFVEN	Noise Filter V Enable	0: The noise filter for the MTIOC5V pin is disabled. 1: The noise filter for the MTIOC5V pin is enabled.	R/W
b2	NFWEN	Noise Filter W Enable	0: The noise filter for the MTIOC5W pin is disabled. 1: The noise filter for the MTIOC5W 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: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIOC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIOC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIOC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

20.2.34 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 000C 1240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4BN and TCI4V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4BN and TCI4V interrupt skipping 1 are linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4BN and TGI3A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4BN and TGI3A interrupt skipping 1 are linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4AN and TCI4V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4AN and TCI4V interrupt skipping 1 are linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4AN and TGI3A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4AN and TGI3A interrupt skipping 1 are linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter up requests (TRG4AN) enabled during MTU4.TCNT down-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	Refer to Table 20.41 for details. These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB.	R/W

Note: MTU4.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are cleared to 0 or the T3ACOR and T4VCOR bits in TITCR1A are cleared to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function.

Table 20.41 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

Bit 15	Bit 14	Description			
BF[1]	**BF[0]**	**In Complementary PWM Mode**	**In Reset-Synchronized PWM Mode**	**In PWM Mode 1**	**In Normal Mode**
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

20.2.35 Timer A/D Converter Start Request Cycle Set Registers (TADCORA, TADCORB)

Address(es): MTU4.TADCORA 000C 1244h, MTU4.TADCORB 000C 1246h

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

Note: TADCORA and TADCORB must not be accessed in 8 bits; it should be accessed in 16 bits.

Note 1. When the A/D converter start request delaying function linked with skipping function 1 (for details, refer to section 20.3.9 (4), A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0002h to TCDRA setting - 2 in MTU4.

Note 2. When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D converter start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

(1) When skipping function 2 is specified with the skipping count set to 0
- The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
- The TADCORA compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value - 4 or smaller).
- The TADCORB compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value - 4 or smaller).

(2) When skipping function 2 is specified with the skipping count set to 1 or greater
- The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
- The TADCORB compare interval should be equal to or greater than 2 PCLKA cycles (the TADCOR update value should be the previous value + 2 or greater, or previous value - 2 or smaller)

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D converter start request when the MTUn.TCNT (n = 4) count reaches the value in TADCORA or TADCORB.
MTUn.TADCORA and TADCORB are initialized to FFFFh by a reset.

20.2.36 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA, TADCOBRB)

Address(es): MTU4.TADCOBRA 000C 1248h, MTU4.TADCOBRB 000C 124Ah

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

Note: TADCOBRA and TADCOBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.
TADCOBRA and TADCOBRB are initialized to FFFFh by a reset.

20.2.37 Timer Interrupt Skipping Mode Register (TITMRA)

Address(es): MTU.TITMRA 000C 123Ah

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

Bit	Symbol	Bit Name	Description	R/W
b0	TITM	Interrupt Skipping Function Select	Selects one of the two types of interrupt skipping functions shown in Table 20.42.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TITMRA is used to select either of two skipping functions for the TITMRA register.

Table 20.42 Interrupt Skipping Function Selected through TITM Bit

Bit 0 TITM	Description
0	Selects interrupt skipping function 1*1
1	Selects interrupt skipping function 2*2

Note 1. TITCR1A enables interrupt skipping function 1.
Note 2. TITCR2A enables interrupt skipping function 2.

20.2.38 Timer Interrupt Skipping Set Register 1 (TITCR1A)

Address(es): MTU.TITCR1A 000C 1230h

	b7	b6	b5	b4	b3	b2	b1	b0
	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 20.43.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 20.44.	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 TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

TITCR1A enables or disables interrupt skipping and specify the interrupt skipping count. This setting is valid only while TITMRA is set to 0; when TITMRA is set to 1, the setting in the corresponding TITCR1 register is cleared.

Table 20.43 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2 T4VCOR[2]	Bit 1 T4VCOR[1]	Bit 0 T4VCOR[0]	Description
0	0	0	Does not skip TCIV4 interrupts.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count to 4.
1	0	1	Sets the TCIV4 interrupt skipping count to 5.
1	1	0	Sets the TCIV4 interrupt skipping count to 6.
1	1	1	Sets the TCIV4 interrupt skipping count to 7.

Table 20.44 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6 T3ACOR[2]	Bit 5 T3ACOR[1]	Bit 4 T3ACOR[0]	Description
0	0	0	Does not skip TGIA3 interrupts.
0	0	1	Sets the TGIA3 interrupt skipping count to 1.
0	1	0	Sets the TGIA3 interrupt skipping count to 2.
0	1	1	Sets the TGIA3 interrupt skipping count to 3.
1	0	0	Sets the TGIA3 interrupt skipping count to 4.
1	0	1	Sets the TGIA3 interrupt skipping count to 5.
1	1	0	Sets the TGIA3 interrupt skipping count to 6.
1	1	1	Sets the TGIA3 interrupt skipping count to 7.

20.2.39 Timer Interrupt Skipping Counter 1 (TITCNT1A)

Address(es): MTU.TITCNT1A 000C 1231h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. To clear the TITCNT1A, clear the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A is 8-bit readable/writable counters. TITCNTA retains their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT.

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is cleared to 0
- When the T4VCOR[2:0] bits in TITCR1A are cleared to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is cleared to 0
- When the T3ACOR[2:0] bits in TITCR1A are cleared to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

20.2.40 Timer Interrupt Skipping Set Register 2 (TITCR2A)

Address(es): MTU.TITCR2A 000C 123Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting	These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 20.45.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TITCR2A specifies the interrupt skipping count for TRG4AN and TRG4BN. This setting is valid only while TITMRA is set to 1.

Table 20.45 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG4COR[2]	**TRG4COR[1]**	**TRG4COR[0]**	**Description**
0	0	0	Does not skip TRG4AN and TRG4BN interrupts.
0	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 1.
0	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 2.
0	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 3.
1	0	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 4.
1	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 5.
1	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 6.
1	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 7.

20.2.41 Timer Interrupt Skipping Counter 2 (TITCNT2A)

Address(es): MTU.TITCNT2A 000C 123Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter	These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0. Writing to these bit has no effect.	R

TITCNT2A starts counting from the values set in the TRG4COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

TRG4CNT[2:0] Bits (TRG4AN/TRG4BN Interrupt Counter)

These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are cleared to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

20.2.42 A/D Conversion Start Request Select Register 0 (TADSTRGR0)

Address(es): MTU.TADSTRGR0 000C 1D30h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TADSTRS0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TADSTRS0[4:0]	A/D Conversion Start Request Select for ADSM0 Pin Output Frame Synchronization Signal Generation	These bits select the A/D conversion start request for generating the frame synchronization signal to be output from the ADSM0 pin. Refer to Table 20.46 for the relationship between the A/D conversion start request and settings. Settings other than those listed in Table 20.46 are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TADSTRGR0 register selects the A/D conversion start request for generating the A/D conversion start request frame synchronization signal to be output from the ADSM0 pin.

Table 20.46 Settings of A/D Conversion Start Request for Generating Frame Synchronization Signal

Source	Remarks	TADSTRS0[4]	TADSTRS0[3]	TADSTRS0[2]	TADSTRS0[1]	TADSTRS0[0]
—	Source not selected	0	0	0	0	0
TRGA0N	Compare match/input capture in MTU0.TGRA	0	0	0	0	1
TRGA1N	Compare match/input capture in MTU1.TGRA	0	0	0	1	0
TRGA2N	Compare match/input capture in MTU2.TGRA	0	0	0	1	1
TRGA3N	Compare match/input capture in MTU3.TGRA	0	0	1	0	0
TRGA4N	Compare match/input capture in MTU4.TGRA or MTU4.TCNT underflow (trough) in complementary PWM mode	0	0	1	0	1
TRG0N	Compare match in MTU0.TGRE	0	1	0	0	0
TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	1	0	0	1
TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	1	0	1	0
TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT	0	1	0	1	1
TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (Interrupt skipping function 2 used)	0	1	1	0	0

20.2.43 Bus Master Interface

The timer counters (MTU0.TCNT to MTU5.TCNT), general registers (MTU0.TGRn to MTU5.TGRn), timer subcounter (TCNTSA), timer cycle buffer register (TCBRA), timer dead time data register (TDDRA), timer cycle data register (TCDRA), timer A/D converter start request control register (MTU4.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) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Access the registers in 16-bit units.
All registers other than the above registers are 8-bit registers. Read from/write to these registers in 8-bit units.

20.3 Operation

20.3.1 Basic Functions

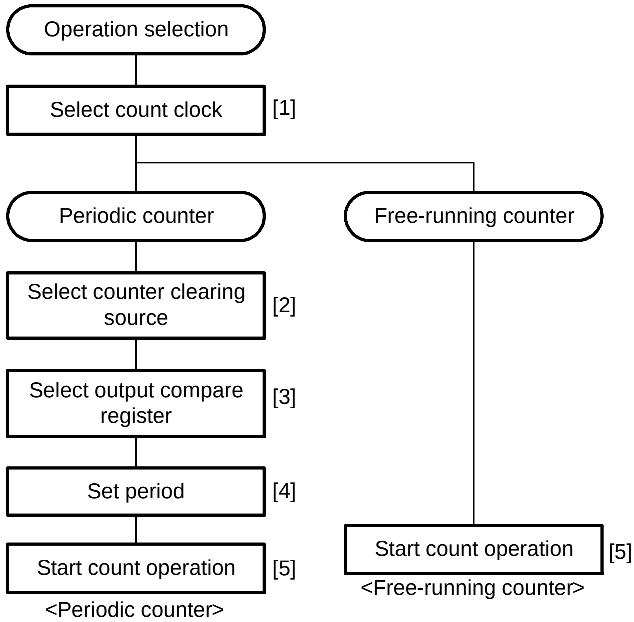
Each channel has TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 in the TSTRA register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 20.4 shows an example of the count operation setting procedure.



[1] Select the count clock with bits TPSC[2:0] in TCR. At the same time, select the input clock edge with bits CKEG[1:0] in TCR.

[2] For periodic counter operation, select the TGR to be used as the TCNT clearing source with bits CCLR[2:0] in TCR.

[3] Designate the TGR selected in [2] as an output compare register by means of TIOR.

[4] Set the periodic counter cycle in the TGR selected in [2].

[5] Set the CST bit in TSTRA or MTU5.TSTR to 1 to start the counter operation.

Figure 20.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TCNT counters are all designated as free-running counters. When the relevant bit in TSTRA or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), an interrupt request is issued to the CPU if the corresponding TIER.TCIEV bit is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 20.5 illustrates free-running counter operation.

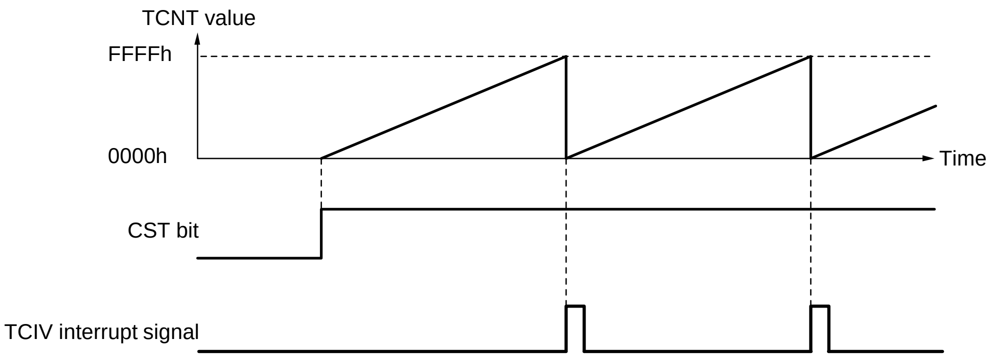


Figure 20.5 Free-Running Counter Operation

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 TSTRA or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from 0000h.
Figure 20.6 illustrates periodic counter operation.

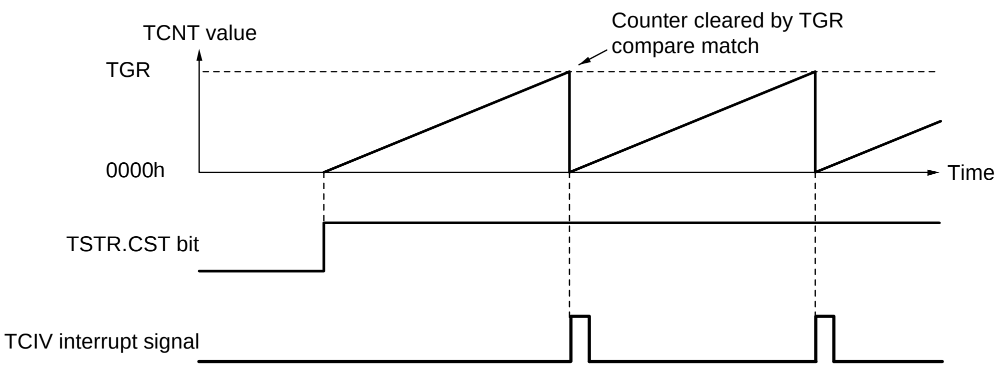


Figure 20.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 20.7 shows an example of the procedure for setting waveform output by compare match

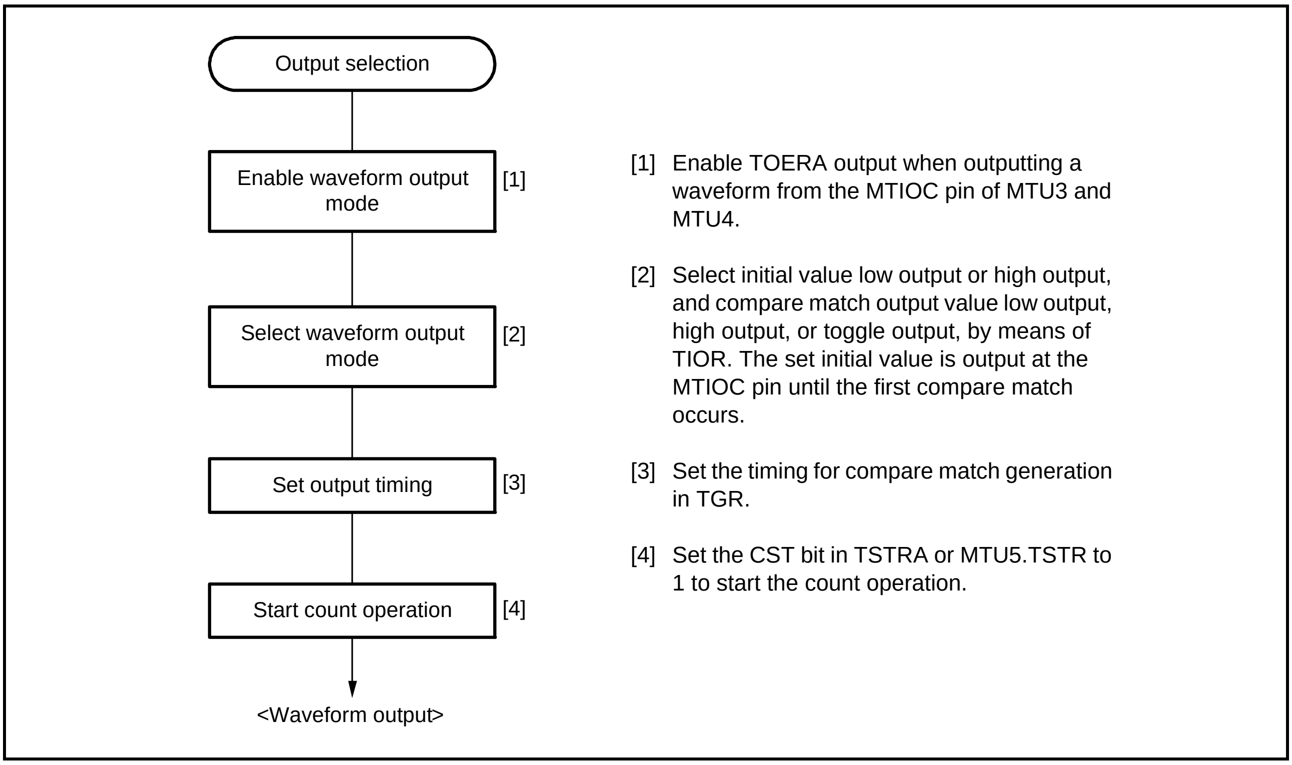


Figure 20.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 20.8 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.

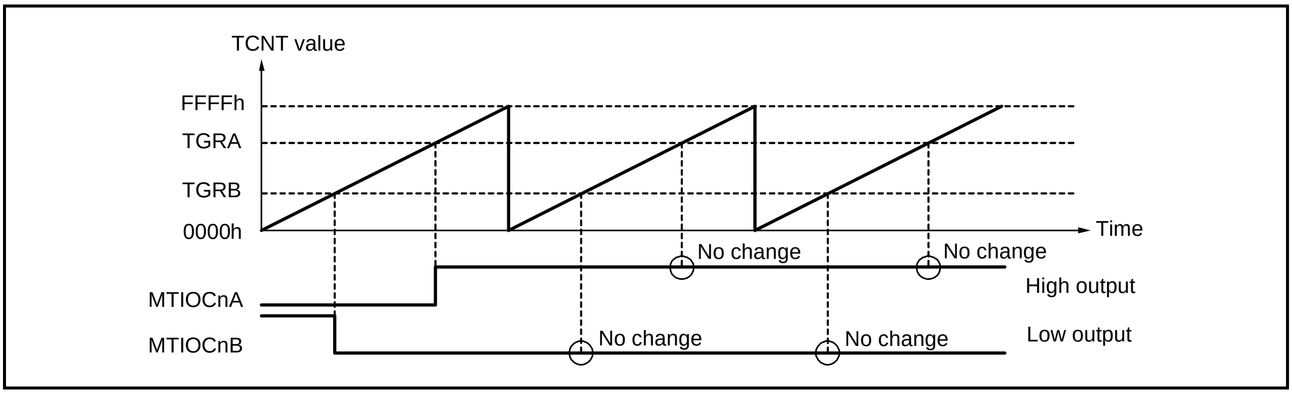


Figure 20.8 Example of low output and high output Operation (n = 0 to 4)

Figure 20.9 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.

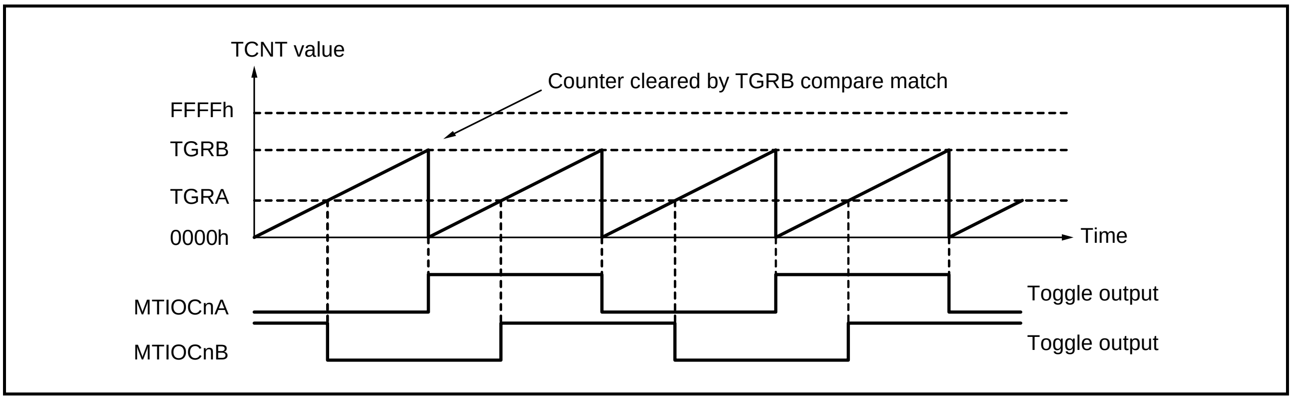


Figure 20.9 Example of Toggle Output Operation (n = 0 to 4)

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm pin (n = 0 to 4; m = A to D) 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, PCLKA/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLKA/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 20.10 shows an example of the input capture operation setting procedure.

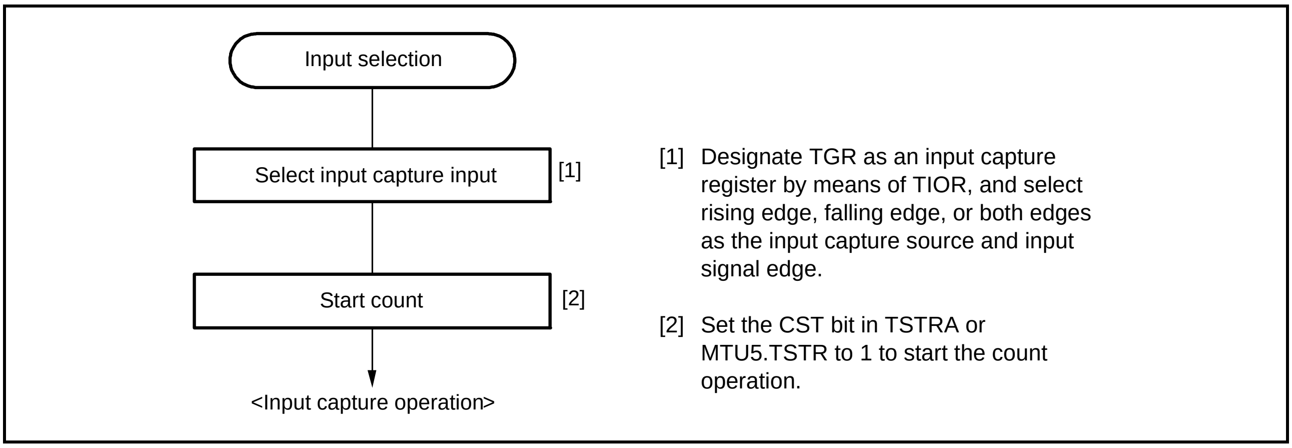


Figure 20.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 20.11 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. (n = 0 to 4)

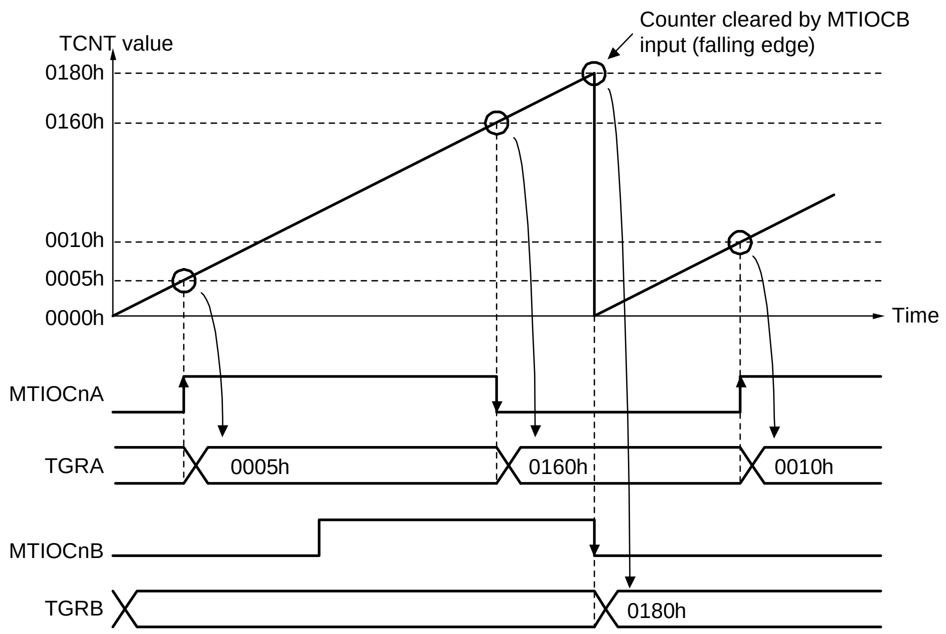


Figure 20.11 Example of Input Capture Operation (n = 0 to 4)

20.3.2 Synchronous Operation

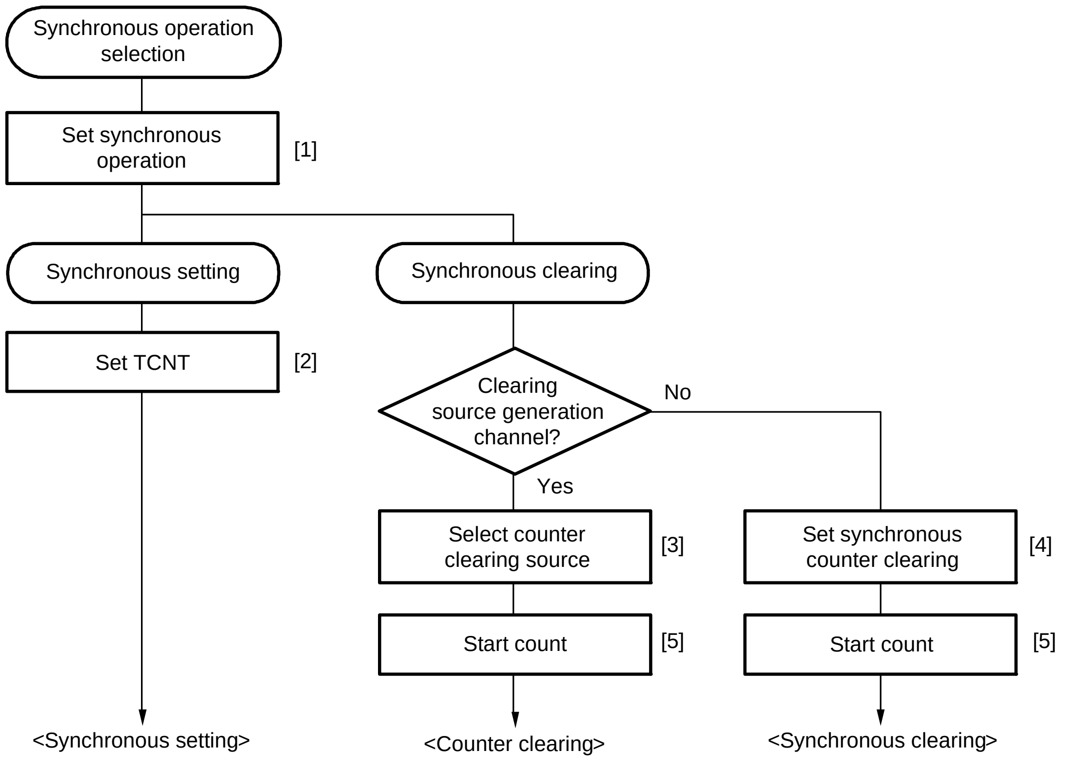
In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation can increase the number of TGR registers assigned to the same time base.

MTU0 to MTU4 can all be designated for synchronous operation. MTU5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 20.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 in the SYNC bits in TSYRA 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 to 1 in the CST bits in TSTRA for the relevant channels, to start the count operation.

Figure 20.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 20.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, refer to section 20.3.5, PWM Modes.

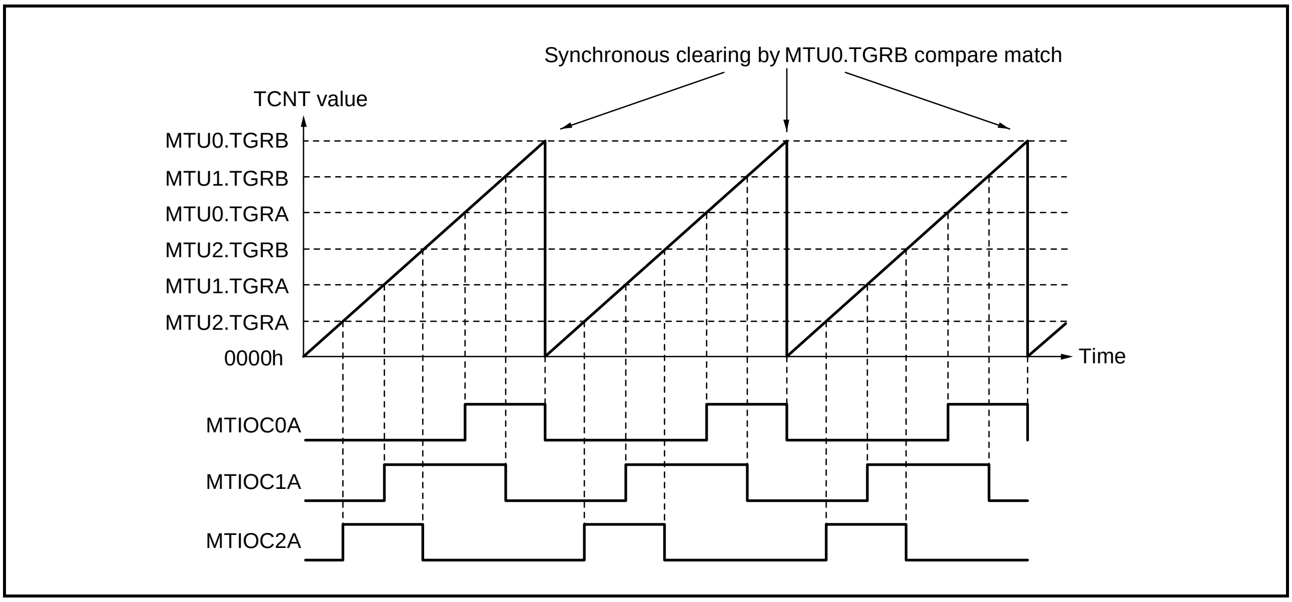


Figure 20.13 Example of Synchronous Operation

20.3.3 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 20.47 shows the register combinations used in buffer operation.

Table 20.47 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 20.14.

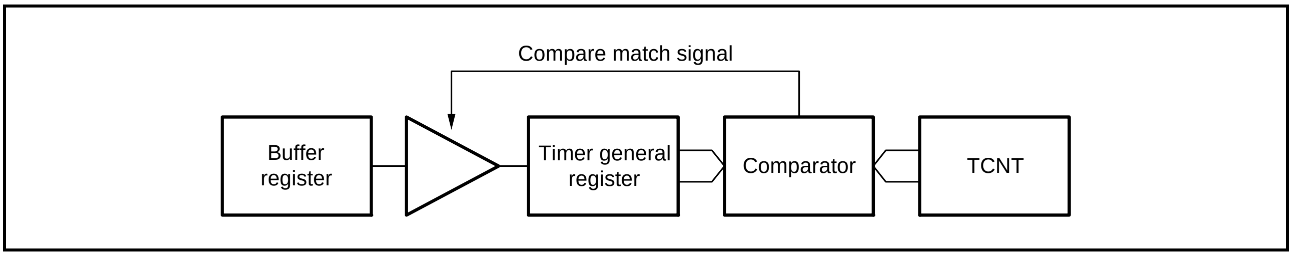


Figure 20.14 Compare Match Buffer Operation

- 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 20.15.

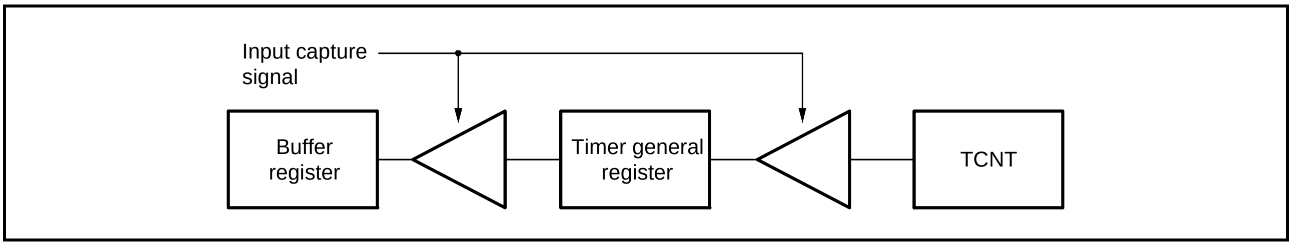


Figure 20.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 20.16 shows an example of the buffer operation setting procedure.

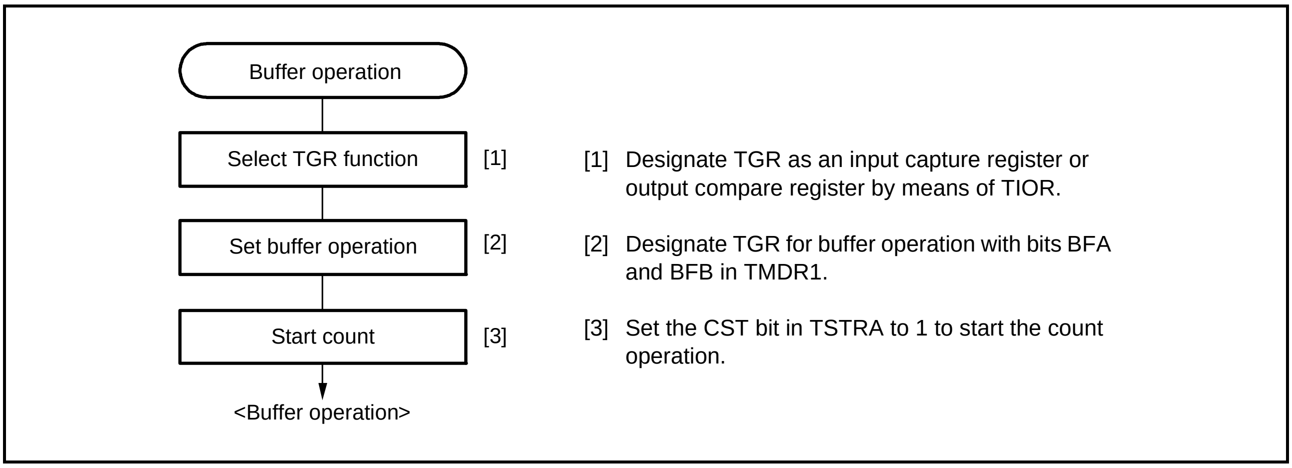


Figure 20.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 20.17 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, refer to section 20.3.5, PWM Modes.

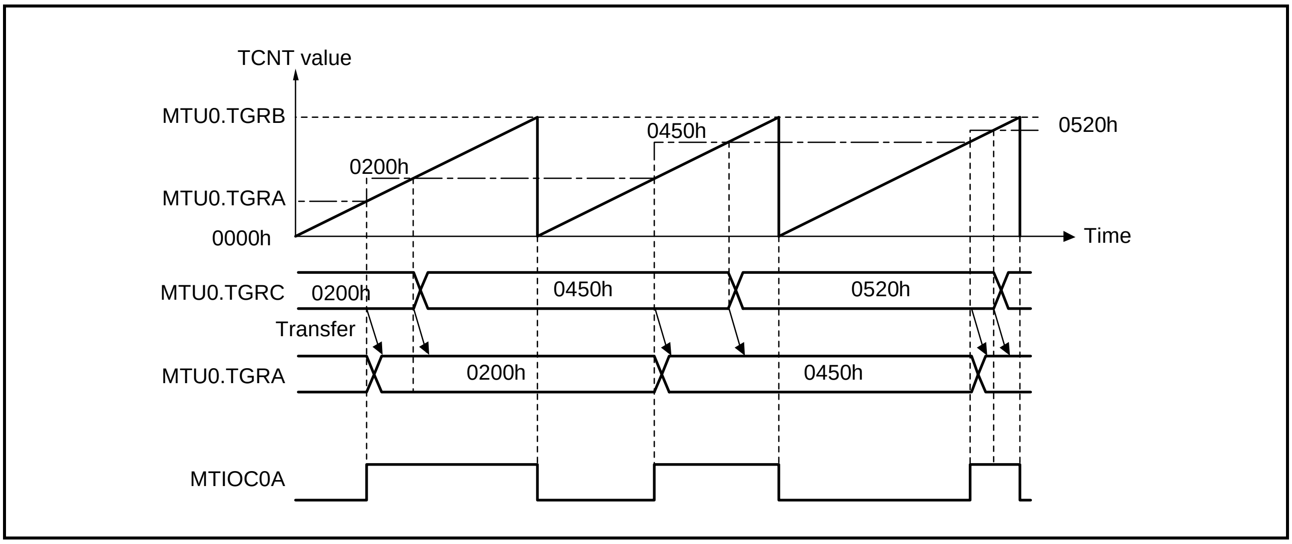


Figure 20.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 20.18 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. (n = 0 to 4)
As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

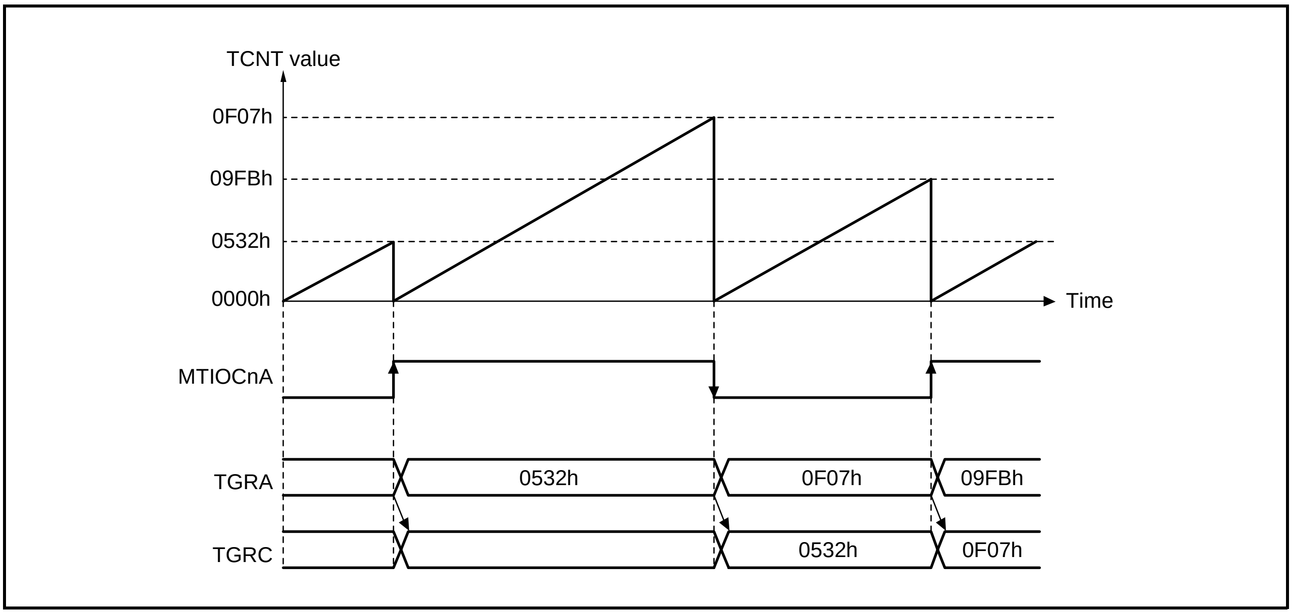


Figure 20.18 Example of Buffer Operation (2) (n = 0 to 4,)

(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 buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT 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 20.19 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.

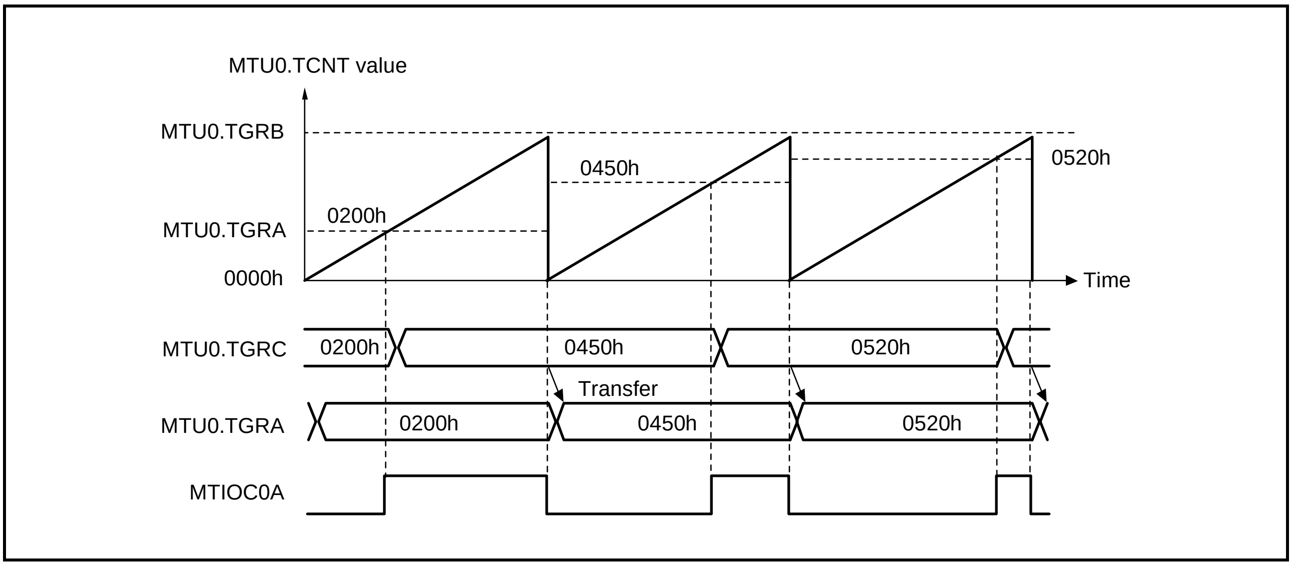


Figure 20.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

20.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.
There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, refer to section 20.3.6.2, Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.
This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.
Table 20.48 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 20.48 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 20.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 20.49 shows the TICCR setting and input capture input pins.

Table 20.49 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 20.20 shows an example of the cascaded operation setting procedure.

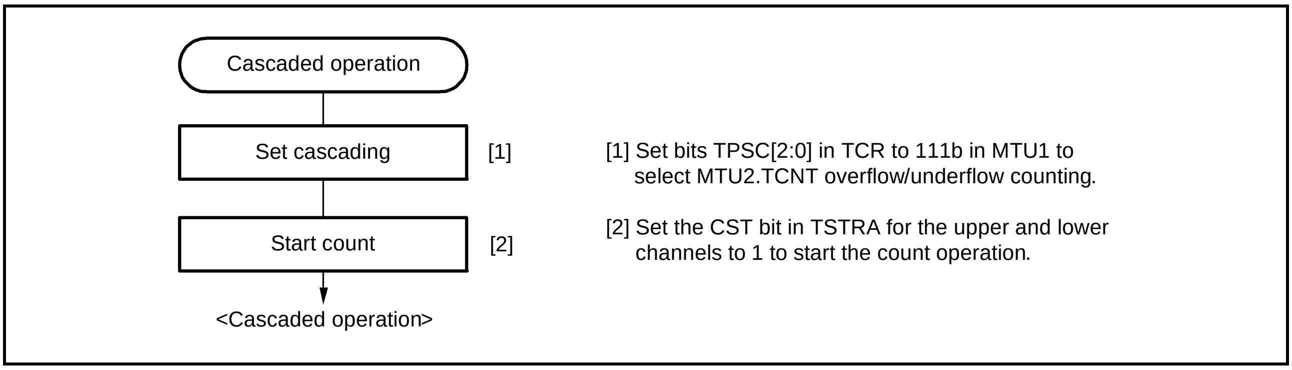


Figure 20.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 20.21 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

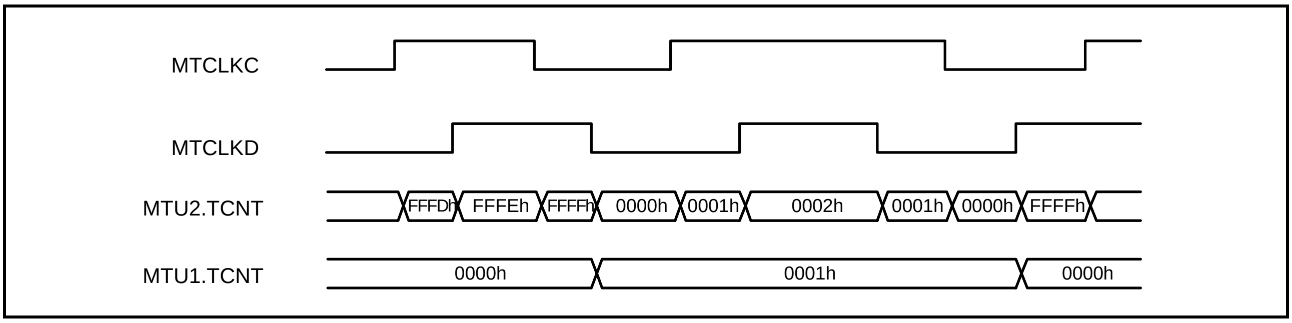


Figure 20.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 20.22 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 MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

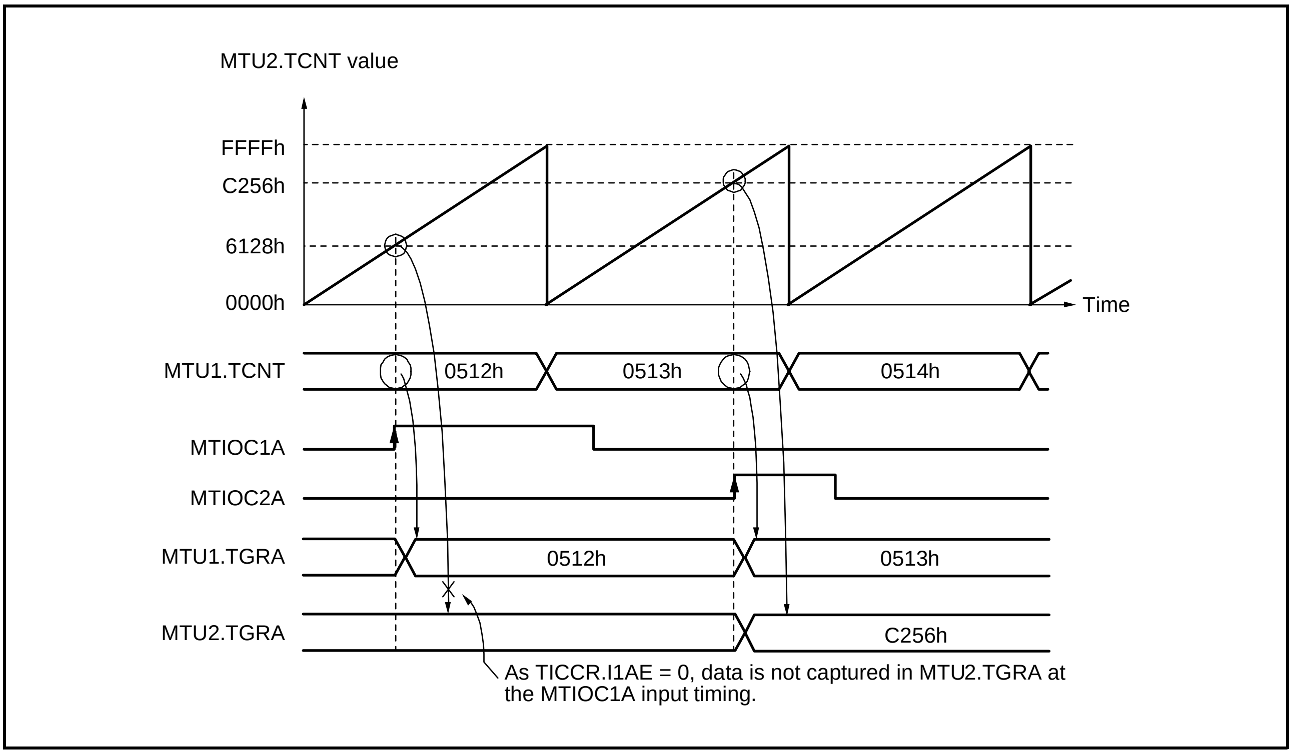


Figure 20.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 20.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.

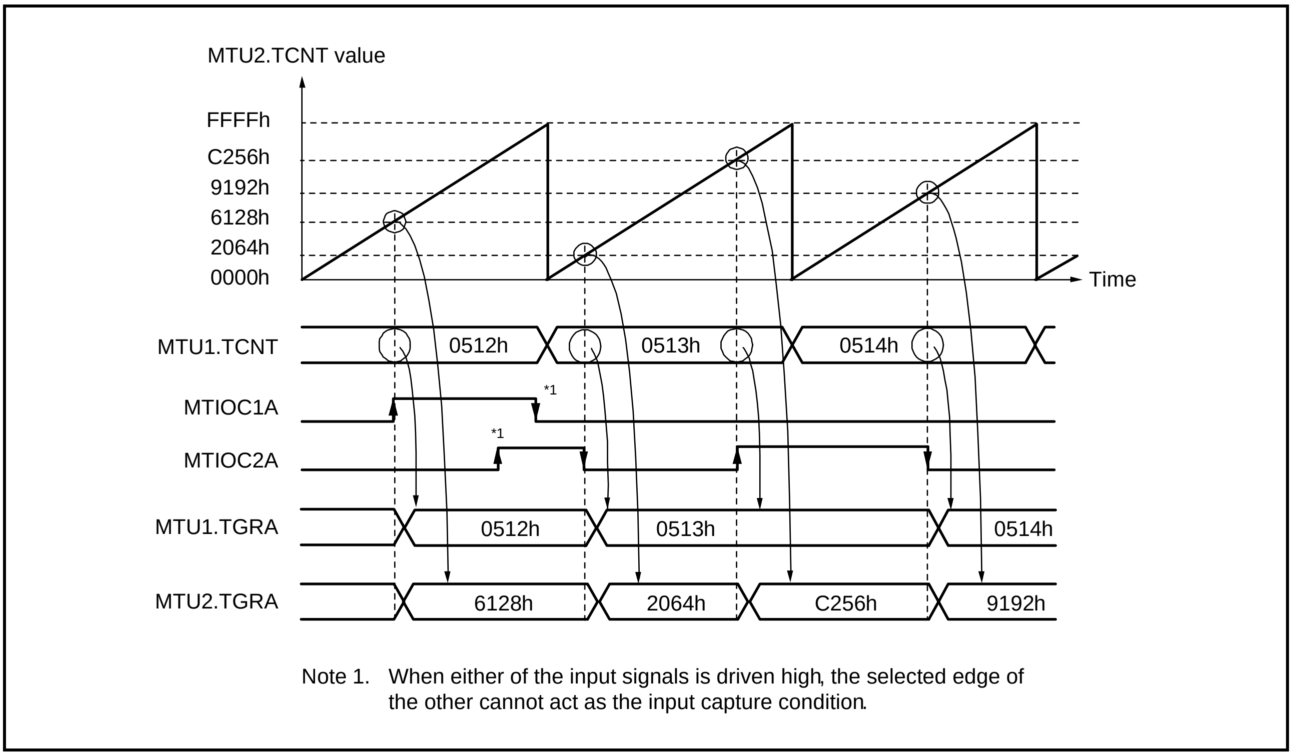


Figure 20.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 20.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

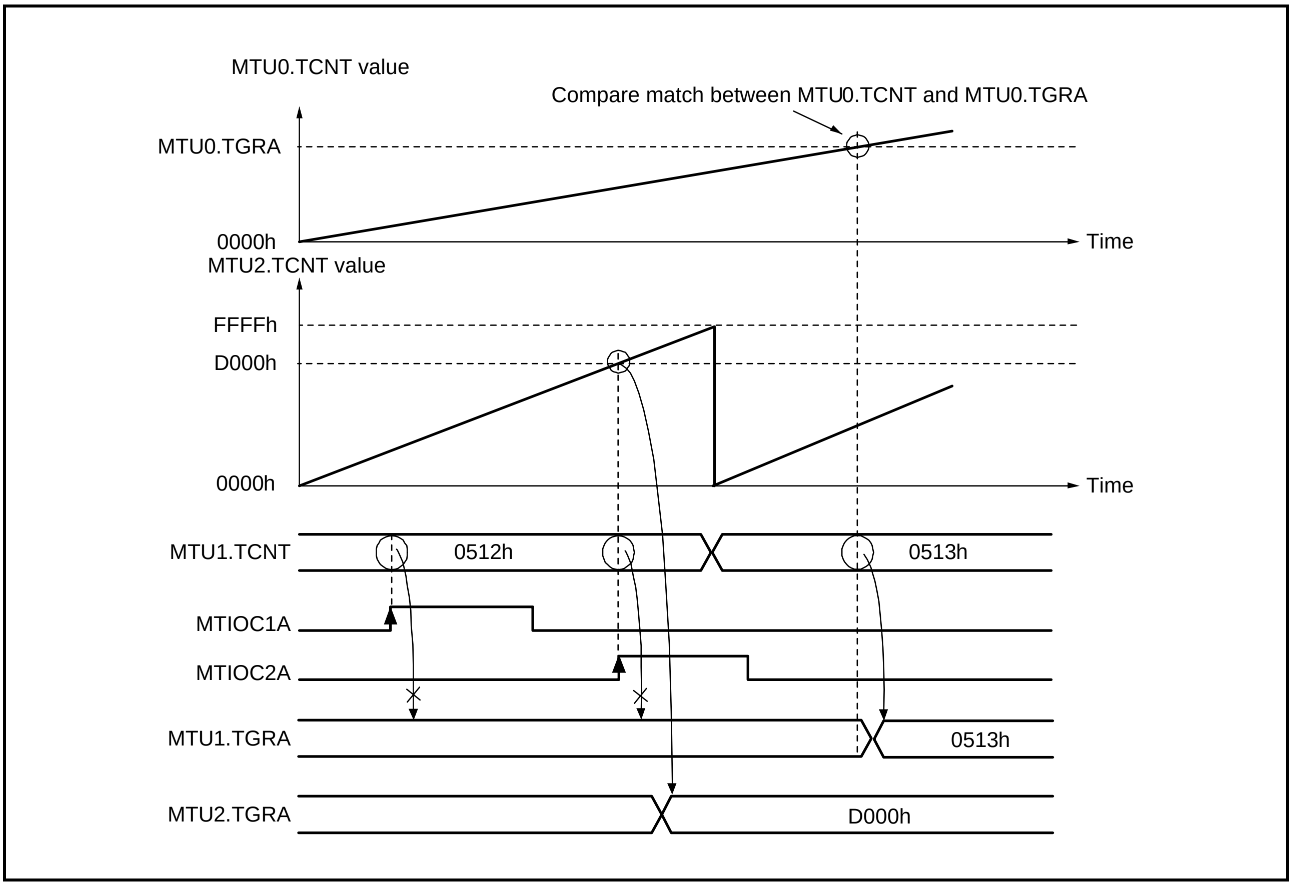


Figure 20.24 Cascaded Operation Example (d)

20.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register.
Every channel can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.
There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, PWM waveforms in up to eight phases can be output.

(b) PWM Mode 2

PWM waveform output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.
Up to eight phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.
The correspondence between PWM output pins and registers is shown in Table 20.50.

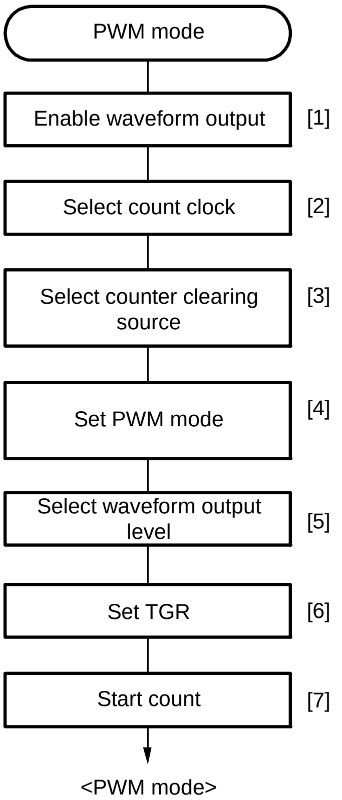
Table 20.50 PWM Output Registers and Output Pins

Channel	Register	Output Pins: PWM Mode 1	Output Pins: PWM Mode 2
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 20.25 shows an example of the PWM mode setting procedure.



[1] Enable TOERA output when outputting a waveform from the MTIOC pin of MTU3 and MTU4.

[2] Select the count clock with bits TPSC[2:0] in TCR. At the same time, select the input clock edge with bits CKEG[1:0] in TCR.

[3] Use bits CCLR[2:0] in TCR to select the TGR to be used as the TCNT clearing source.

[4] Select the PWM mode with bits MD[3:0] in TMDR.

[5] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[6] Set the cycle in the TGR selected in [3], and set the duty ratio in the other TGR.

[7] Set the CST bit in TSTRA to 1 to start the count operation.

Figure 20.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 20.26 shows an example of operation in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty ratio.

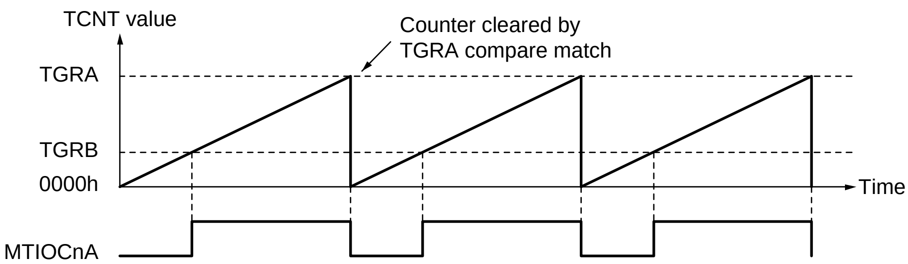


Figure 20.26 Example of PWM Mode 1 Operation (n = 0 to 4)

Figure 20.27 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 ratio.

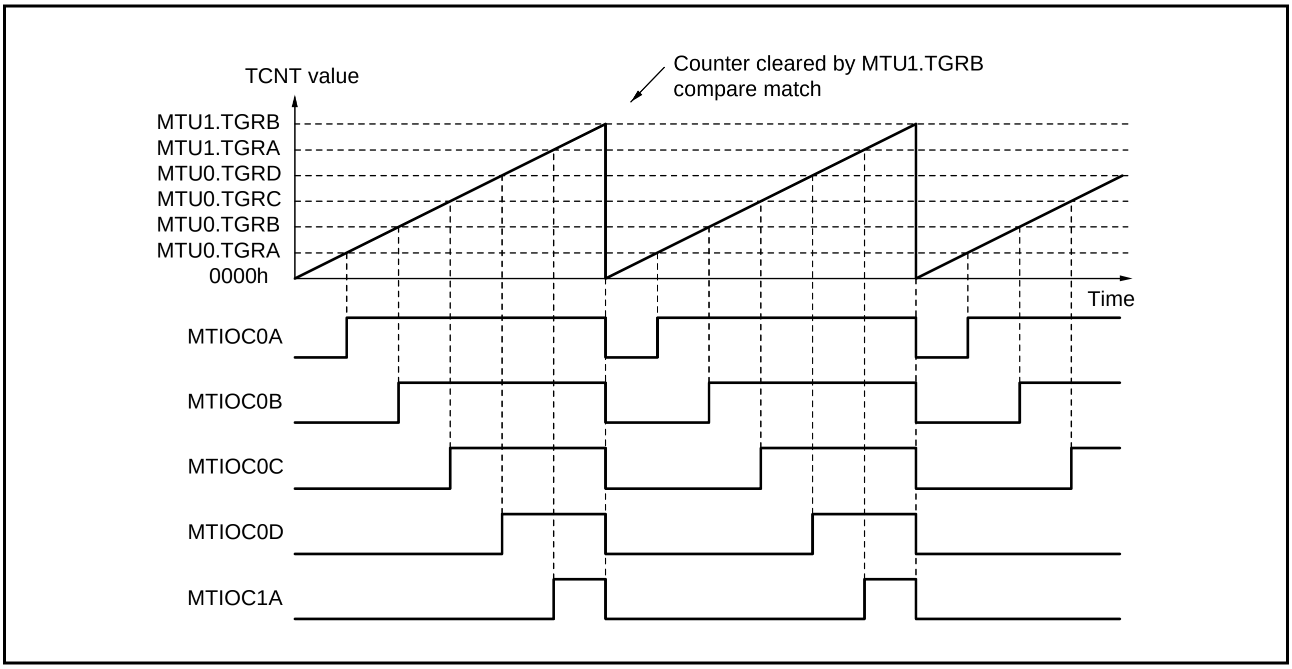


Figure 20.27 Example of PWM Mode 2 Operation

Figure 20.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

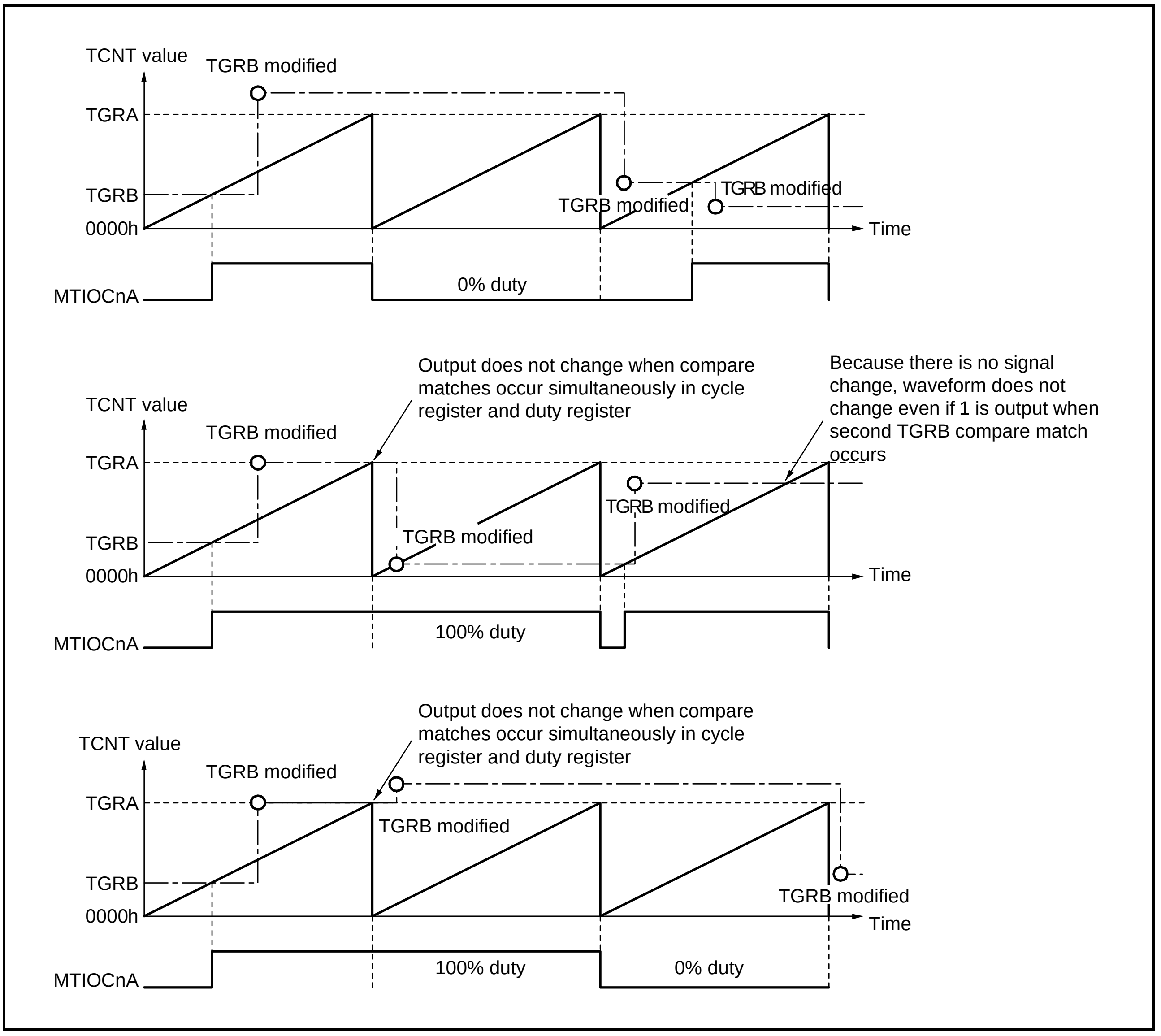


Figure 20.28 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4)

20.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.
In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.
Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.
Table 20.51 lists the external clock input pins to be connected in each phase counting mode.

Table 20.51 Clock Input Pins in Phase Counting Mode

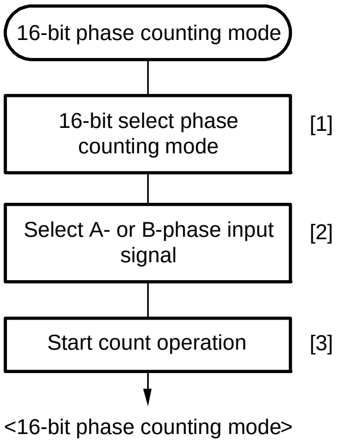
Phase Counting Mode	TMDR3.PHCKSEL bit	External Clock Input Pins	
		A-Phase	B-Phase
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

20.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.
In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.
When 16-bit phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0],TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.
These external input pins can be used for two-phase encoder pulse input.
When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

(1) Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 20.29 shows an example of the phase counting mode setting procedure.



[1] Set the TMDR3.LWA bit of MTU1 to 0.
Set the TMDR1.MD[3:0] bits to select the phase counting mode.

[2] When the MTU2 is set to phase counting mode, select the A- or B-phase input signal with the PHCKSEL bit in TMDR3.
This setting is not necessary for MTU1 because MTCLKA and MTCLKB are always selected for A-phase and B-phase, respectively.

[3] Set the CST bit in TSTRA to 1 to start the count operation.

Figure 20.29 Example of 16-Bit Phase Counting Mode Setting Procedure

(2) Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(a) Phase Counting Mode 1

Figure 20.30 shows an example of operation in phase counting mode 1, and Table 20.52 summarizes the TCNT up-counting and down-counting conditions.

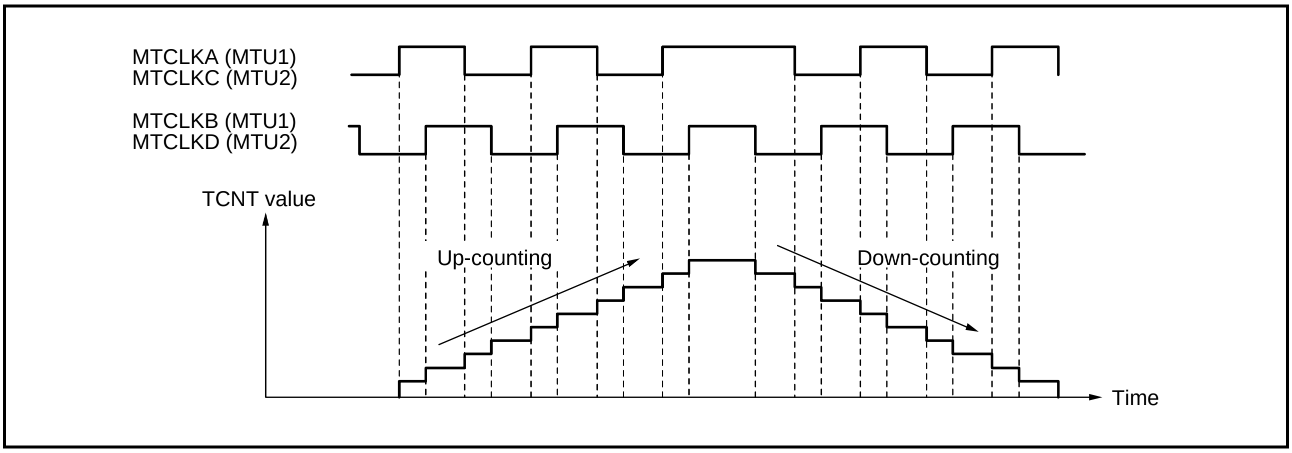


Figure 20.30 Example of Operation in Phase Counting Mode 1

Table 20.52 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 20.31 to Figure 20.33 show the examples of operation in phase counting mode 2 and Table 20.53 summarizes the TCNT up-counting and down-counting conditions.

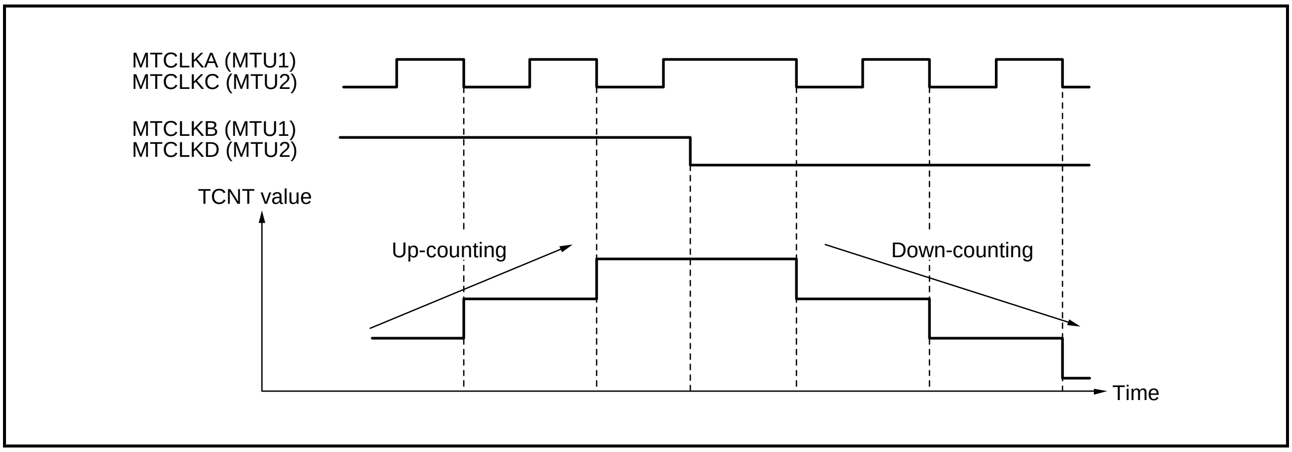


Figure 20.31 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

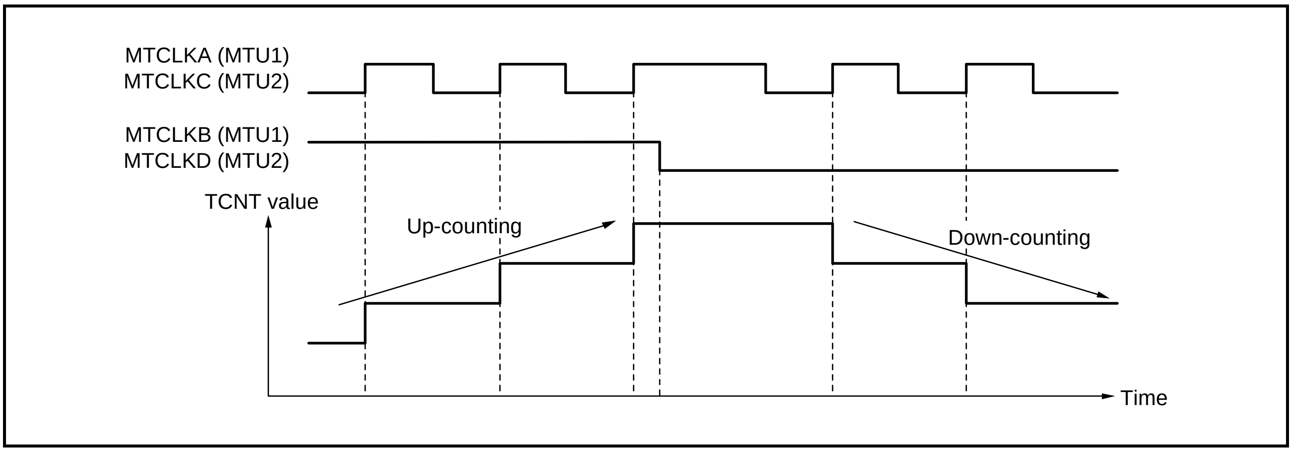


Figure 20.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

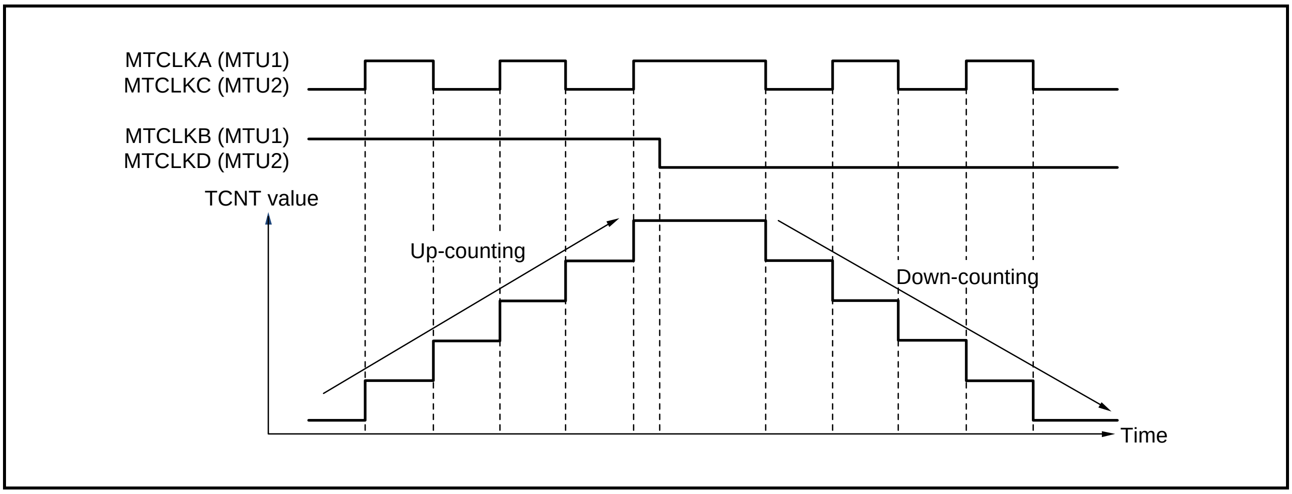


Figure 20.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 20.53 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	
	Low	↑	
	↑	High	
	↓	Low	Down-counting
01b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Not counted (Don't care)
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	Up-counting
	↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 20.34 to Figure 20.36 show the examples of operation in phase counting mode 3 and Table 20.54 summarizes the TCNT up-counting and down-counting conditions.

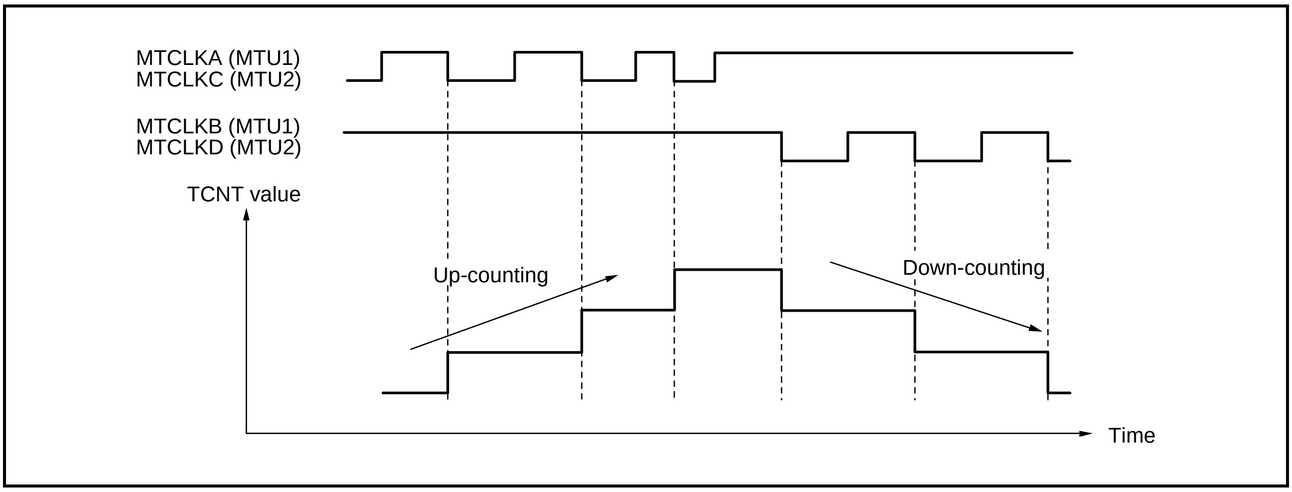


Figure 20.34 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

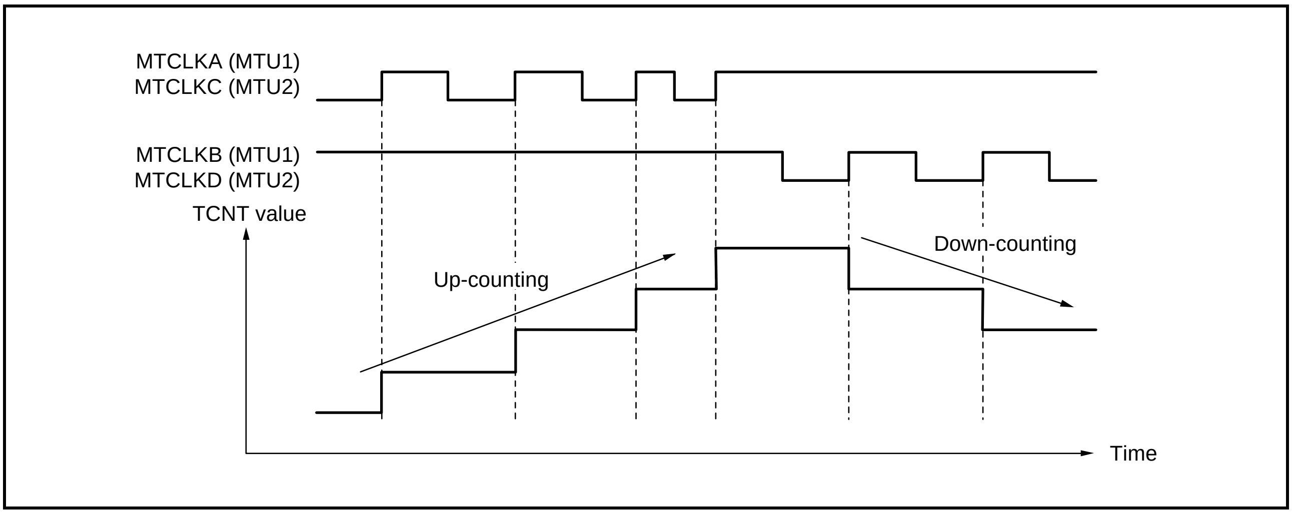


Figure 20.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

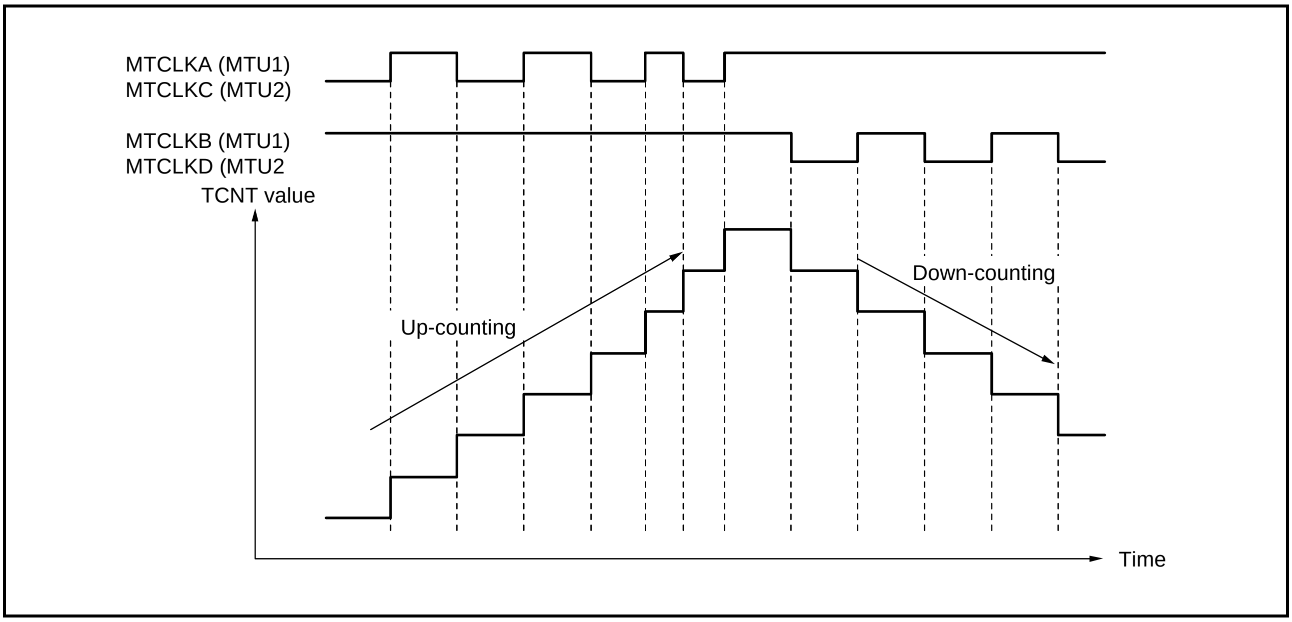


Figure 20.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 20.54 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	
01b	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 20.37 shows an example of operation in phase counting mode 4, and Table 20.55 summarizes the TCNT up-counting and down-counting conditions.

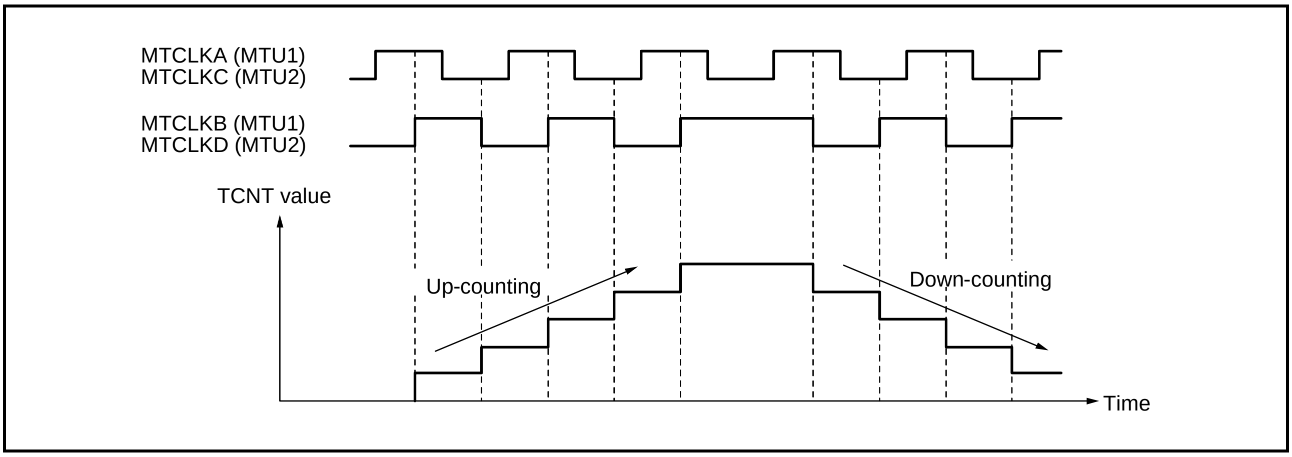


Figure 20.37 Example of Operation in Phase Counting Mode 4

Table 20.55 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

(e) Phase Counting Mode 5

Figure 20.38 and Figure 20.39 show the examples of operation in phase counting mode 5 and Table 20.56 summarizes the TCNT up-counting and down-counting conditions.

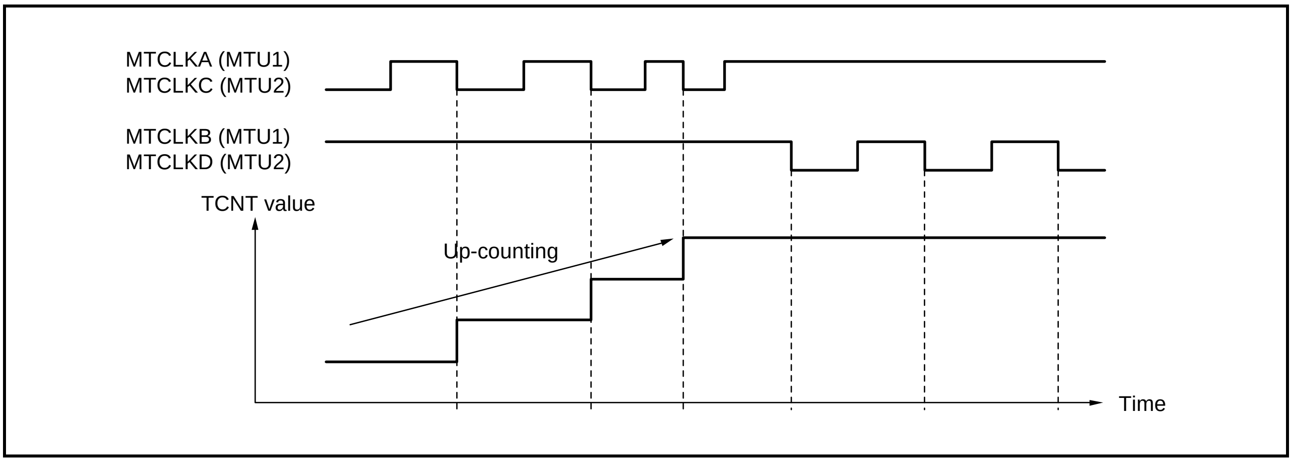


Figure 20.38 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

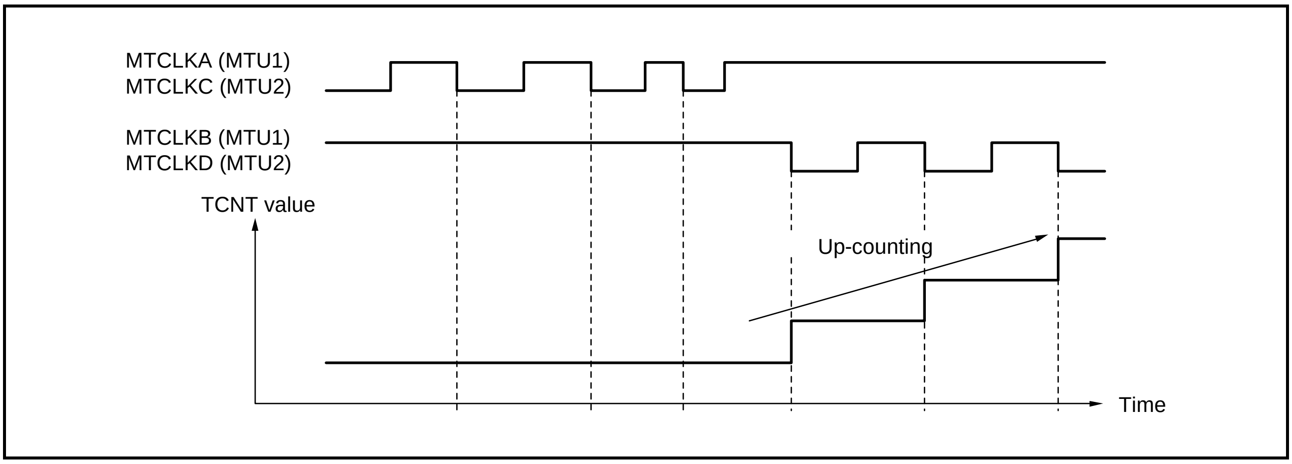


Figure 20.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 20.56 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Up-counting
1xb	High	↑	Not counted (Don't care)
	Low	↓	Up-counting
	↑	Low	Not counted (Don't care)
	↓	High	
	High	↓	Up-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	

↑ : Rising edge
↓ : Falling edge

(3) 16-Bit Phase Counting Mode Application Example

Figure 20.40 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 counter input clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

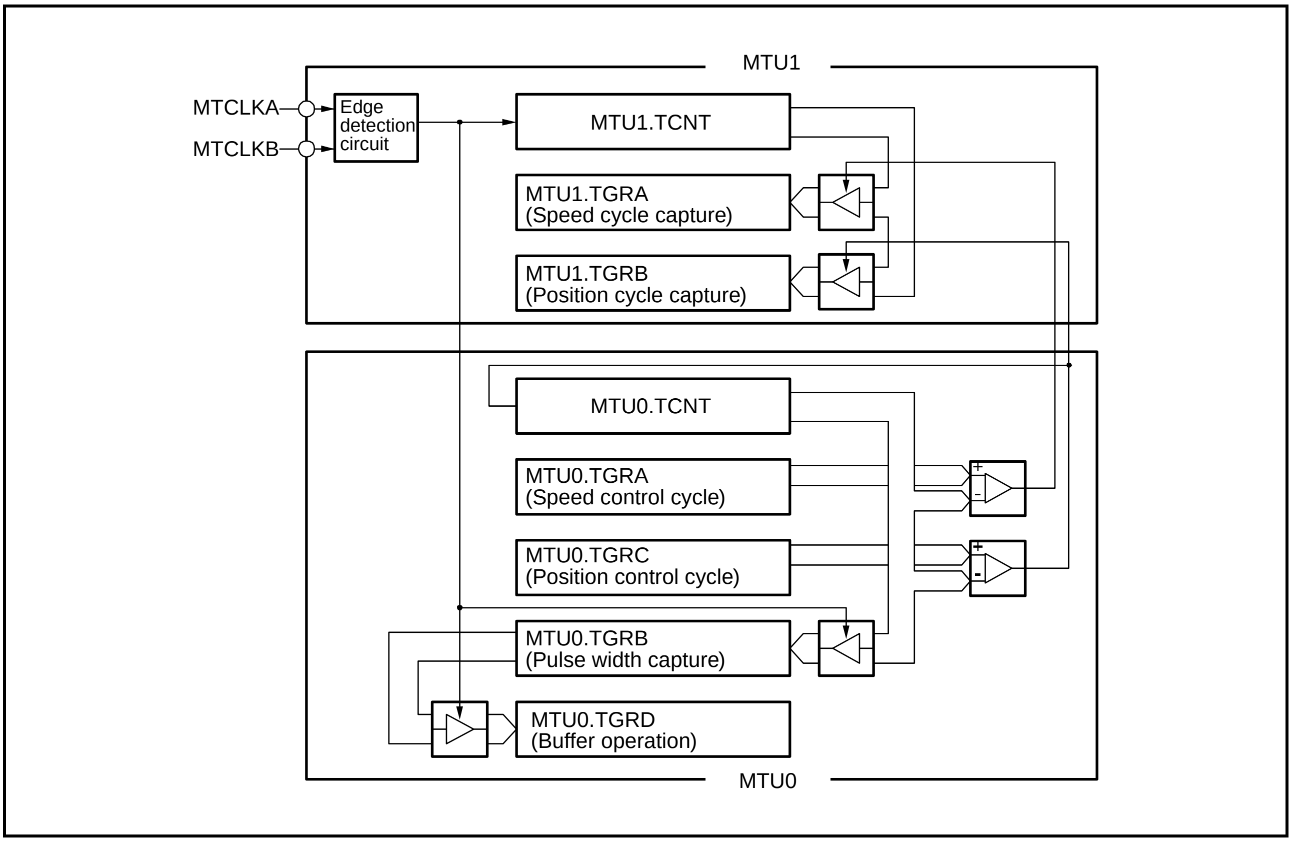


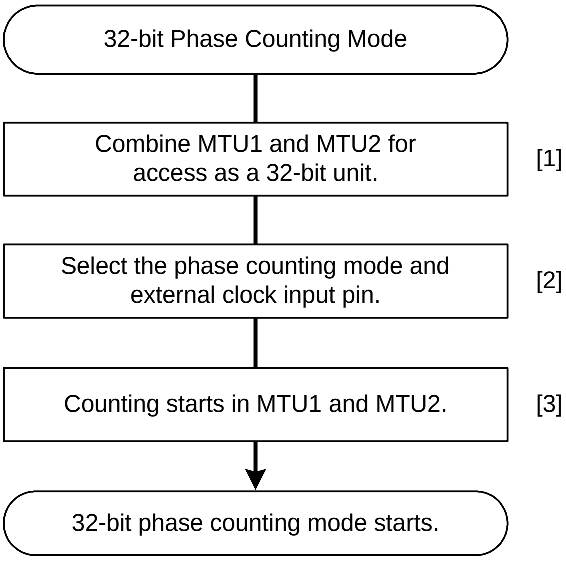
Figure 20.40 16-Bit Phase Counting Mode Application Example

20.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. Refer to Figure 20.41 for the procedure for setting cascade connection 32-bit phase counting mode.
Refer to section 20.3.4, Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 20.41 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.



[1] Set the LWA bit in MTU1.TMDR3 to 1 and combine MTU1 and MTU2 for access as a 32-bit unit.*1

[2] Select phase counting mode with the MD[3:0] bits in MTU1.TMDR1 and select the external clock input pin with the PHCKSEL bits in MTU1.TMDR3.

[3] Start counting in MTU1 and MTU2 by setting the TSTRA register to 00001111b.

Note 1. When the LWA bit in MTU1.TMDR3 is set to 1, the TCR, TCR2, TMDR, and TIOR registers in MTU1 are effective in controlling cascaded operation. In this case, the settings of these registers in MTU2 are ignored.

Figure 20.41 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

20.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms (six phases in total) 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 20.57 shows the PWM output pins used. Table 20.58 shows the settings of the registers.

Table 20.57 Output Pins for Reset-Synchronized PWM Mode

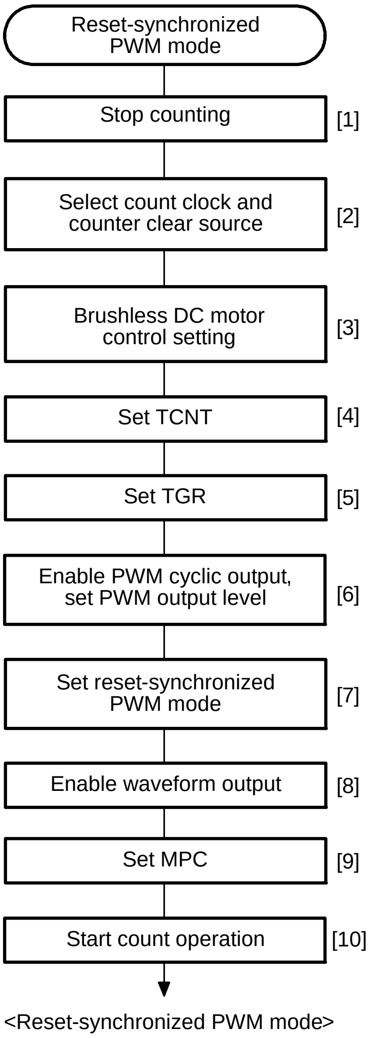
Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 20.58 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 20.42 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Clear the TSTRA.CST3 and TSTRA.CST4 bits to 0 to stop the TCNT count operation.
Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 to select the count clock and clock edge for MTU3.
Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set MTU3.TCNT and MTU4.TCNT to 0000h.

[5] MTU3.TGRA is the cycle register. Set the PWM cycle in MTU3.TGRA.
Set the PWM output transition timing in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB.
Note that the setting values should be within the range of compare match with MTU3.TCNT.

X ≤ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A register to enable or disable toggle output synchronized with the PMW cycle, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A and TOLBRA in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA).

[7] Set the MD[3:0] bits in MTU3.TMDR1 register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA register and the buffer operation (BFB = 1) of the MTU3.TGRB register when necessary. Set the MTU4.TMDR1 register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL registers to 00h.

[9] Set the MPU and the port mode register (PMR) of the I/O ports.

[10] Set the TSTRA.CST3 bit to 1 to start 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 ratio.

Figure 20.42 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 20.43 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.

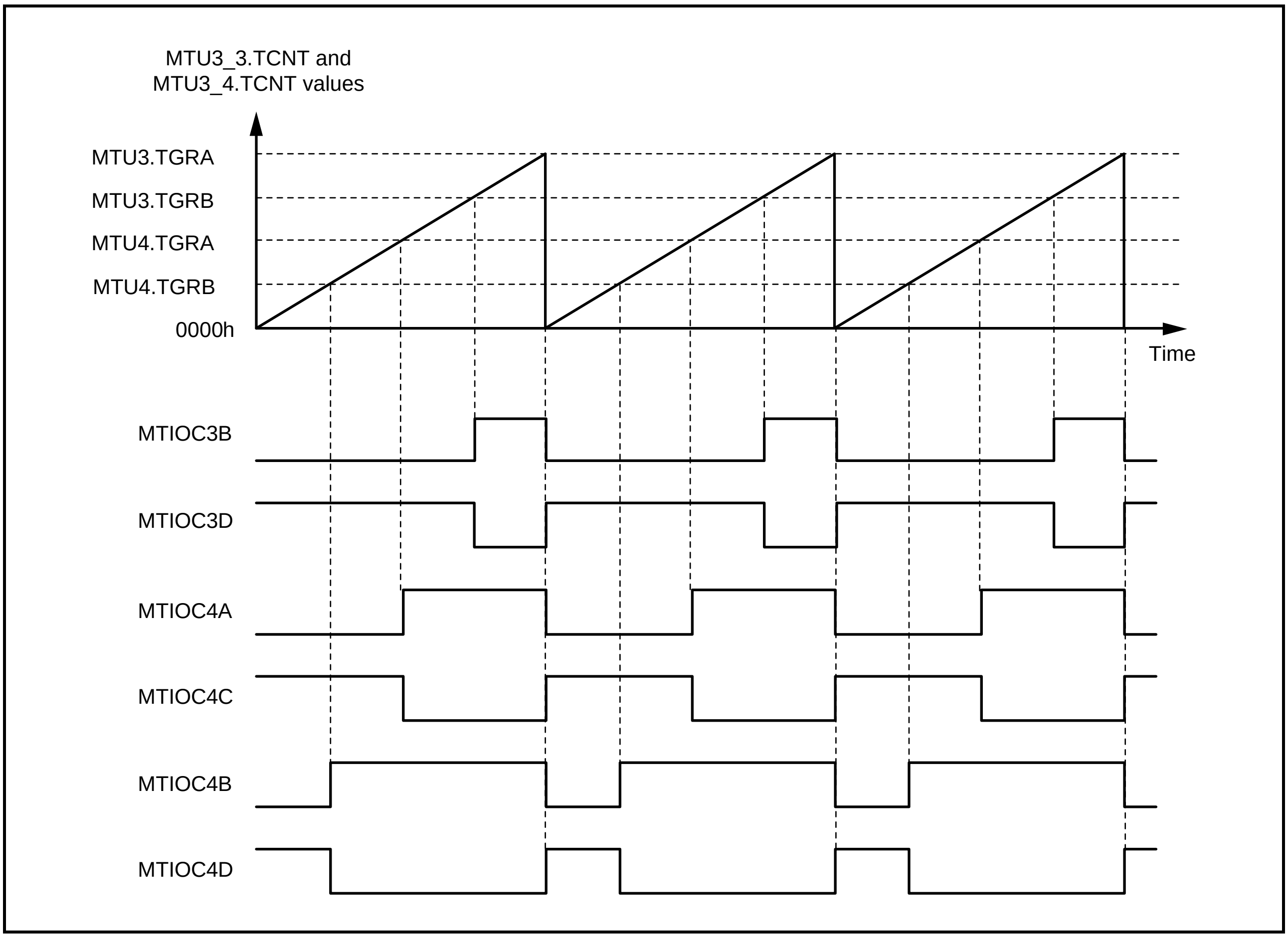


Figure 20.43 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

20.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.
Six positive-phase and three negative-phase PWM waveforms (six phases in total) with dead time can be output by combining MTU3/ MTU4. PWM waveforms without dead time can also be output.
In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle.
MTU3.TCNT and MTU4.TCNT function as up/down-counters.
Table 20.59 shows the PWM output pins used. Table 20.60 shows the settings of the registers used.
A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 20.59 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 1)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 1)

Note 1. Avoid setting the MTIOC3C pin as timer I/O pins in complementary PWM mode.

Table 20.60 Register Settings for Complementary PWM Mode (1/2)

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

Table 20.61 Register Settings for Complementary PWM Mode (2/2)

Channel	Counter/ Register	Description	Read/Write from CPU
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer cycle data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWERA setting*1
Timer cycle buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

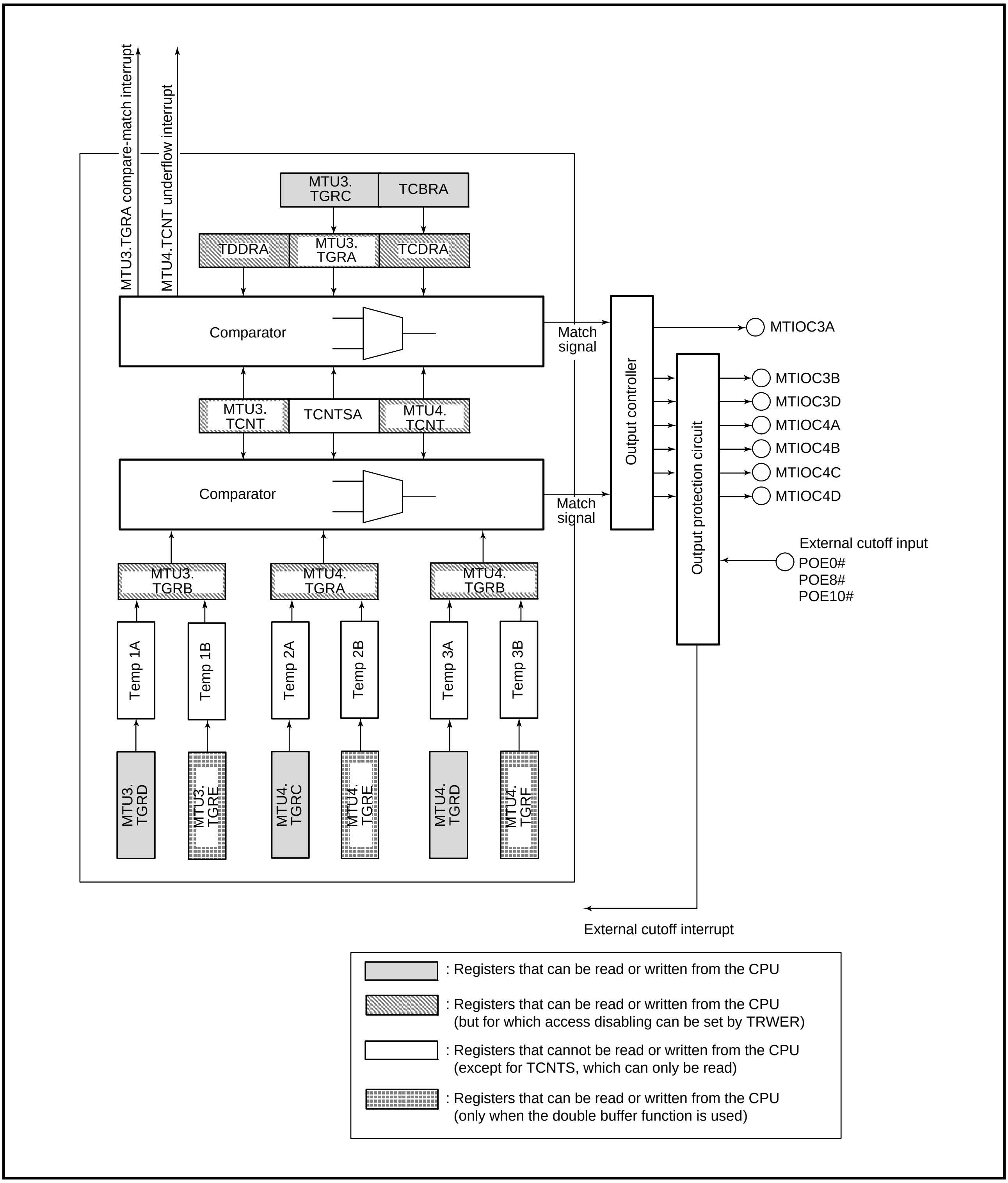
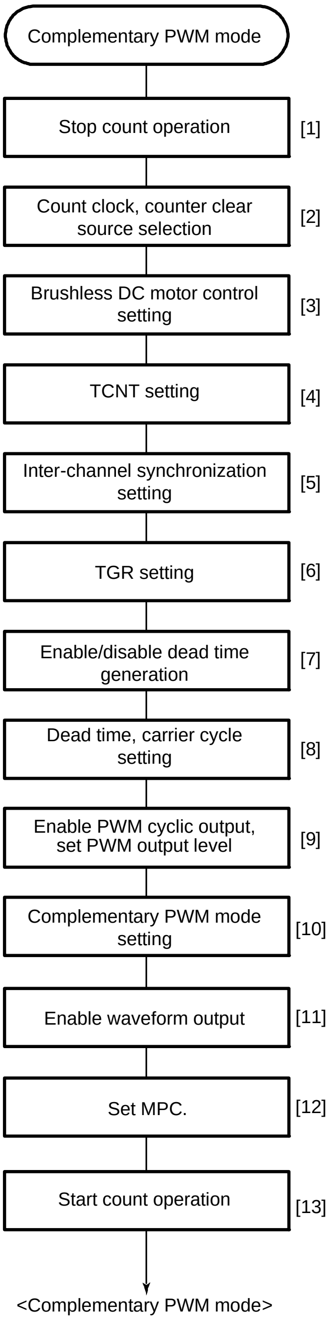


Figure 20.44 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 20.45 shows an example of the complementary PWM mode setting procedure.



[1] Clear the TSTRA.CST3 and TSTRA.CST4 bits to 0 to stop the TCNT count operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 to select the same count clock and clock edge for MTU3 and MTU4.
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0000h.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register to synchronize the channel generating the synchronous clear with MTU3 and MTU4.

[6] When MTU3 is used, set the PWM output duty ratio in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Be sure to set the same value in each pair of the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the PWM output duty value - 1 in the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).
Buffer registers when the double buffer function is used (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF)

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA register so that no dead time generated.

[8] Set the dead time in the TDDRA register, 1/2 the PWM cycle in the TCDRA register and TCBRA register, and 1/2 the PWM cycle plus the dead time in MTU3.TGRA and MTU3.TGRC.
When no dead time generation is selected, set 1 in the TDDRA register and 1/2 the PWM cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[9] Set the PSYE bit in the TOCR1A register to enable or disable toggle output synchronized with the PWM cycle, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A and TOLBRA in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA).

[10] Select complementary PWM mode in MTU3.TMDR1. Specify the buffer operation (BFA = 1) of MTU3.TGRA and the buffer operation (BFB = 1) of MTU3.TGRB.
Only when the double buffer function is used, set the TMDR2A register to enable the double buffer function.

[11] Set the TOERA register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL to 00h.

[12] Set the MPU and the port mode register (PMR) of the I/O ports.

[13] Set the TSTRA.CST3 and TSTRA.CST4 bits to 1 simultaneously to start count operation.

Figure 20.45 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 20.46 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 20.47 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA —in each unit perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDRA when complementary PWM mode is selected and the CST bit in TSTRA 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 MTU4.TCNT value matches 0000h, MTU3.TCNT switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h after a reset. When the CST bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when MTU3.TCNT matches MTU3.TGRA. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way. TCNTSA is a read-only counter. It does not need to be initialized after a reset.

In counting up by MTU3.TCNT and MTU4.TCNT, MTU3.TCNT starts counting up when it matches TCDRA and switches to counting down when it matches MTU3.TGRA. Furthermore, when MTU4.TCNT matches TDDRA, TCNTSA is set to the value in MTU3.TGRA and counting is stopped.

When MTU4.TCNT matches TDDRA during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTSA starts up-counting, and when MTU4.TCNT matches 0000h, the operation switches to down-counting. When MTU3.TCNT matches TCDRA, TCNTSA is cleared to 0000h and stops counting.

TCNTSA is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

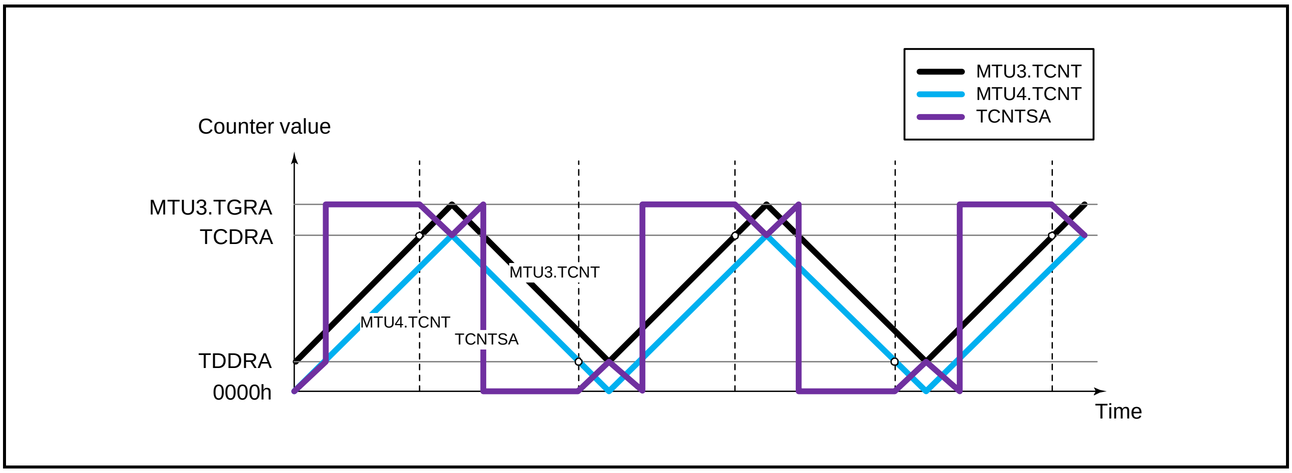


Figure 20.46 Counter Operation in Complementary PWM Mode

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 20.47 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control register (TOCR1A) is output from the PWM output pin.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF are also used as buffer registers B. For details of double buffer operation, refer to section 20.3.8 (2) (r), Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA register and MTU3.TGRC register, which operate as buffer registers for the timer cycle registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA while TCNTSA is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTSA is counting down). The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 20.47 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 20.47), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

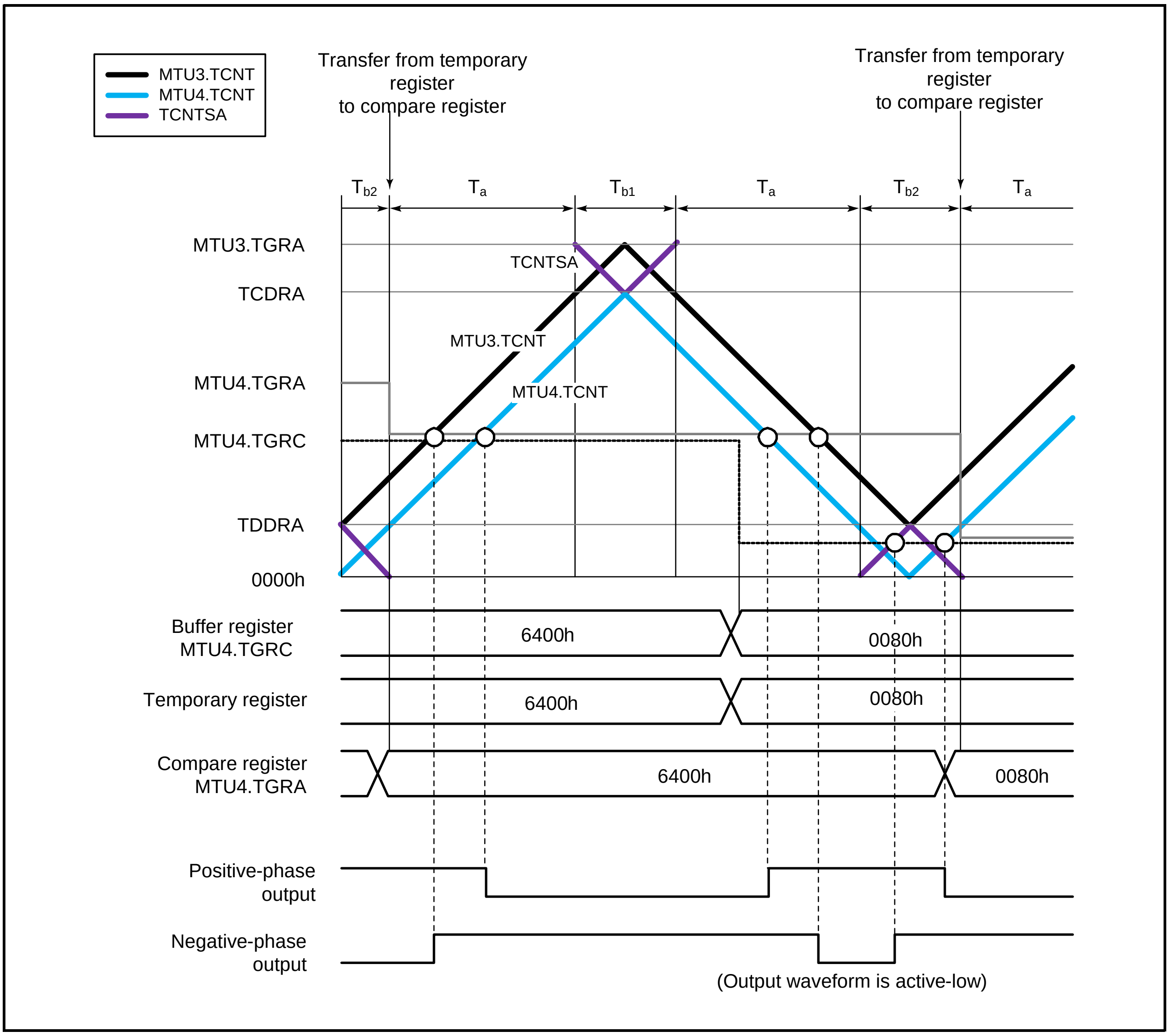


Figure 20.47 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(c) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] bits, initial values should be set in the following registers.
The TOCR1A and TOCR2A registers are used to set the PWM output level. MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM cycle + dead time Td. The timer cycle buffer register (TCBRA) operates as the buffer register for the timer cycle data register (TCDRA), and should be set with 1/2 the PWM cycle. Set dead time Td in the timer dead time data register (TDDRA).
When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA) should be cleared to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM carrier cycle + 1, and TDDRA should be set to 1.
Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Set the respective (initial PWM duty – 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF) only when the double buffer function is used.
The values set in the five buffer registers excluding TDDRA 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 20.62 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
TOCR1A, TOCR2A	PWM output level
MTU3.TGRC	1/2 PWM cycle + dead time Td (1/2 PWM cycle + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA	1/2 PWM cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF	Initial PWM duty ratio – 1 value for each phase (only when double buffer function is used)
MTU4.TCNT	0000h

Note: The value set in MTU3.TGRC should be the sum of 1/2 the PWM cycle set in TCBRA and dead time Td set in TDDRA. When dead time generation is disabled by TDERA, TGRC should be set to 1/2 the PWM cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A).
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.
The dead time is set in the timer dead time data register (TDDRA). The value set in TDDRA 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 TDDRA.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDERA) to 0. TDERA 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 cycle + 1 and the timer dead time data register (TDDRA) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 20.48 shows an example of operation without dead time (MTU3 and MTU4).

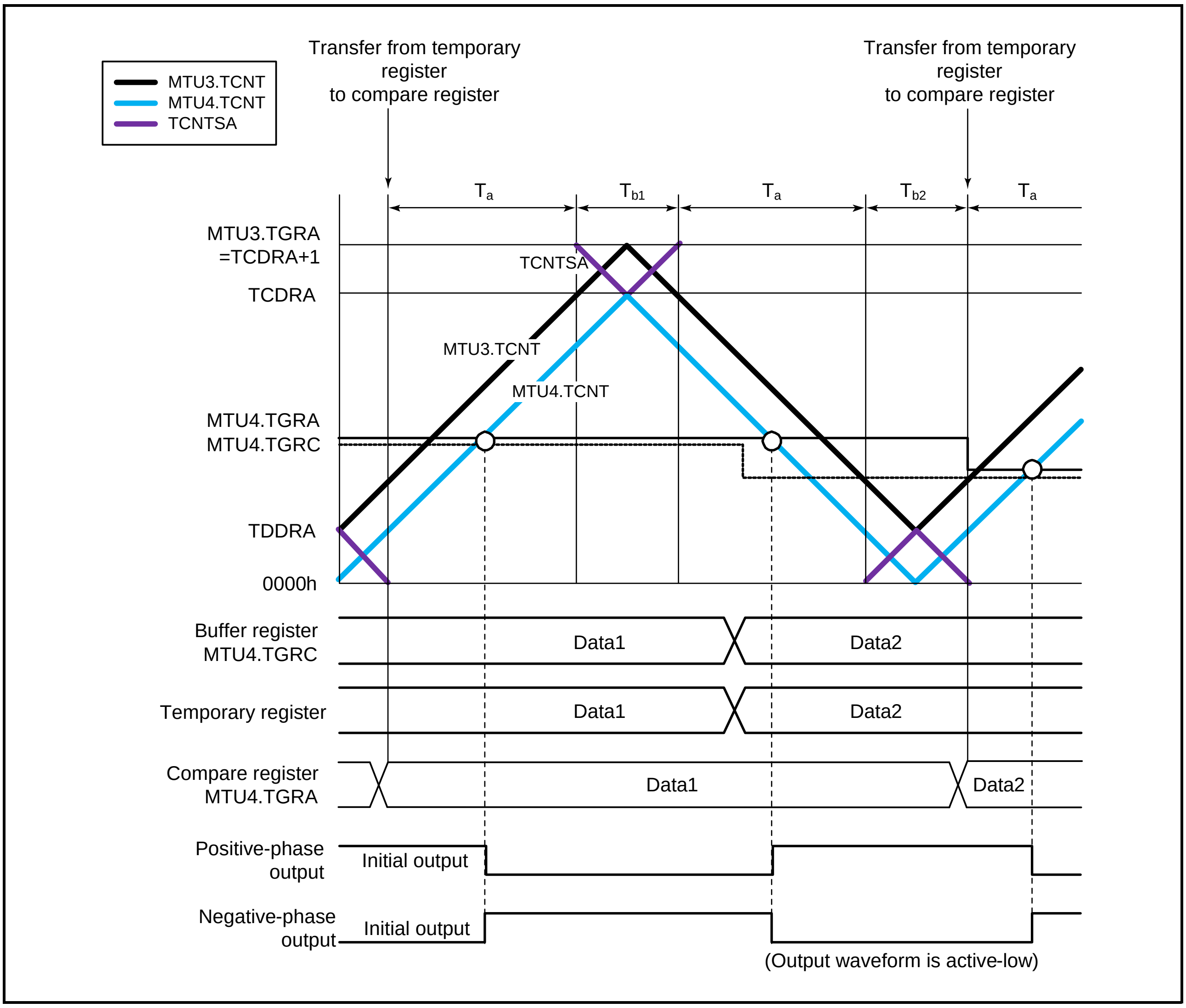


Figure 20.48 Example of Operation without Dead Time (MTU3 and MTU4)

(g) PWM Cycle Setting

In complementary PWM mode, the PWM cycle is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDRA, 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 = TCDRA setting + TDDRA setting
Without dead time: MTU3.TGRA setting = TCDRA setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA register and the TDDRA register:

TCDRA setting > TDDRA setting × 2 + 2

The MTU3.TGRA and TCDRA settings are made by setting values in buffer registers MTU3.TGRC and TCBRA. When data is written to MTU4.TGRD to enable transfers, the values set in MTU3.TGRC and TCBRA are transferred simultaneously to the MTU3.TGRA and TCDRA with the transfer timing selected with the MTU3.TMDR1.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 20.49 illustrates the operation when the PWM cycle is updated at the crest.
Refer to the following section, (h), Register Data Updating, for the method of updating the data in each buffer register.

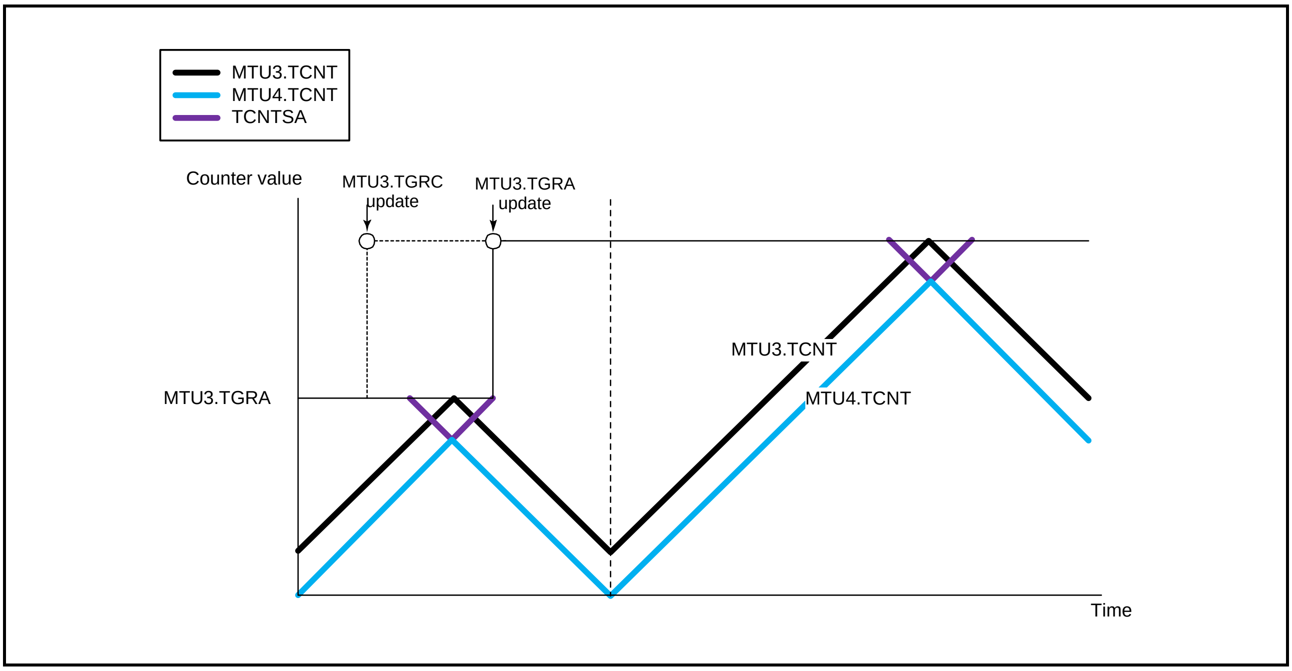


Figure 20.49 Example of PWM Cycle Updating (MTU3 and MTU4)

(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 PWM 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 TCNTSA is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA is counting; in this case, the value written to a buffer register is transferred after TCNTSA halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] bits. Figure 20.50 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD 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.

Refer to section 20.3.8 (2) (r), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

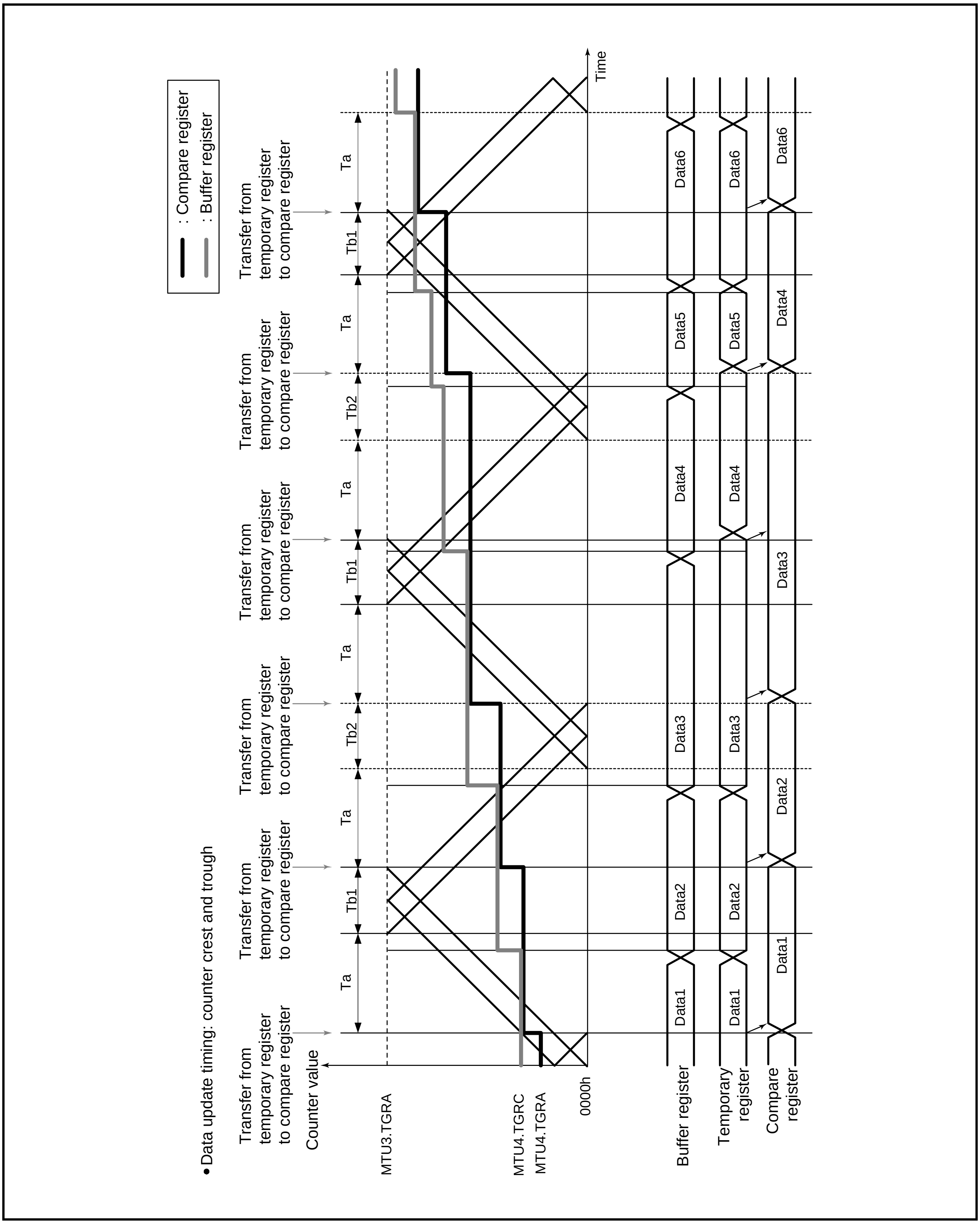


Figure 20.50 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register.
This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 until MTU4.TCNT exceeds the value set in the TDDRA register. Figure 20.51 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA value is shown in Figure 20.52.

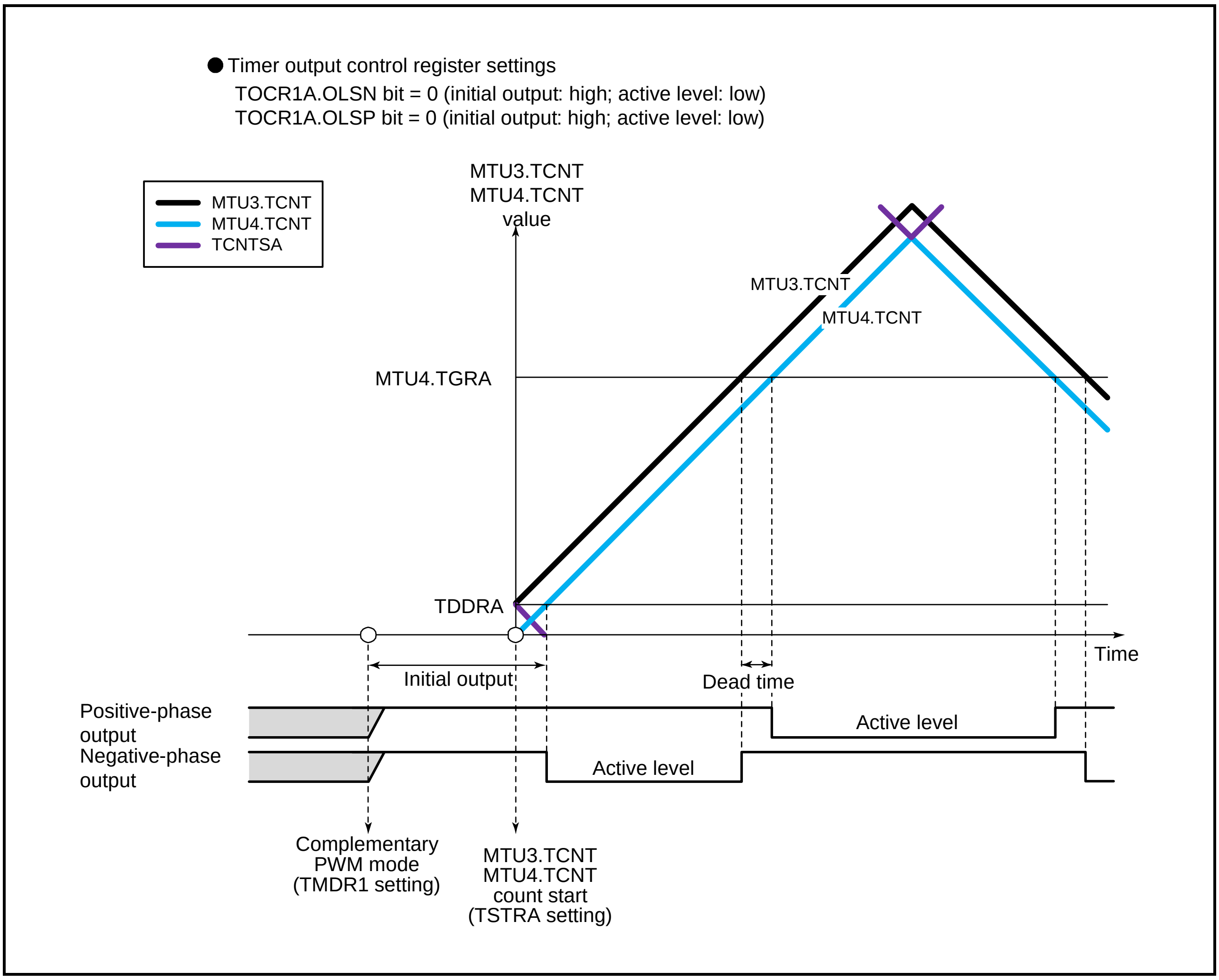


Figure 20.51 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

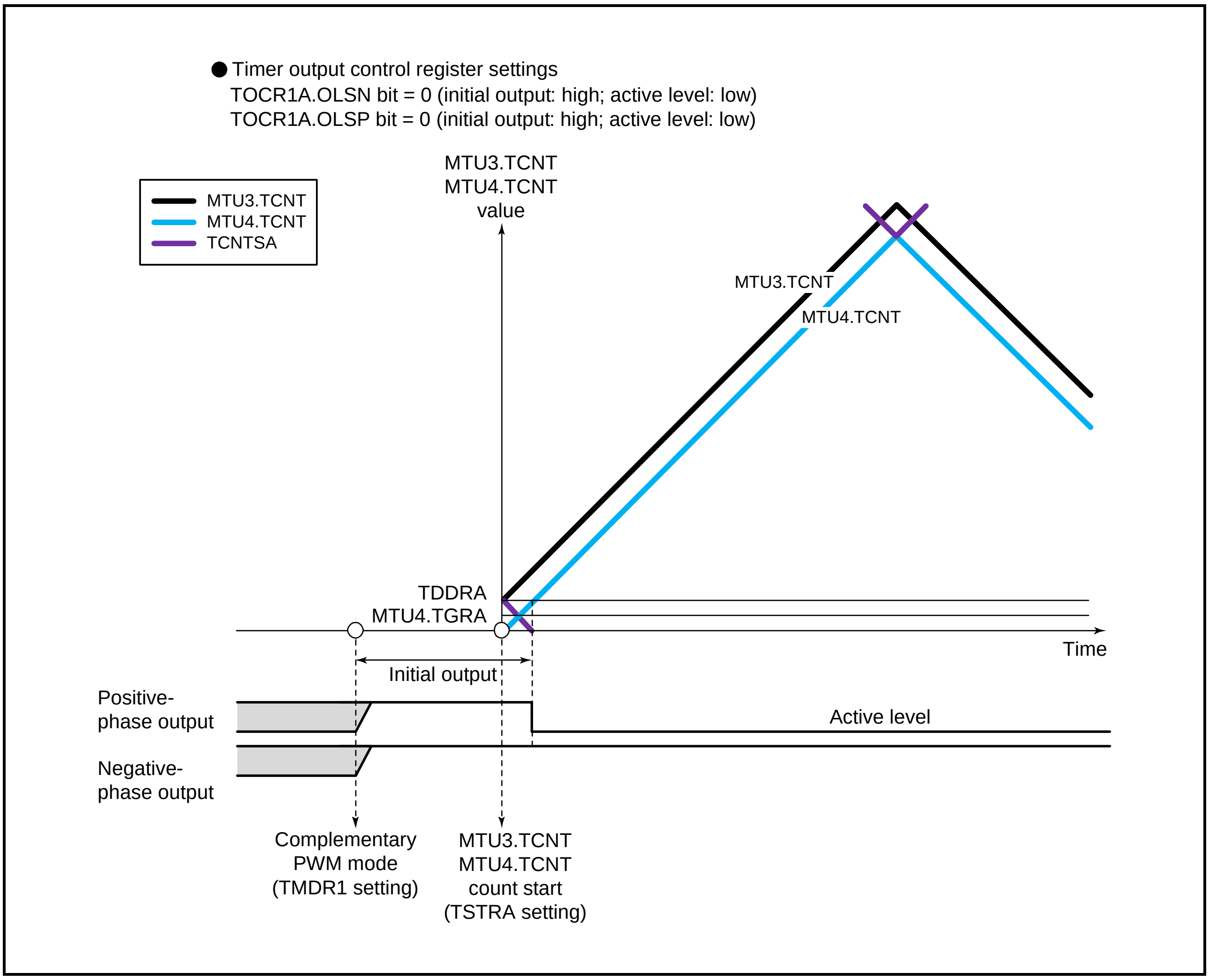


Figure 20.52 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.
A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTSA is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output from 0 to 100% duty ratio.
The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 20.53 to Figure 20.55 show examples of waveform generation in complementary PWM mode.
The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.
In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 20.53.
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 20.54, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).
Similarly, in the example in Figure 20.55, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.
Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

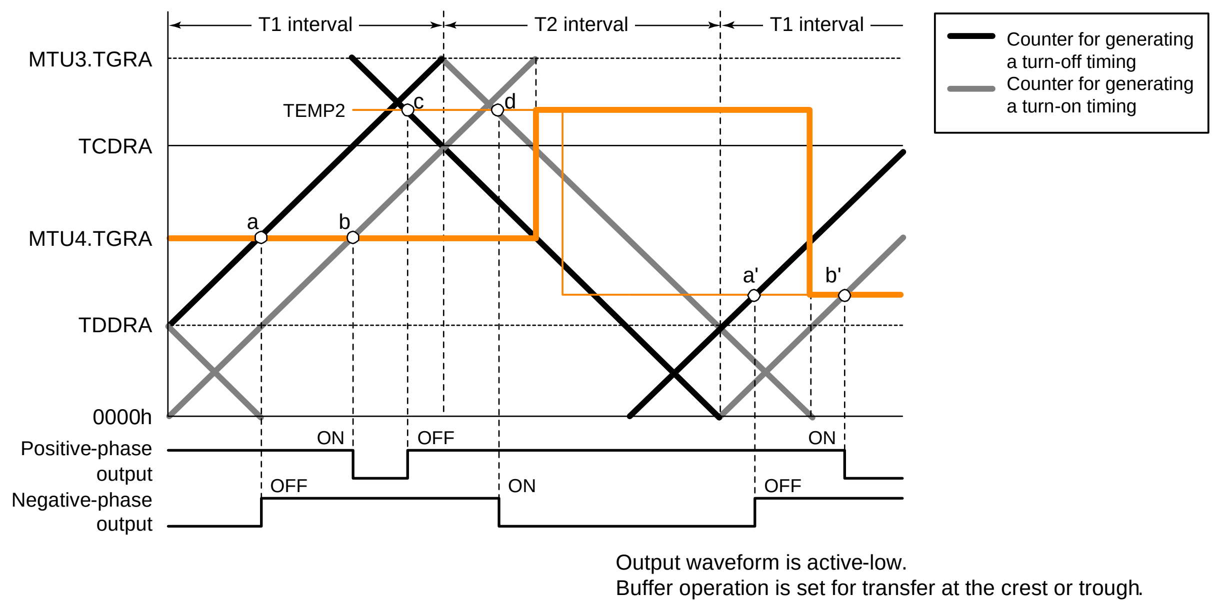


Figure 20.53 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

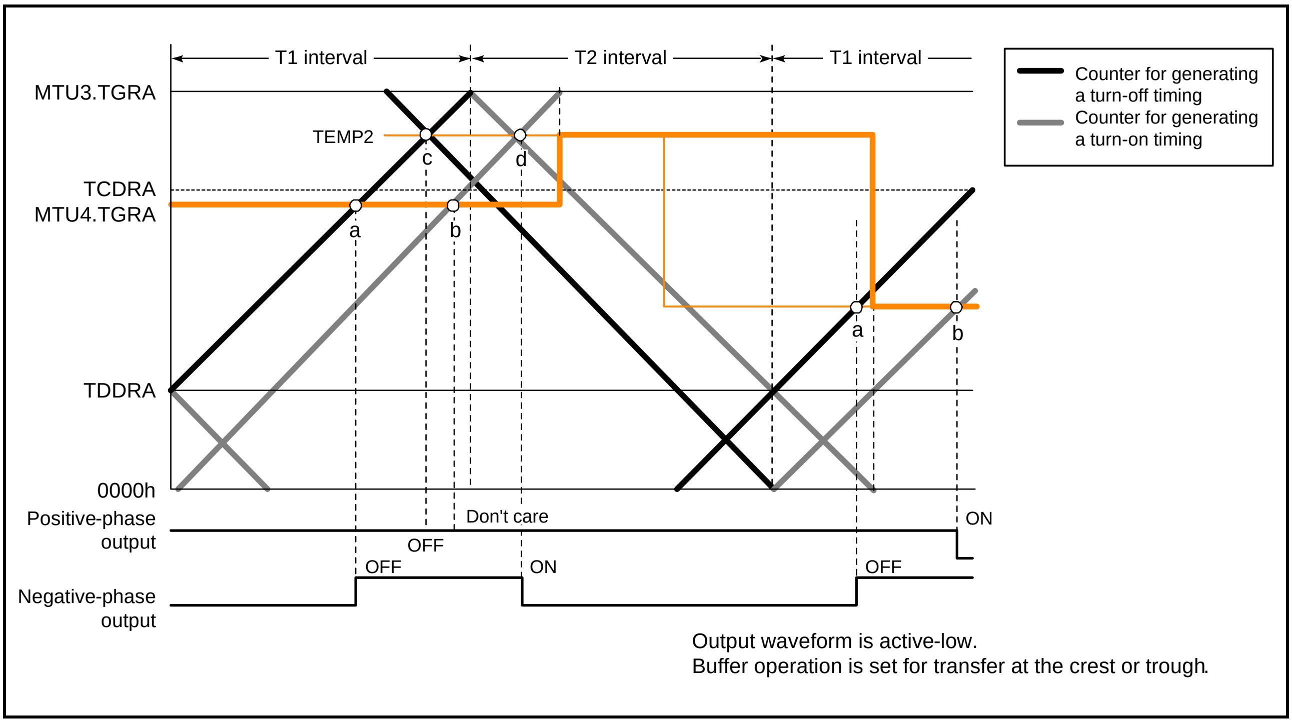


Figure 20.54 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

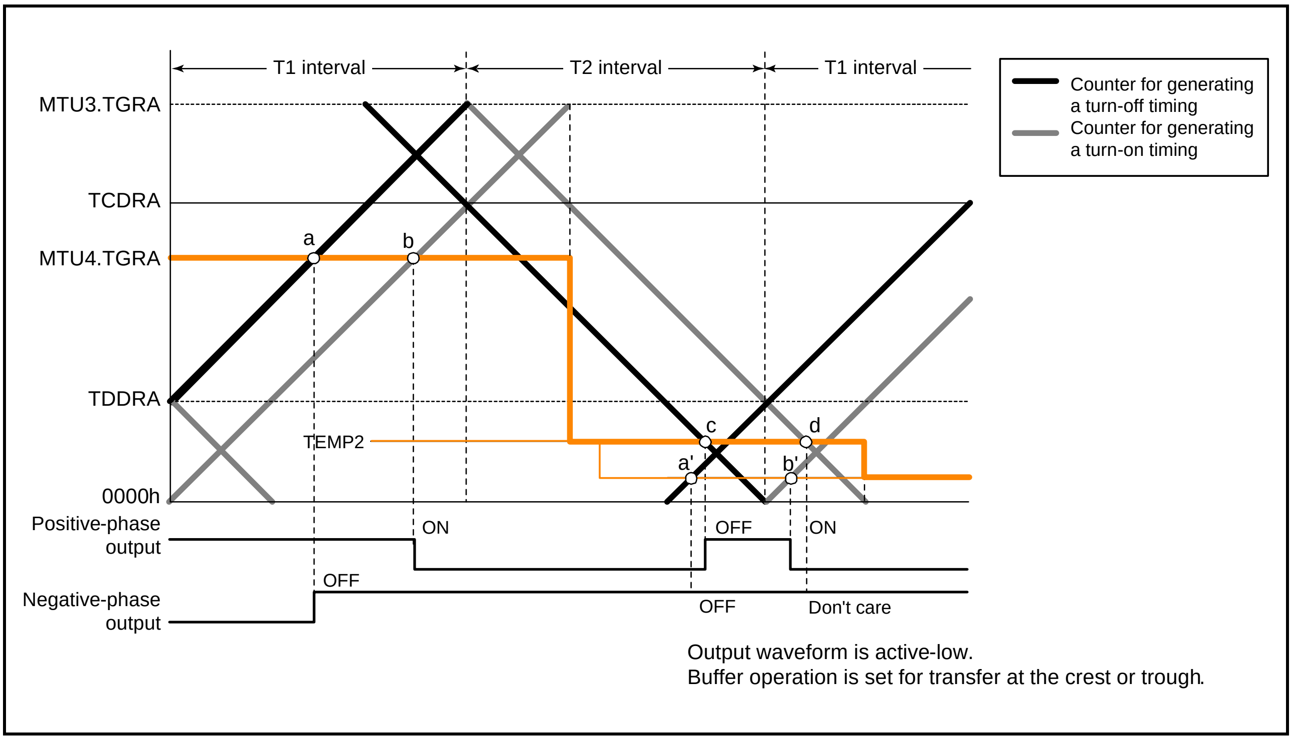


Figure 20.55 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. Figure 20.56 to Figure 20.60 show output examples.

A 100% duty waveform is output when the compare register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA. The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

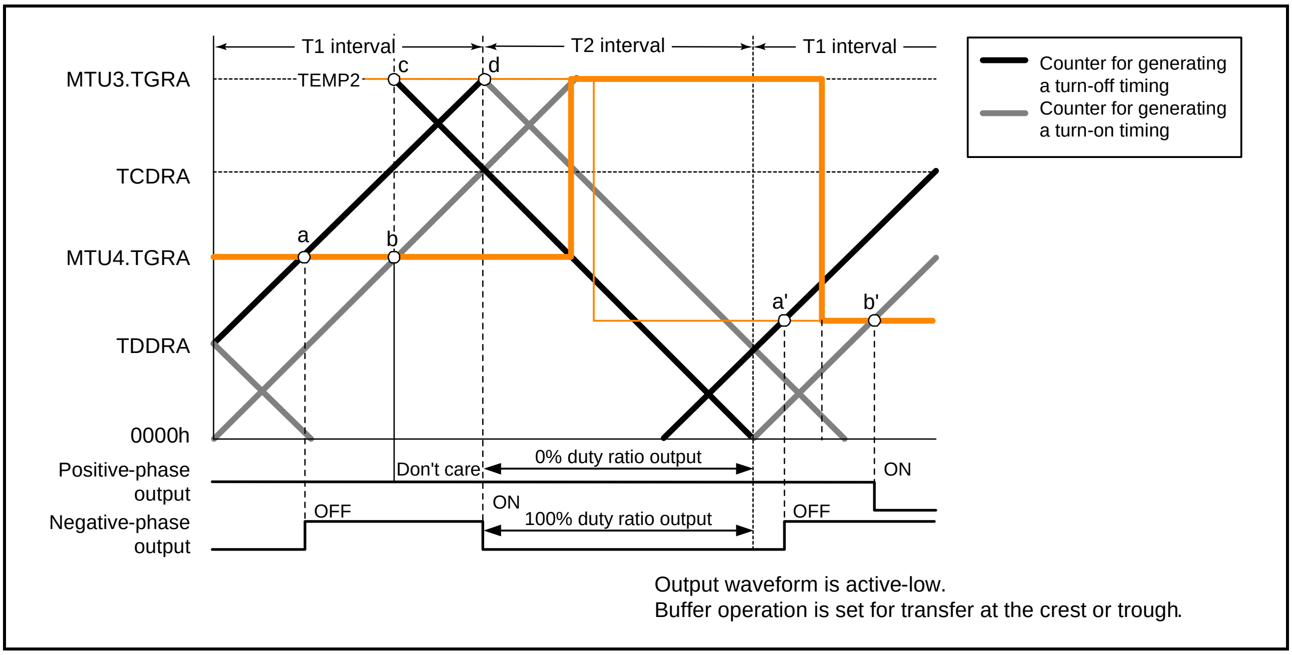


Figure 20.56 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

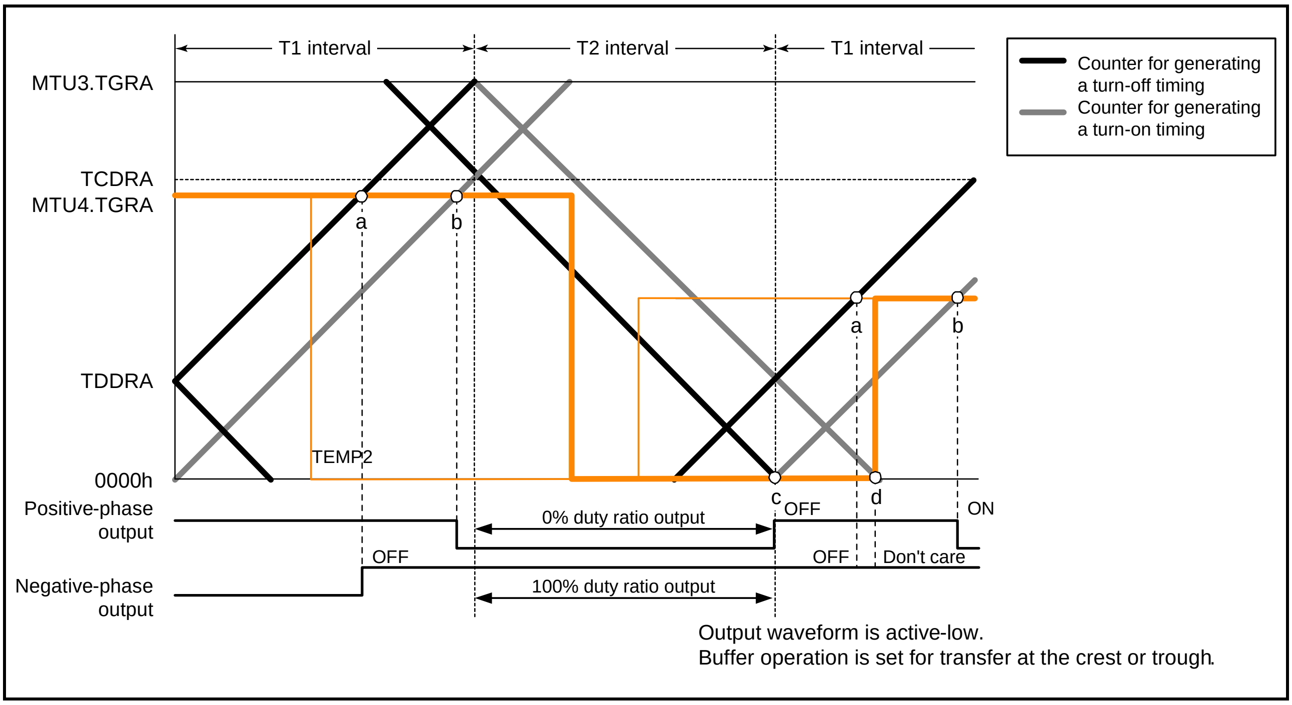


Figure 20.57 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

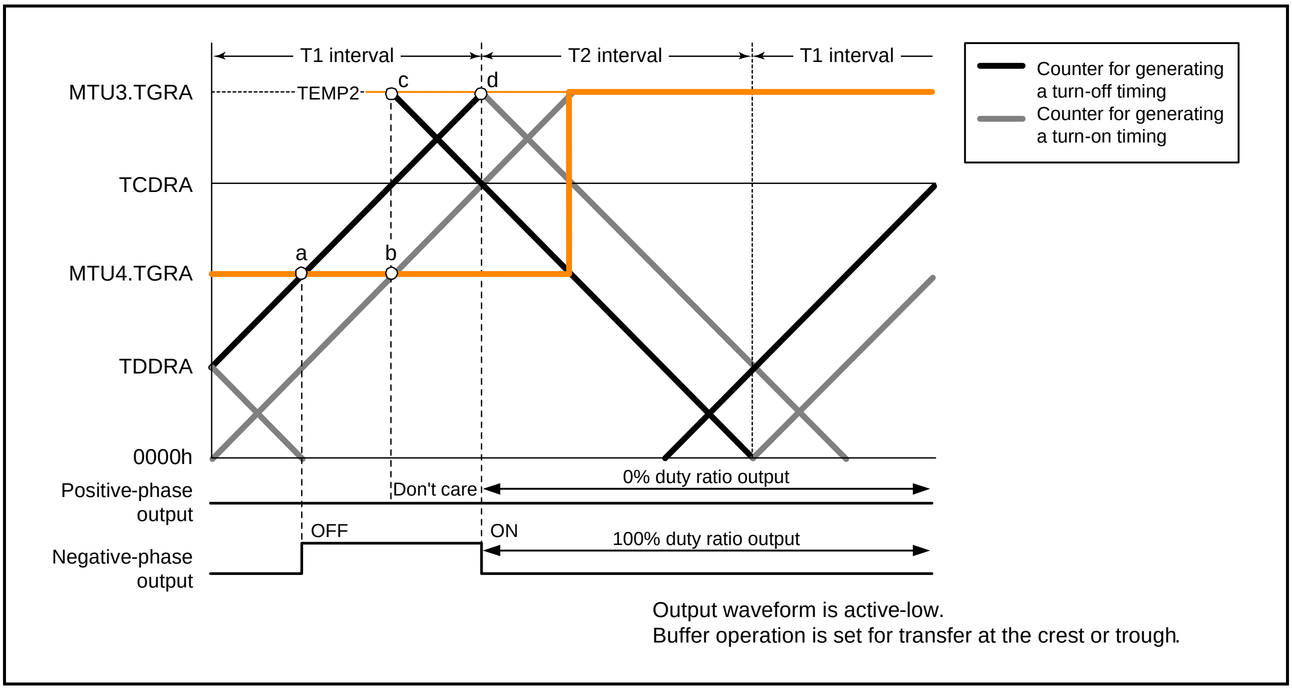


Figure 20.58 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

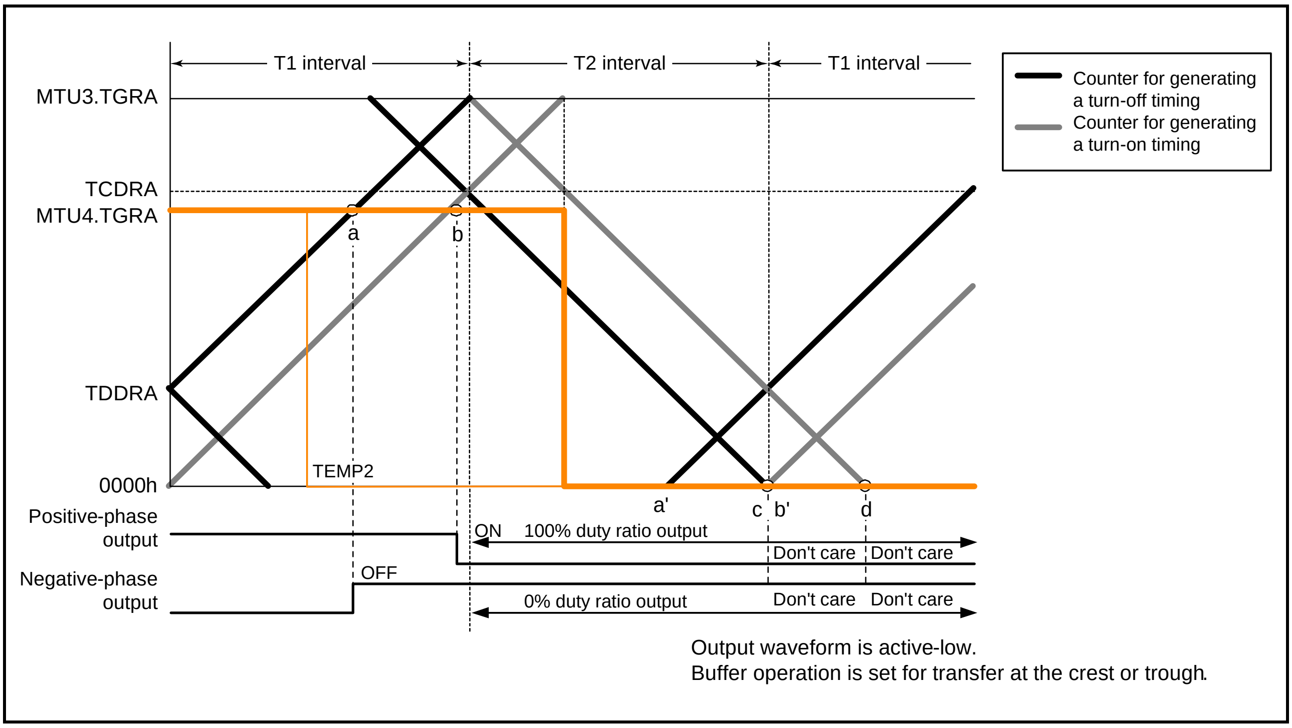


Figure 20.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

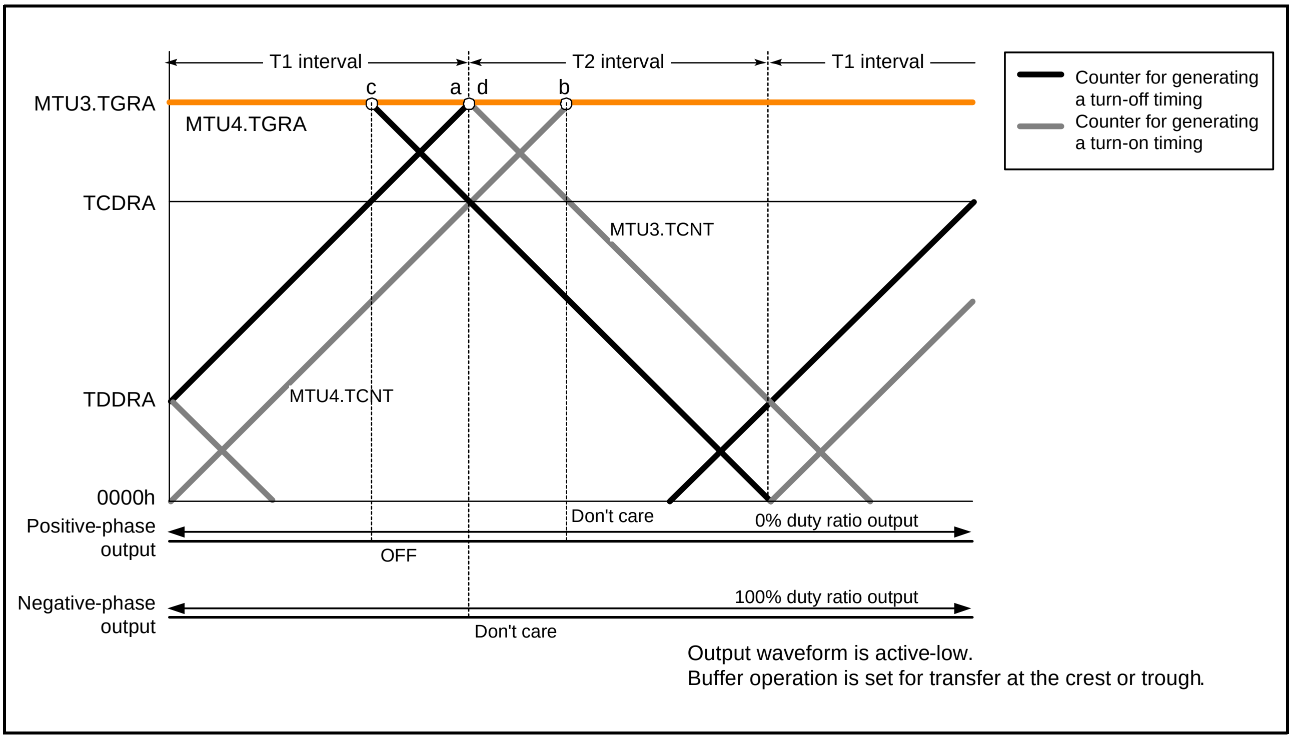


Figure 20.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM cycle can be enabled by setting the PSYE bit in the TOCR1A register to 1. An example of a toggle output waveform is shown in Figure 20.61.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRAand a compare match between MTU4.TCNT and 0000h.

pin is assigned for this toggle output. The initial output is high-level output.

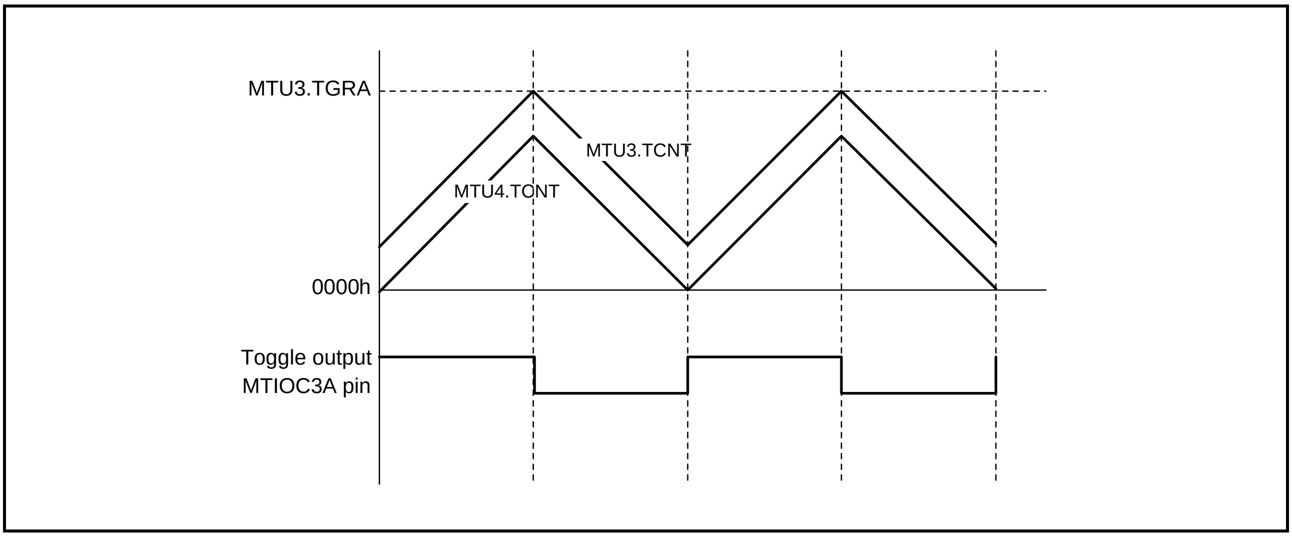


Figure 20.61 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] bits.
Figure 20.62 illustrates an example of this operation.
Use of this function enables a counter to be cleared and restarted through an external signal.

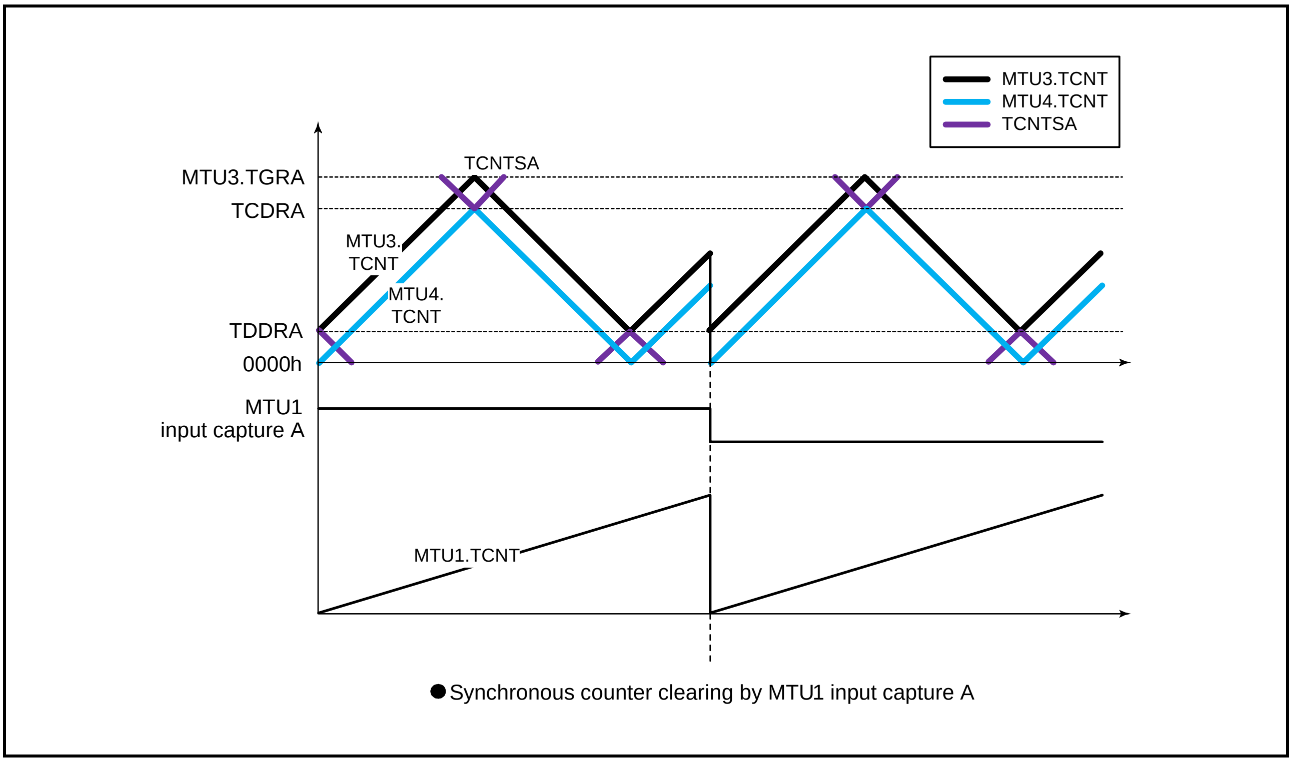


Figure 20.62 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 20.63. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 20.63) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4. In MTU3 and MTU4, synchronous clearing in any of MTU0 to MTU2 can cause counter clearing.

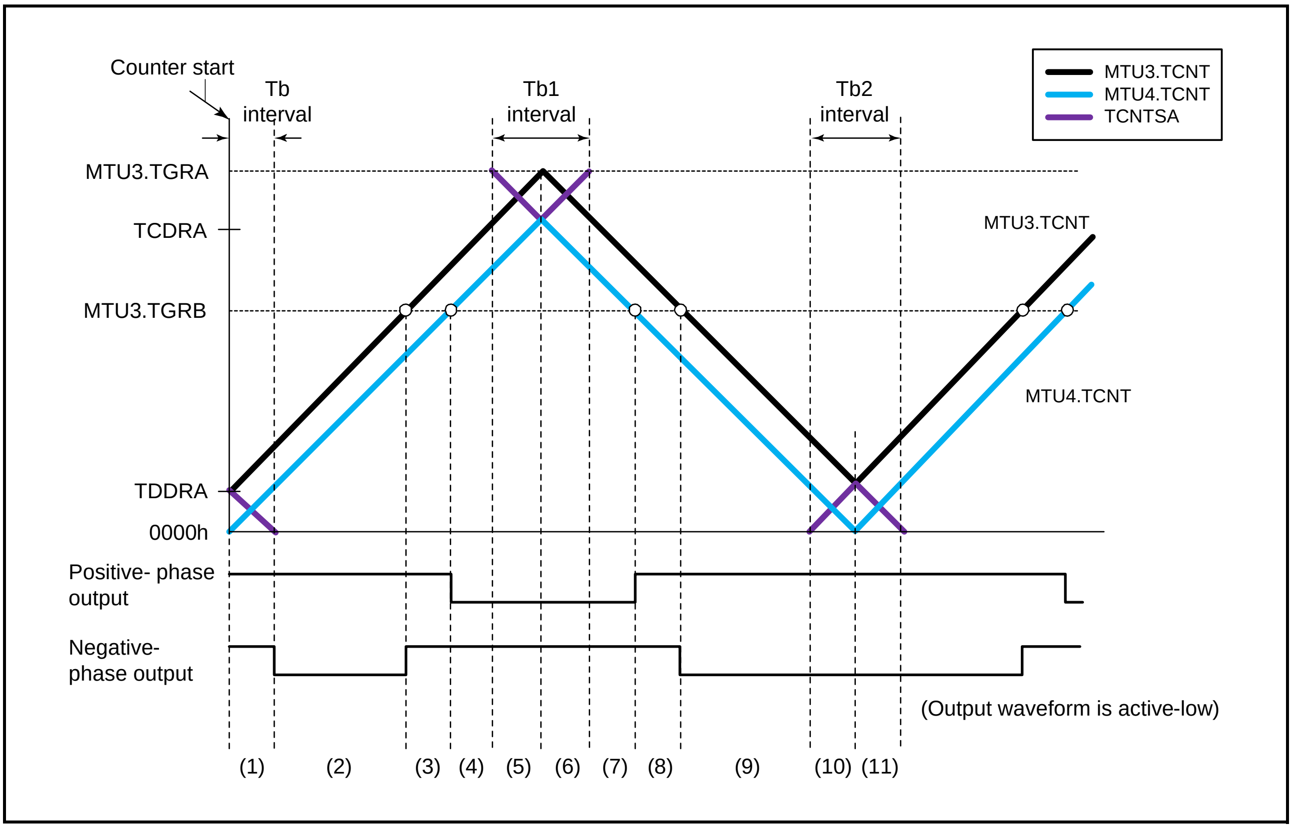


Figure 20.63 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode.

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 20.64.

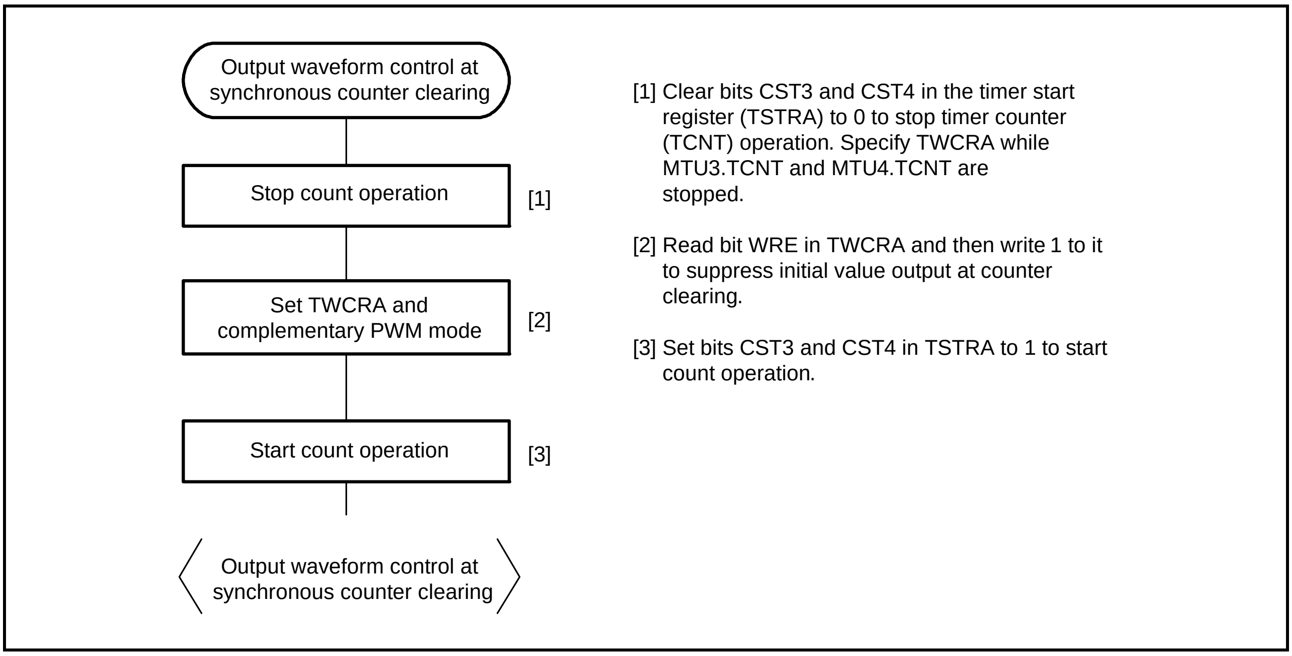


Figure 20.64 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 20.65 to Figure 20.68 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 20.65 to Figure 20.68, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 20.63, respectively.

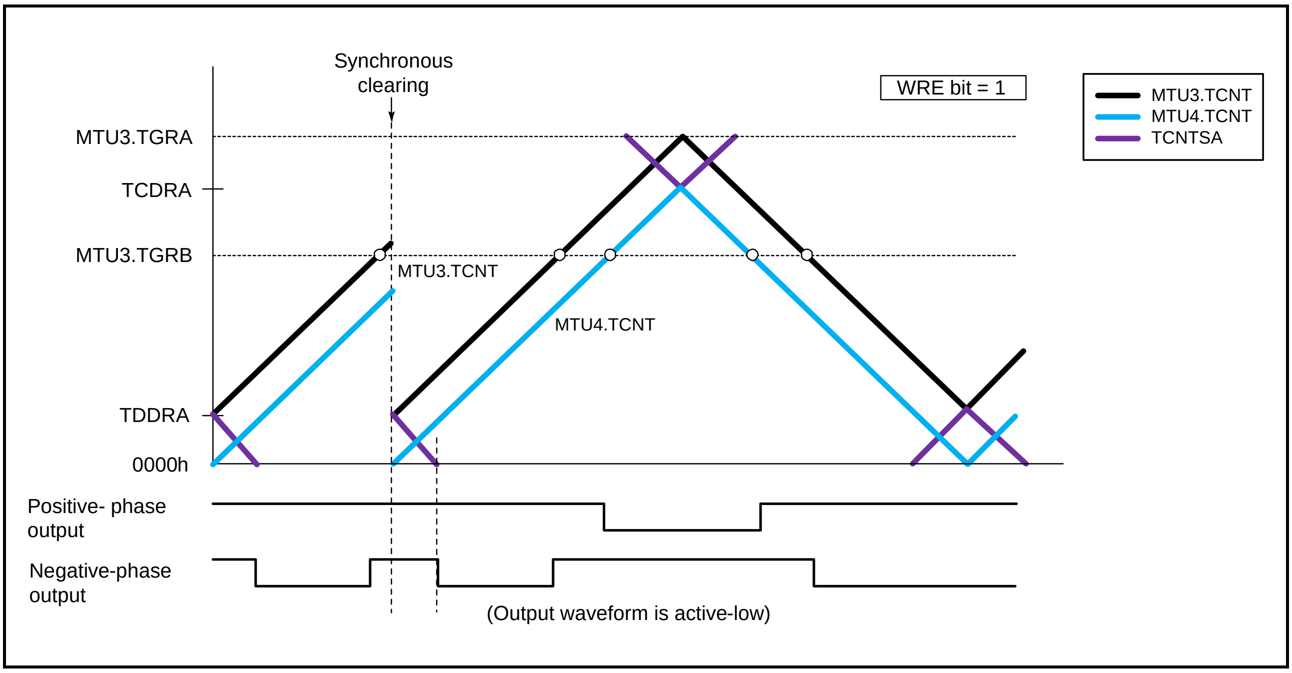


Figure 20.65 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 20.63; TWCRA.WRE Bit is 1)

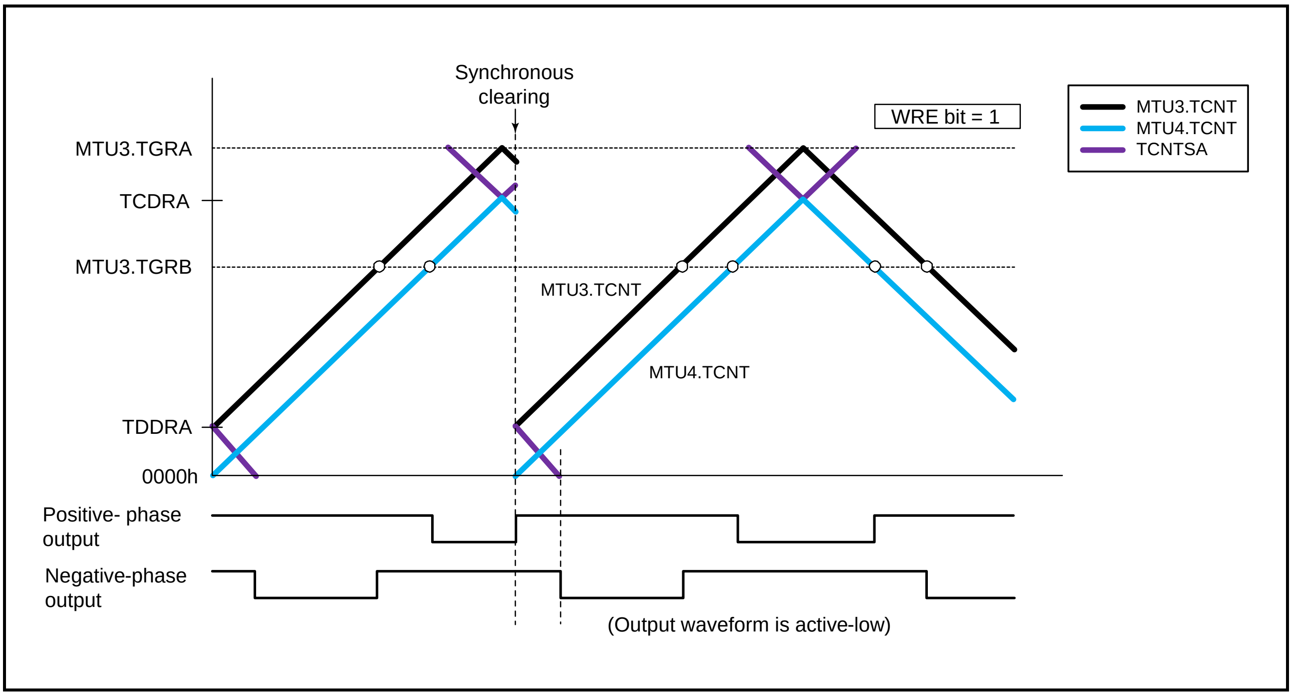


Figure 20.66 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 20.63; TWCRA.WRE Bit is 1)

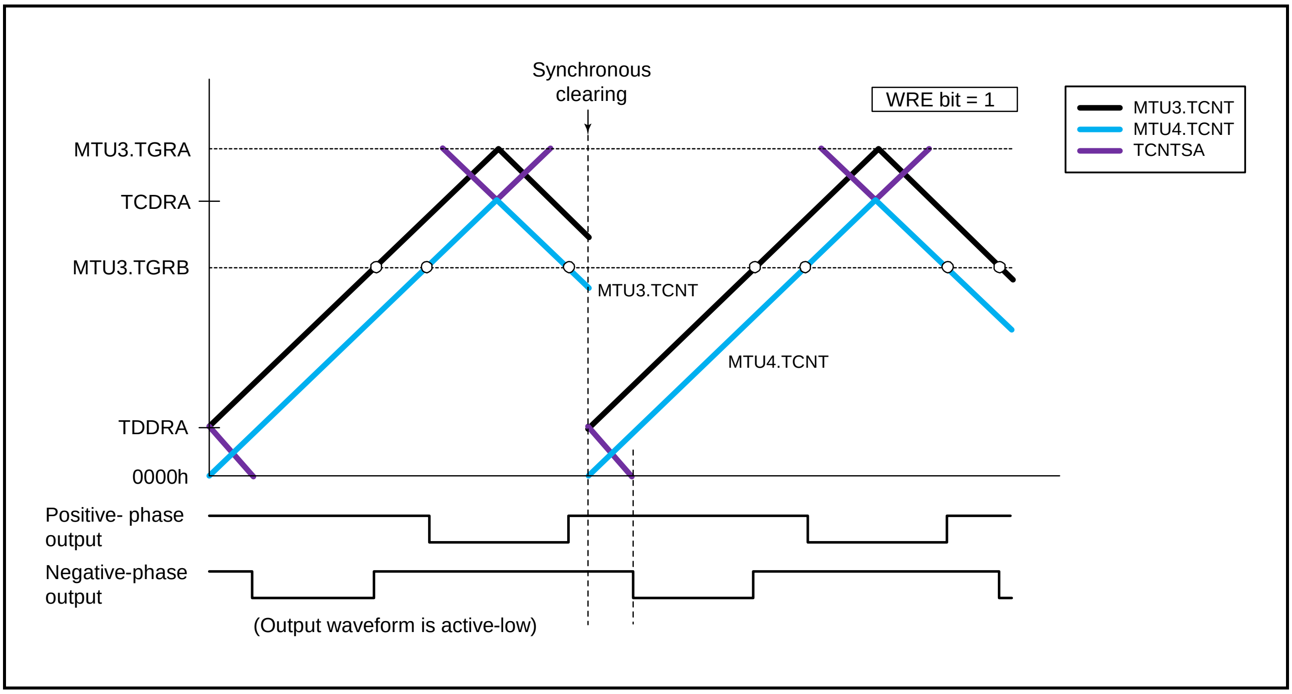


Figure 20.67 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 20.63; TWCRA.WRE Bit is 1)

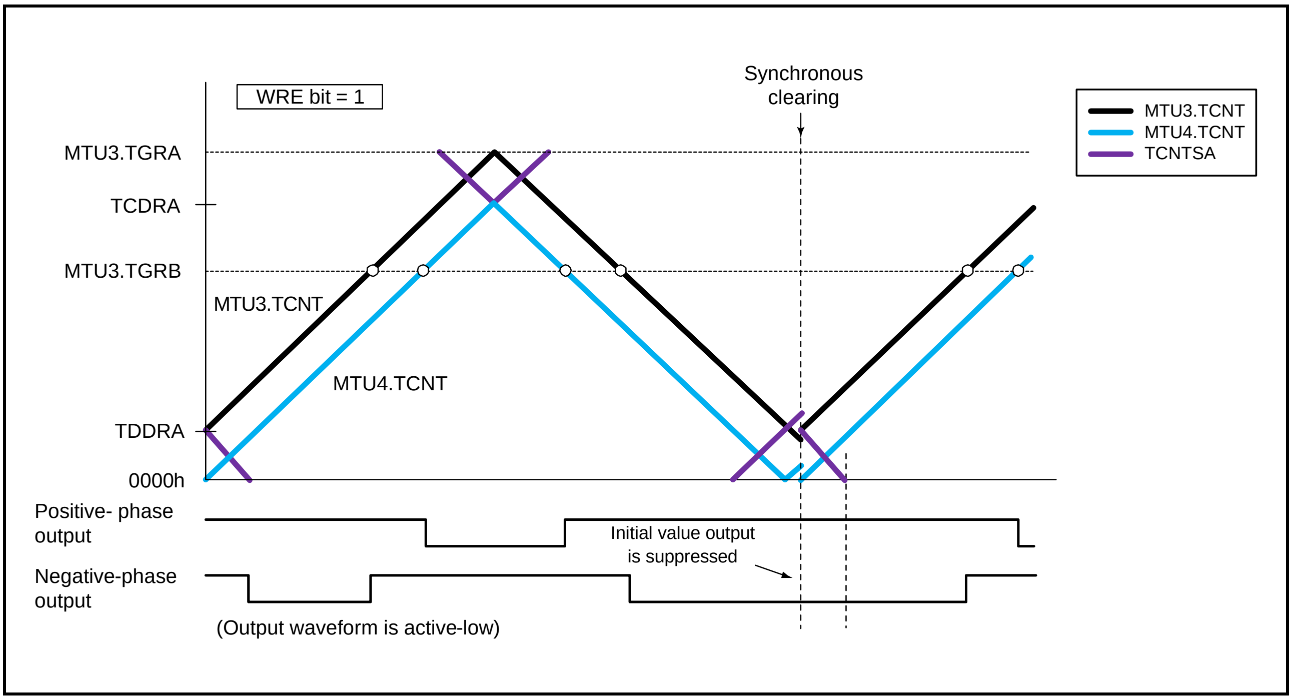


Figure 20.68 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 20.63; TWCRA.WRE Bit is 1)

(o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA can be cleared by MTU3.TGRA compare match when the TWCRA.CCE bit.

Figure 20.69 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
Note 2. Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4 bits in the timer synchronous register (TSYRA)).
Note 3. Do not set the PWM duty value to 0000h.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1A) to 1.

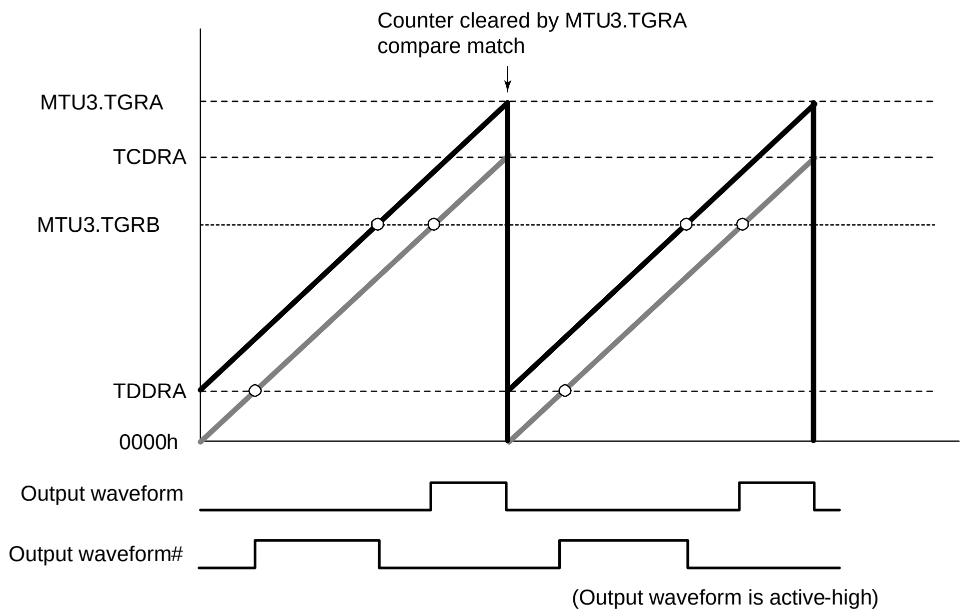
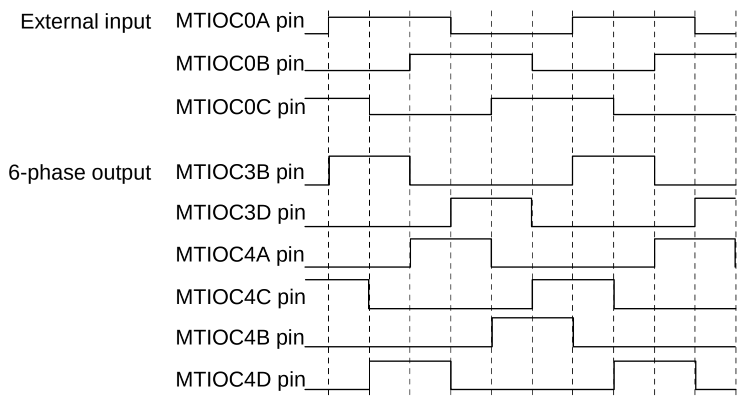


Figure 20.69 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

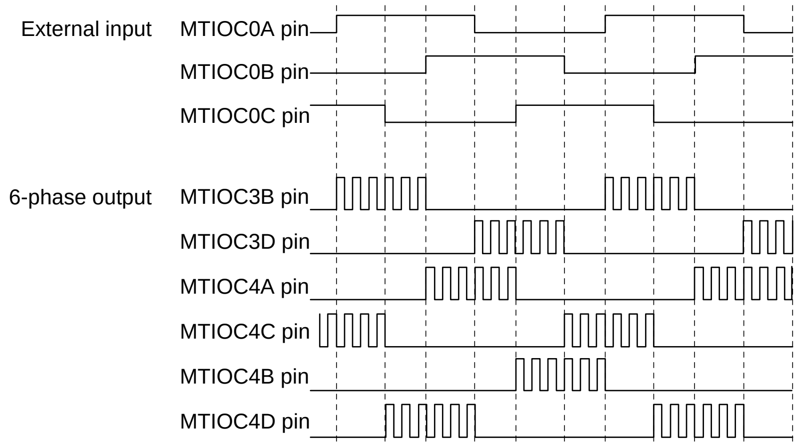
(p) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode when MTU3 and MTU4 are used, a brushless DC motor can easily be controlled using the TGCRA register. Figure 20.70 to Figure 20.73 show examples of brushless DC motor driving waveforms created using TGCRA.
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 TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (make appropriate settings with the MPC and port mode registers (PMR) of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.
When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is cleared to 0 or set to 1.
The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.
The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.



■ When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 0, and output active level = high

Figure 20.70 Example of Output Phase Switching by External Input (1)



■ When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 0, and output active level = high

Figure 20.71 Example of Output Phase Switching by External Input (2)

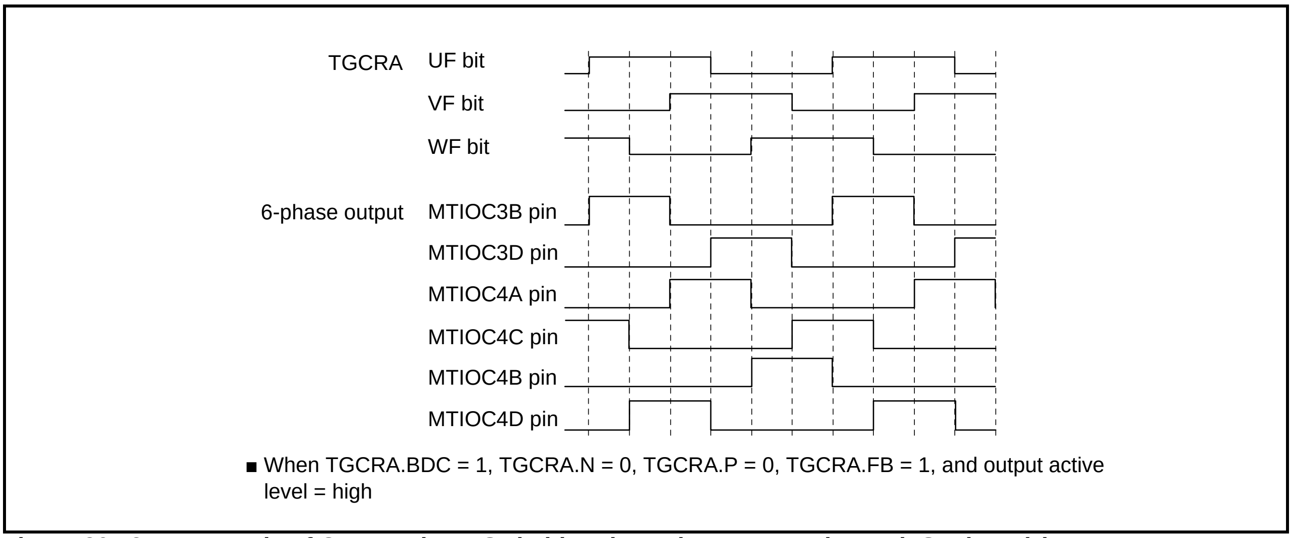


Figure 20.72 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

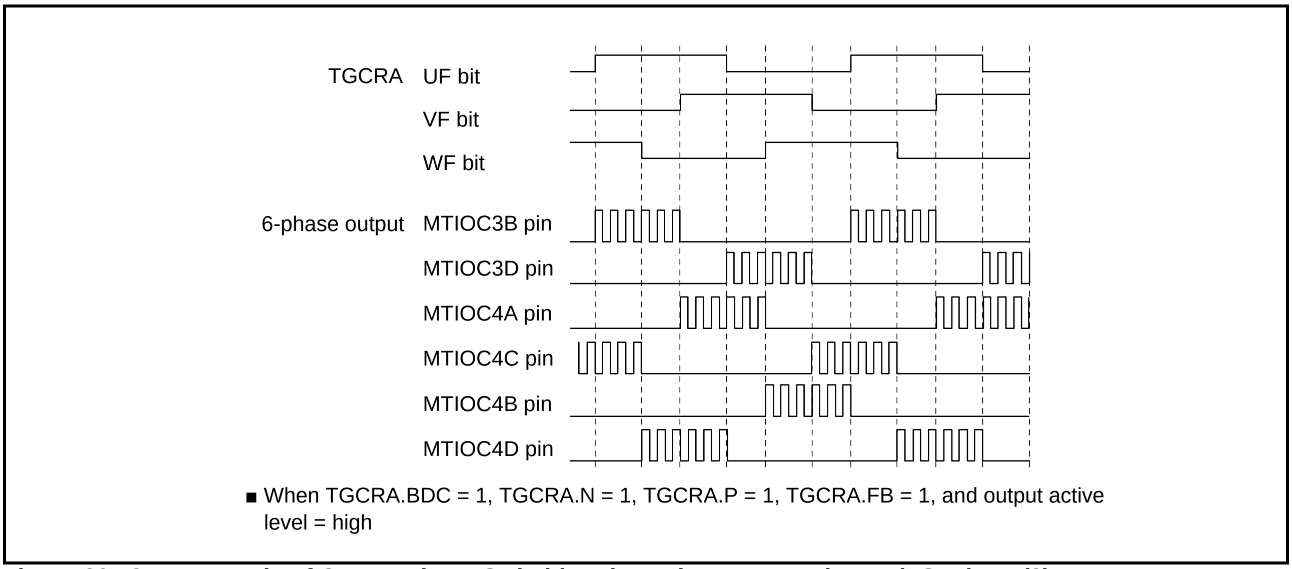


Figure 20.73 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA compare match, MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.
When start requests using MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.
A/D converter start requests can be specified by setting the TIER.TTGE bit. To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2bit to 1.

(r) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS bit to 1.
When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF) at the same time. For details of the setting procedure, refer to section 20.3.8 (1), Example of Complementary PWM Mode Setting Procedure

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is not set to the buffer register A value, asymmetric PWM waveforms are output.

Figure 20.74 shows an example of double buffer operation.
Each register data is transferred as follows.

- After MTU4.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD (buffer A) to Temp3A (temporary A) and from MTU4.TGRF (buffer B) to Temp3B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A (temporary A) to MTU4.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B (temporary B) to MTU4.TGRB (compare).

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

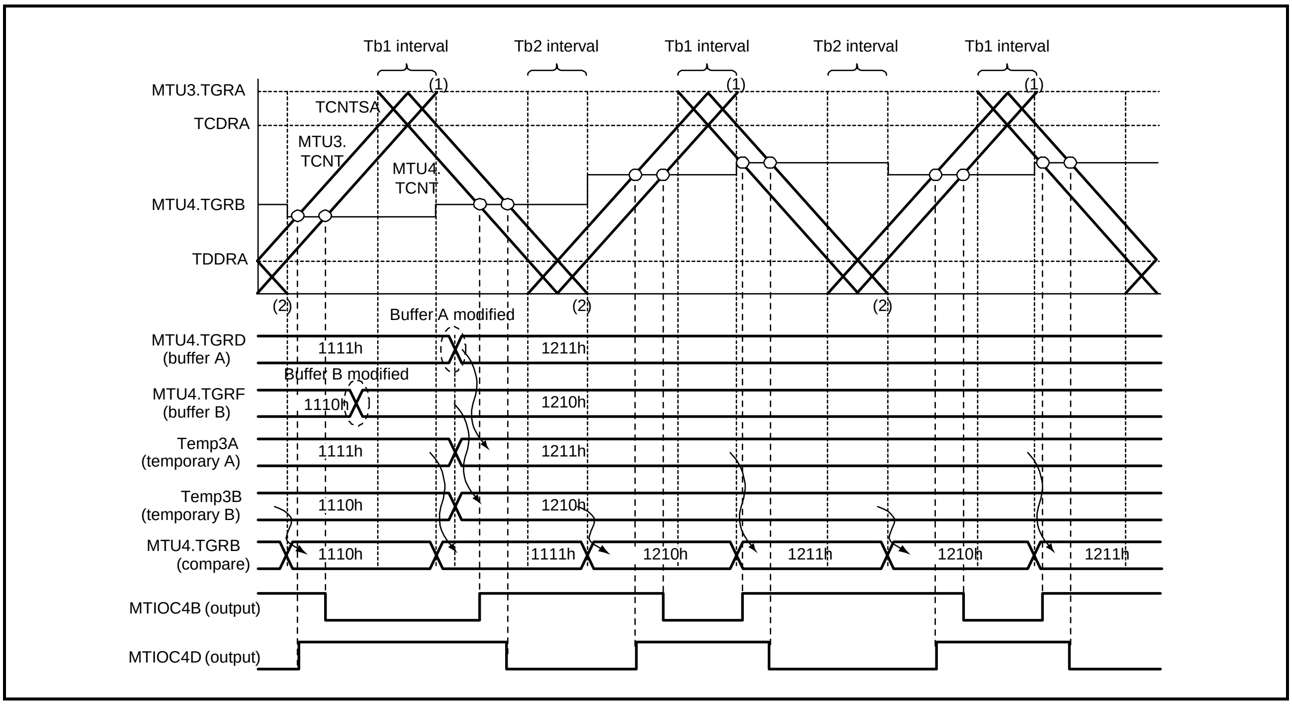


Figure 20.74 Example of Double Buffer Operation

Figure 20.75 shows an example when the buffer write value is smaller than the TDDRA value, and Figure 20.76 shows an example when the write value is greater than TCDRA.
In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

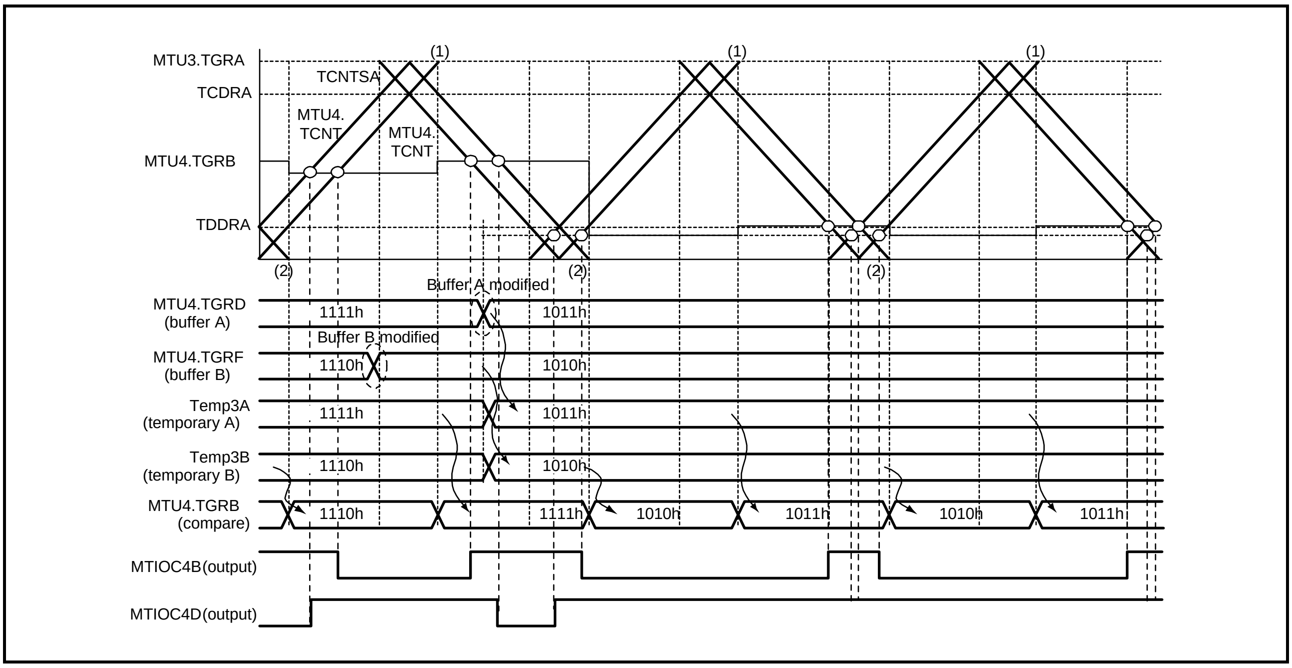


Figure 20.75 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

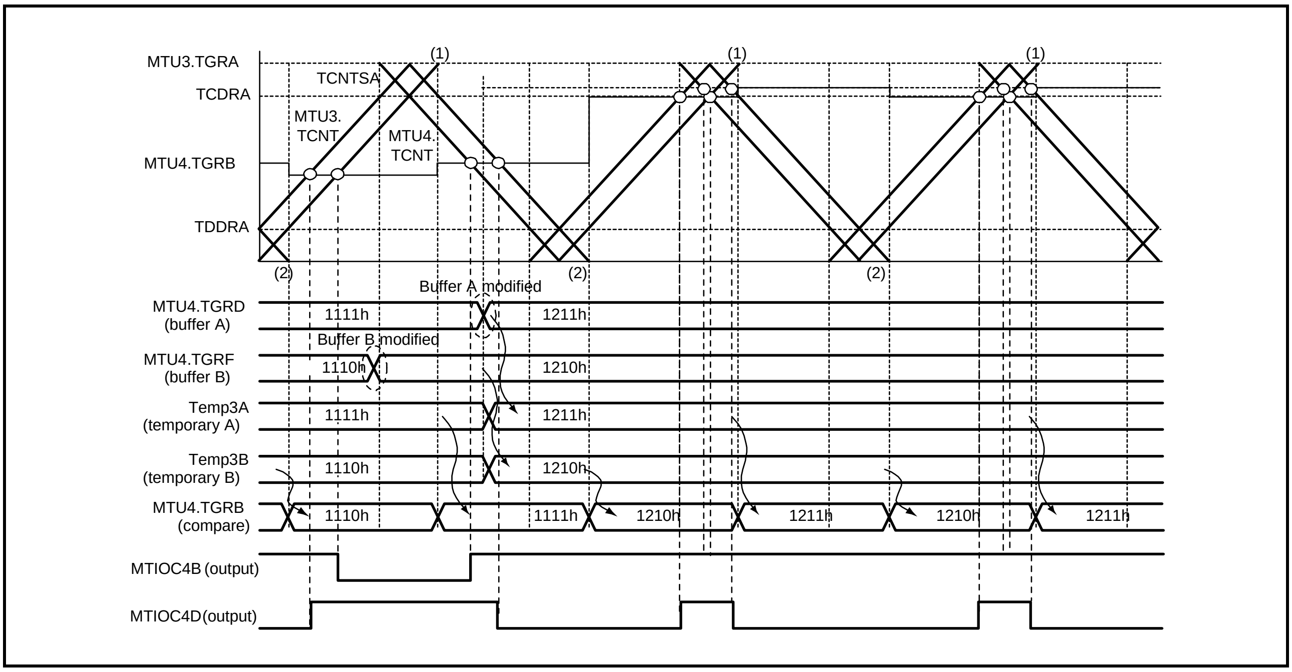


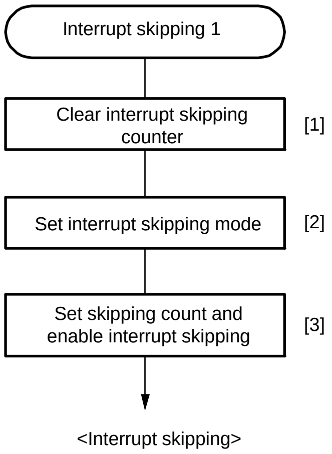
Figure 20.76 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(3) Interrupt Skipping Function 1 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 TITCR1A register.
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA register. 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 MTU4.TADCR register. For the linkage with the A/D converter start request delaying function, refer to section 20.3.9, A/D Converter Start Request Delaying Function.
The TITCR1A register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA) to 0, TGIA3 interrupt requests are disabled by setting the MTU3.TIER register, TCIV4 interrupt requests are disabled by setting the MTU4.TIER register, and a compare match is not generated. Before changing the skipping count, be sure to clear the T3AEN and T4VENbits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 20.77 shows an example of the interrupt skipping function 1 setting procedure. Figure 20.78 shows the periods during which interrupt skipping count can be changed.



[1] Set bits T3AEN and T4VEN in the timer interrupt skipping set register 1 (TITCR1A) to 0 to clear the skipping counter.

[2] Set bit TITM in the timer interrupt skipping mode register (TITMRA) to 0 to select interrupt skipping function 1.

[3] Specify the interrupt skipping count within the range from 0 to 7 times in bits T3ACOR2 to T3ACOR0 and T4VCOR2 to T4VCOR0 in TITCR1A, and enable interrupt skipping through bits T3AEN and T4VEN.

Note: The setting of TITCR1A must be done while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which a compare match is not generated. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

Figure 20.77 Example of Interrupt Skipping Function 1 Setting Procedure

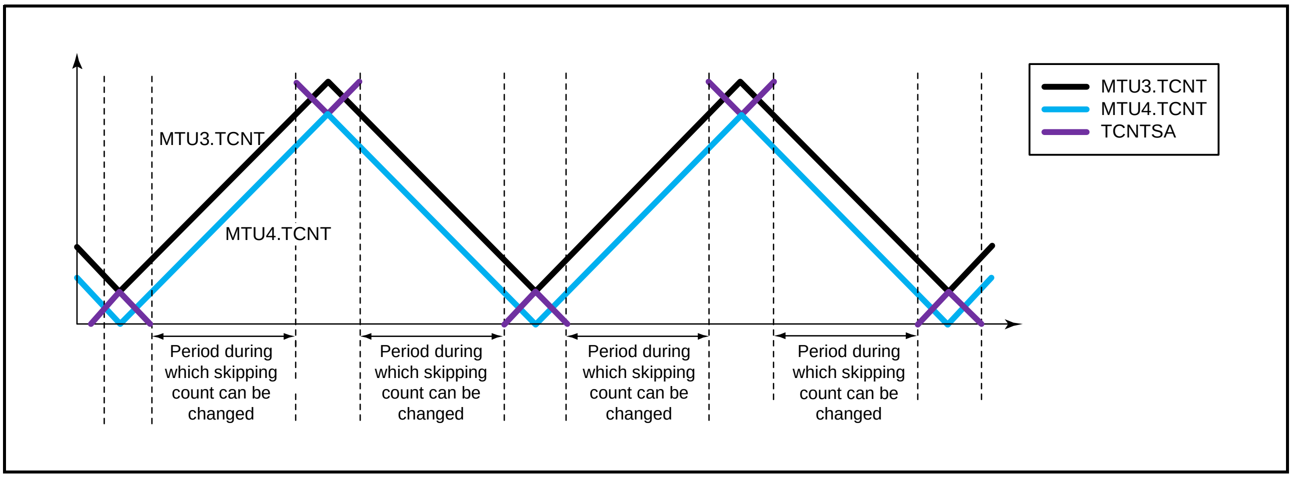


Figure 20.78 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 20.79 shows an example of TGIA3 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 TITCR1A register.

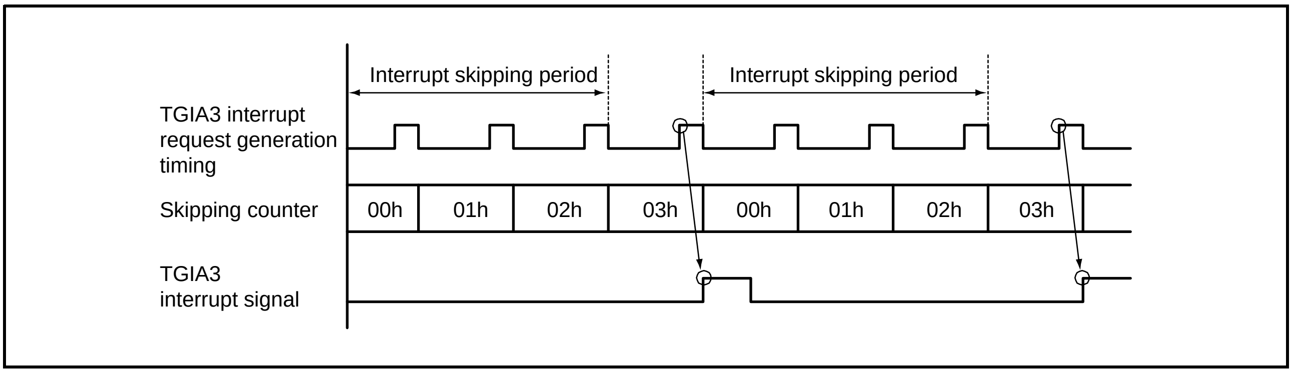


Figure 20.79 Example of Interrupt Skipping Function 1

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA register.

Figure 20.80 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 20.81 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Note that the buffer transfer-enabled period differs depending on whether only the T3AEN bit in the TITCR1A register is set to 1, only the T4VEN bit in the TITCR1A register is set to 1, or both the T3AEN and T4VEN bits are set to 1. Figure 20.82 shows the relationship between the T3AEN and T4VEN bit settings in TITCR1A and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register 1 (TITCR1A) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in TBTERA to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

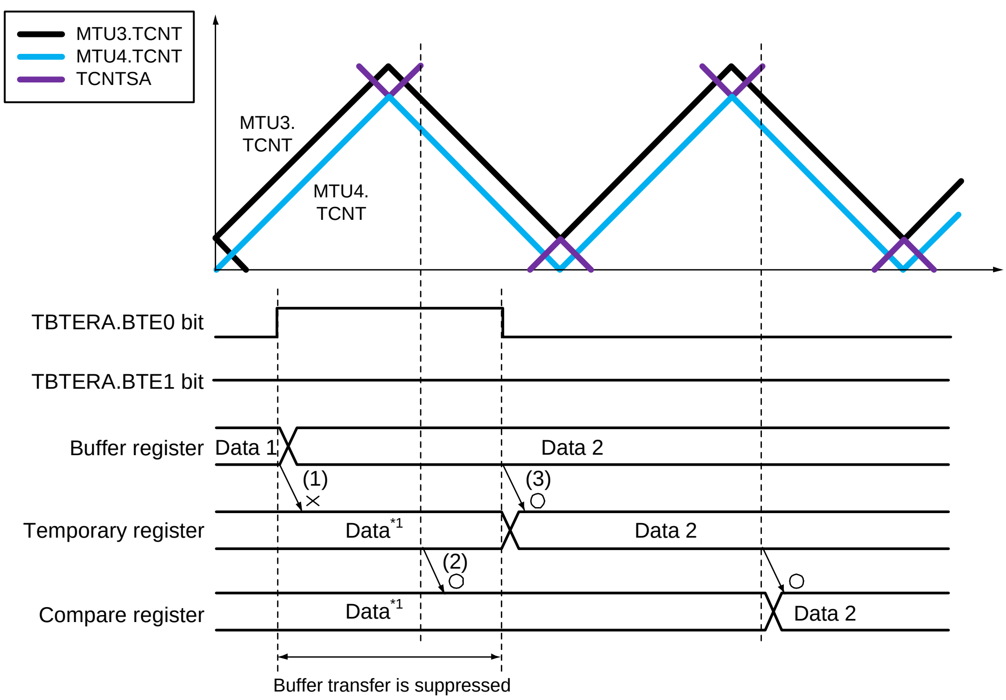


Figure 20.80 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

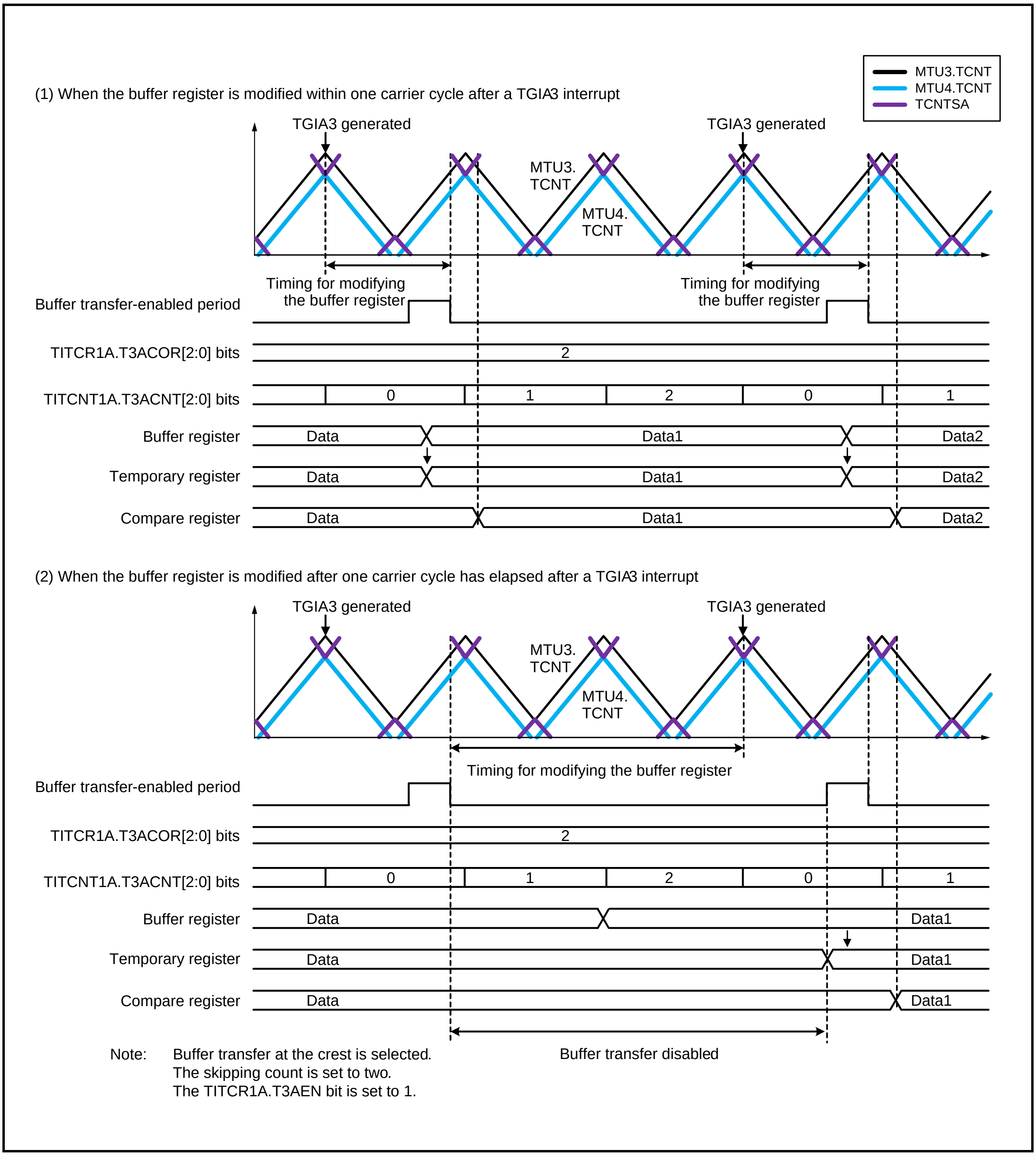


Figure 20.81 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

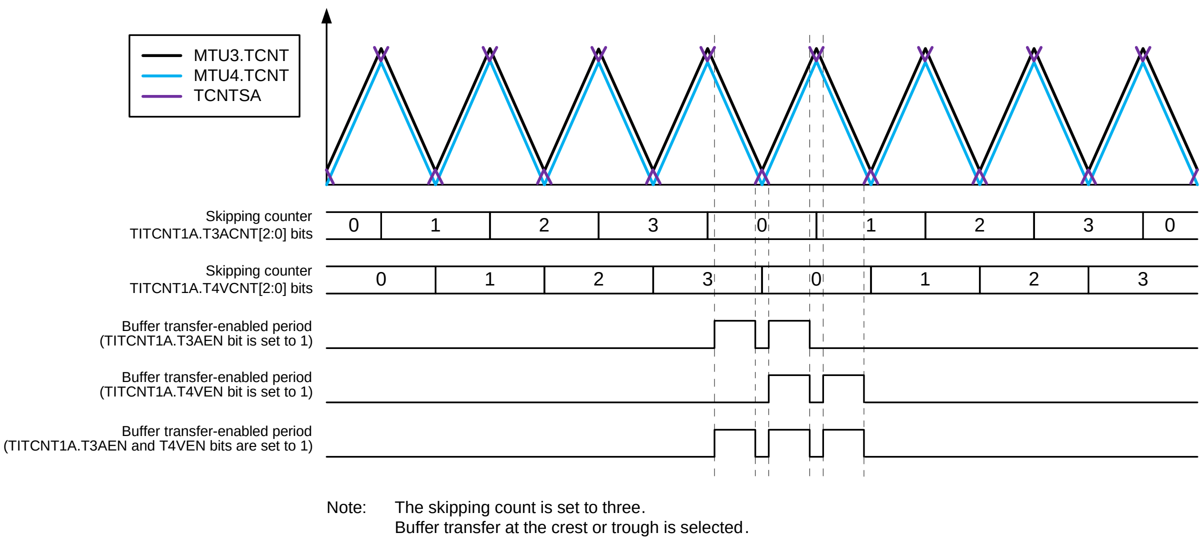


Figure 20.82 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA register. The applicable registers are some of the registers in MTU3 and MTU4 shown below:
24 registers in total
MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, and MTU.TDDRA
This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

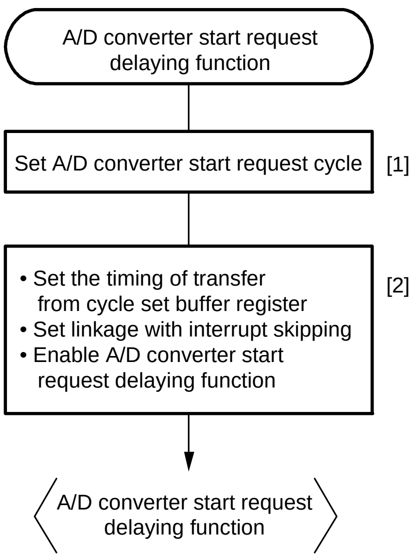
The PWM output pins of MTU0, MTU3, and MTU4 can be set to the high-impedance state automatically.
Refer to section 21, Port Output Enable 3 (POE3b), for details.

20.3.9 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 (MTU4.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 MTU4.TADCR.

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 20.83 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle buffer register (MTU4.TADCOBRA or MTU4.TADCOBRB) and timer A/D converter start request cycle register (MTU4.TADCORA or MTU4.TADCORB).
(When the timing of transfer from the cycle buffer register to cycle register is set to the trough, or both trough and crest in step [2] below, the same initial value must be specified in the cycle buffer register and cycle register because transfer is performed immediately after counting starts.)

[2] Use the BF[1:0] bits in the timer A/D converter start request control register (MTU4.TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
• Specify whether to link with interrupt skipping through the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits.
• Use the UT4AE, DT4AE, UT4BE, and DT4BE bits to enable A/D converter start requests (TRG4AN or TRG4BN).

Note: Perform MTU4.TADCR setting while MTU4.TCNT is stopped.
Note: Set MTU4.TADCR.BF1 to 0 when complementary PWM mode is not selected.
Note: Set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, and DT4BE to 0 when complementary PWM mode is not selected.
Note: Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits to 0 when interrupt skipping is disabled.

Figure 20.83 Example of Procedure for Specifying A/D Converter Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 20.84 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.

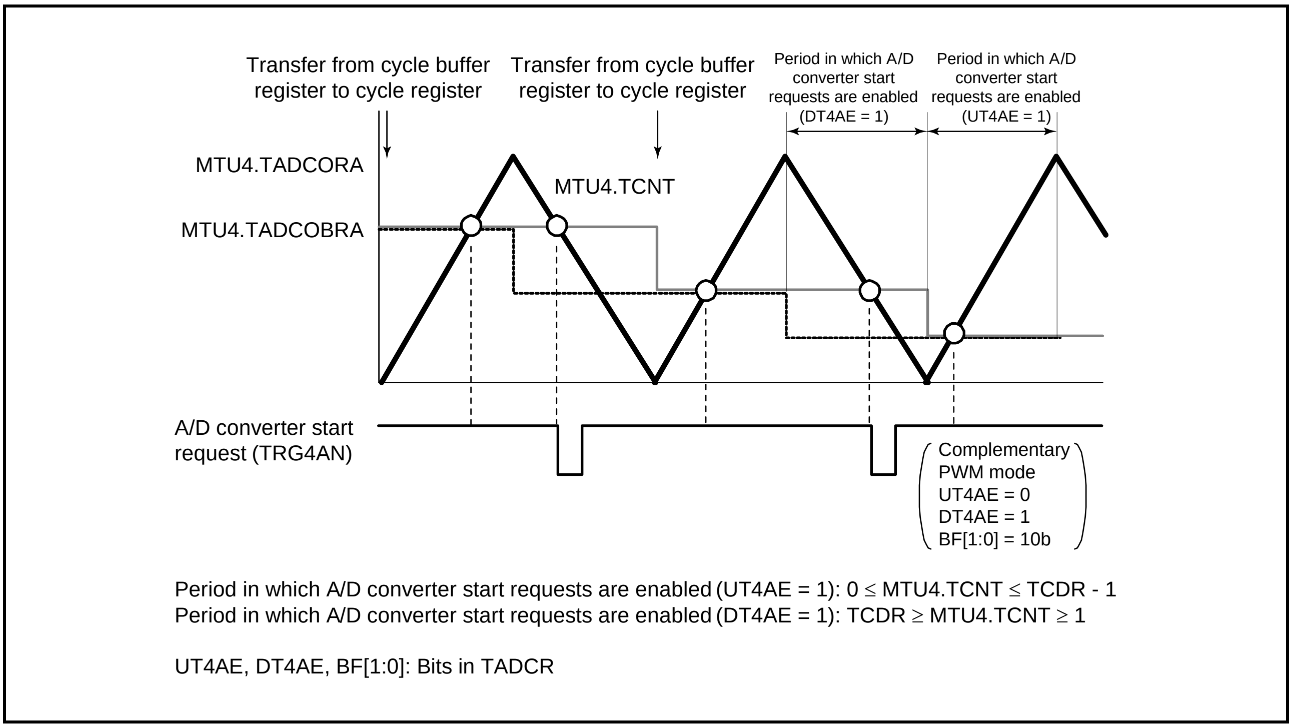


Figure 20.84 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Converter Start Requests are Enabled

When the MTU4.TCNT counter and the MTU4.TADCORA or MTU4.TADCORB register matches within the period enabled by the UT4AE and UT4BE bits, the corresponding A/D converter start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1 in complementary PWM mode, A/D converter start requests are enabled during the MTU4.TCNT up-counting (0 ≤ MTU4.TCNT ≤ TCDR - 1). When the DT4AE and DT4BE bits in the MTU4.TADCR register are set to 1, A/D converter start requests are enabled during MTU4.TCNT down-counting (TCDR ≥ MTU4.TCNT ≥ 1). Refer to Figure 20.84.

(4) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR register.
In complementary PWM mode, data is also transferred from the timer A/D converter start request cycle set buffer registers to the timer A/D converter start request cycle set registers when MTU4.TGRD register is updated.
There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode.
For details, section 20.6.27, Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode.
In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR register to 0.

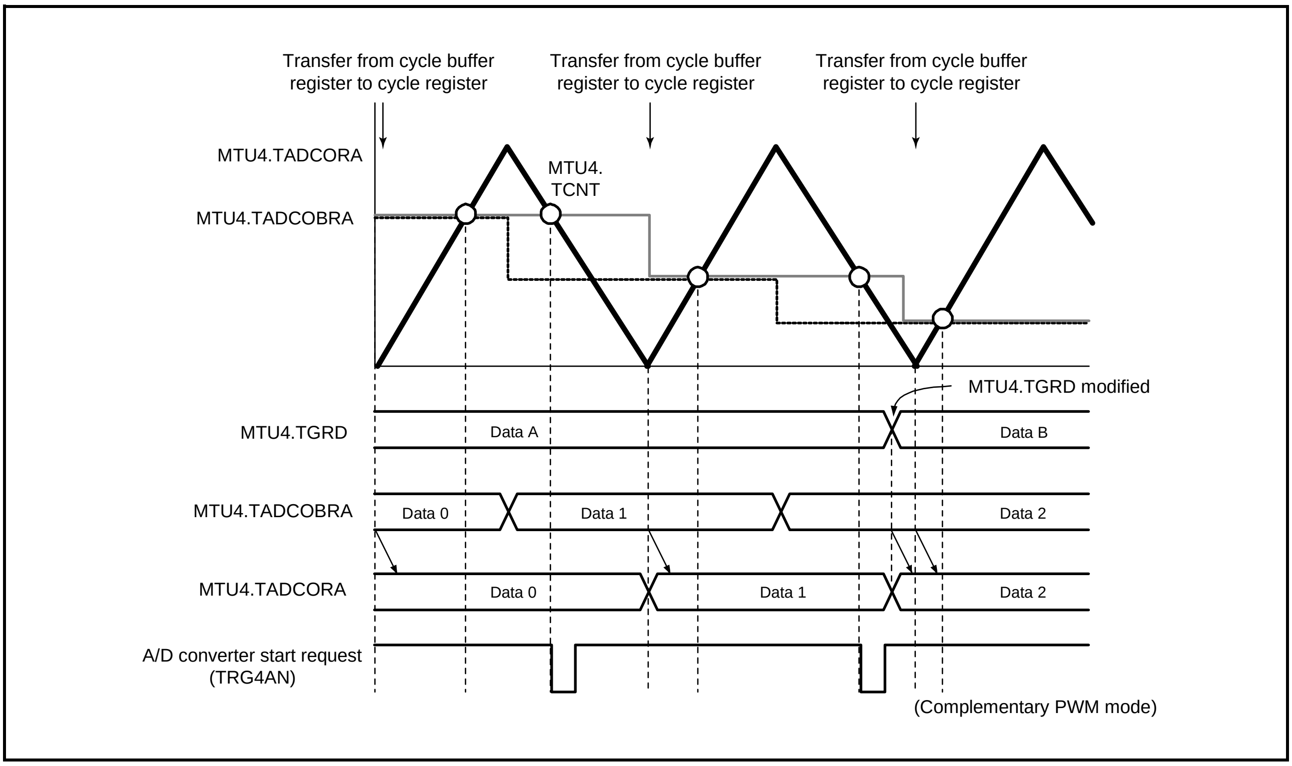


Figure 20.85 Example of A/D Converter Start Request Signal (TRG4AN) and Buffer Transfer Operation

(5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register.

Figure 20.86 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 1.

Figure 20.87 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 1.

In modes other than complementary PWM mode, do not use the A/D converter start request delaying function linked with the interrupt skipping function 1.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register to 0.

Note: This function should be used in combination with interrupt skipping 1.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR1A) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping 1 (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (MTU4.TADCR) to 0).
When this function is used, MTU4.TADCORA and MTU4.TADCORB should be set with the value ranging 0002h to the value set in TCDRA minus 2.

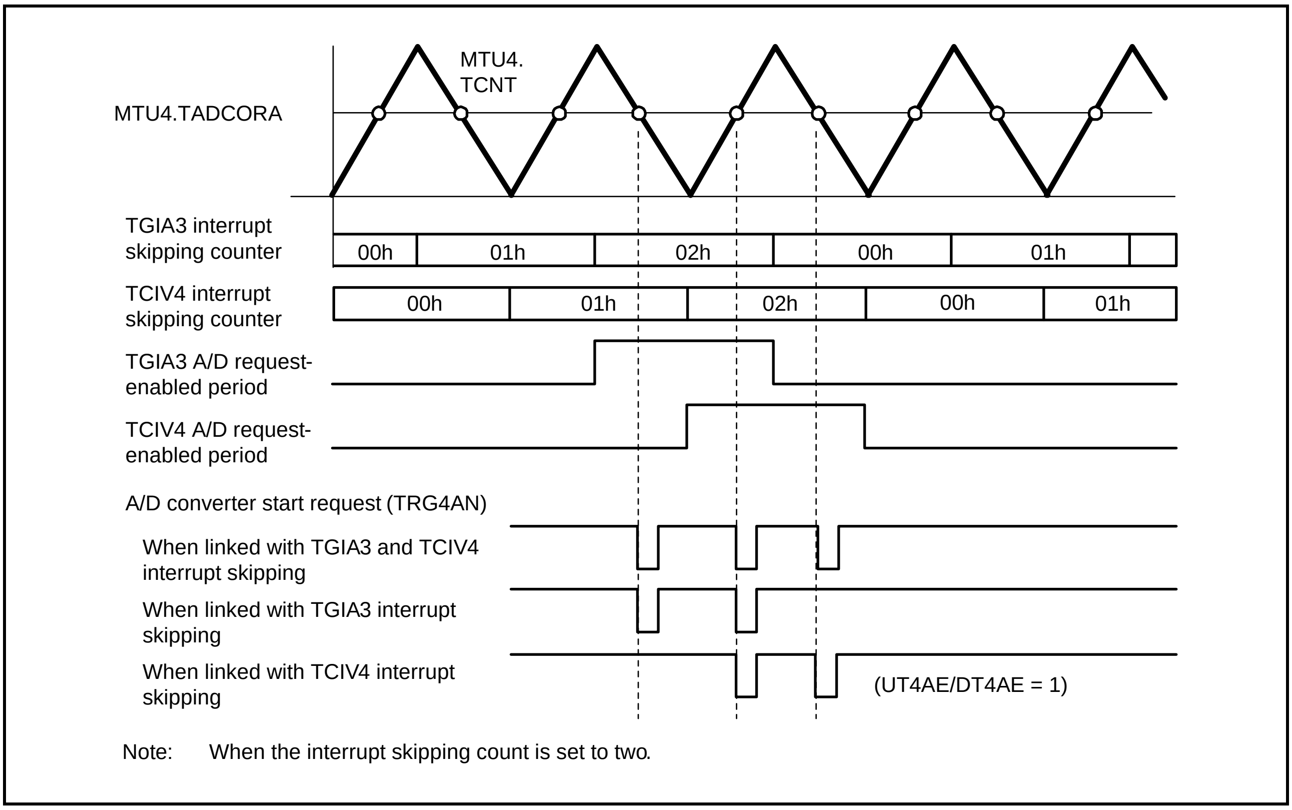


Figure 20.86 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

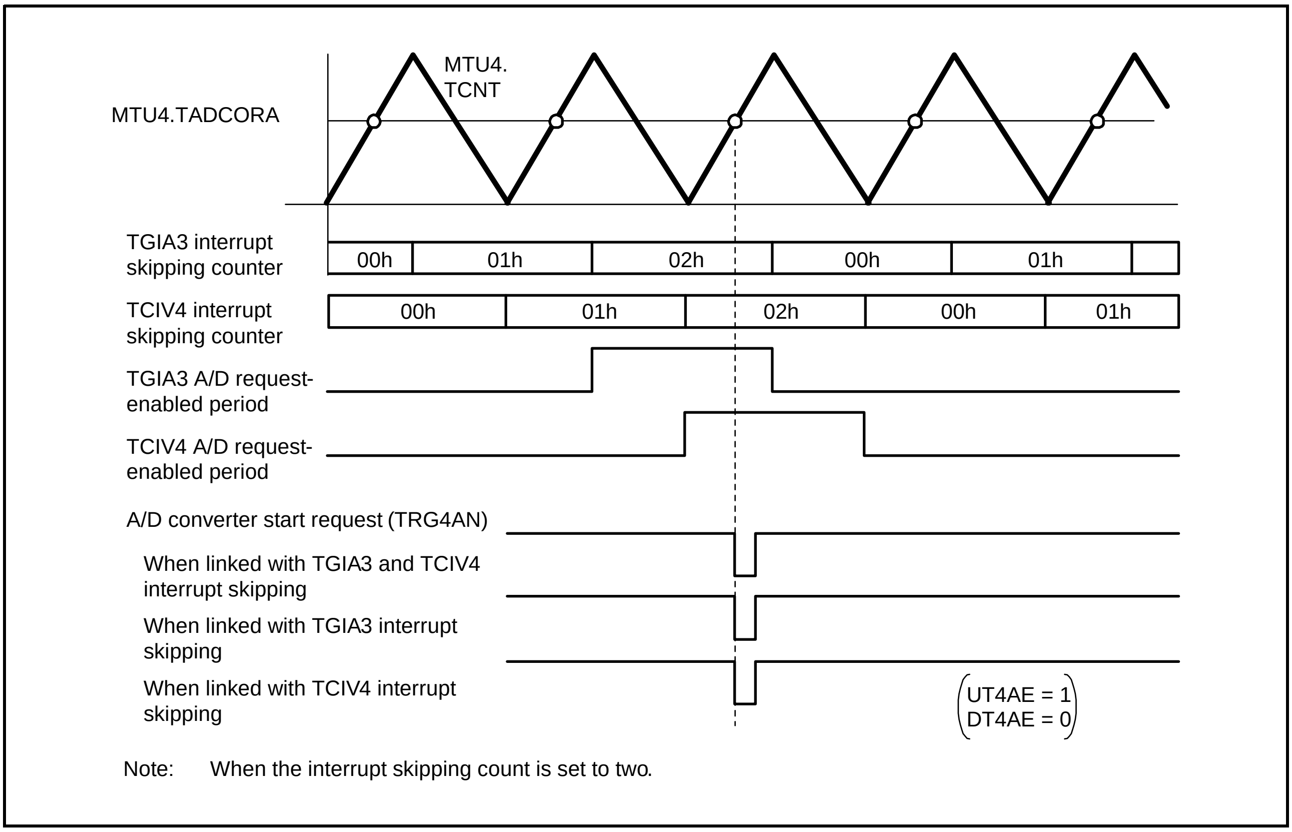


Figure 20.87 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(6) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] bits in TITCR2A register every time an A/D converter start trigger (TRG4AN or TRG4BN) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid and an A/D converter start request signal (TRG4ABN) is output.
This function is valid only when the A/D converter request delaying function is enabled.

(a) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 20.88 shows an example of procedure for setting interrupt skipping function 2.

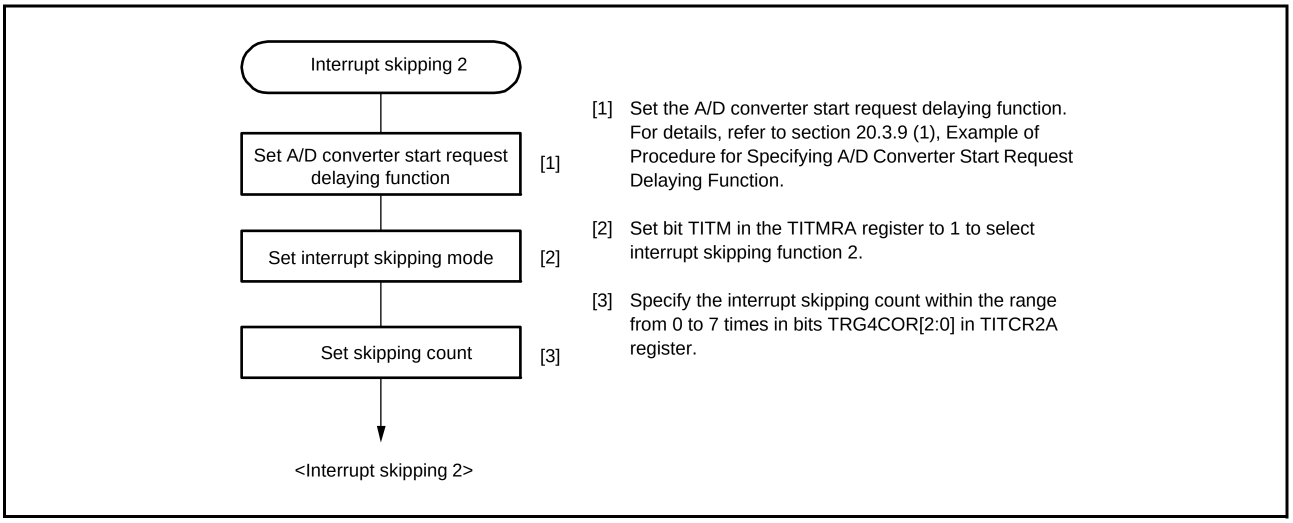


Figure 20.88 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 20.89 shows an example of interrupt skipping function 2 operation.

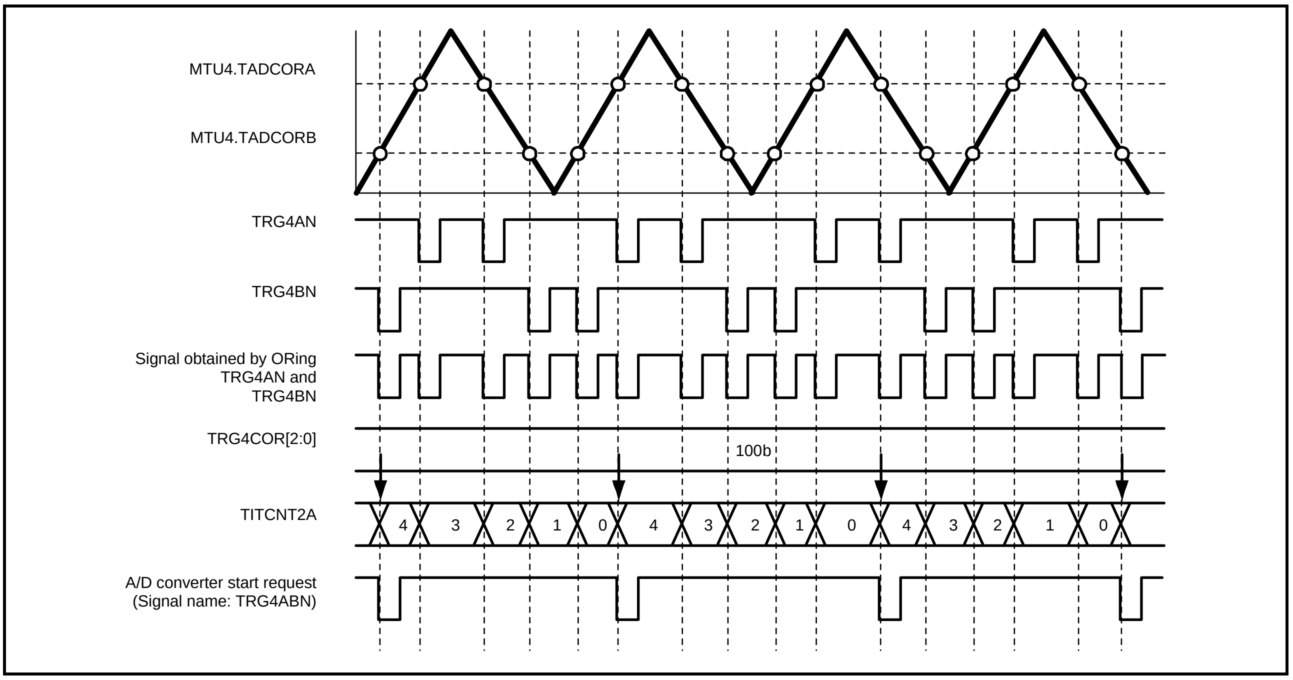


Figure 20.89 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

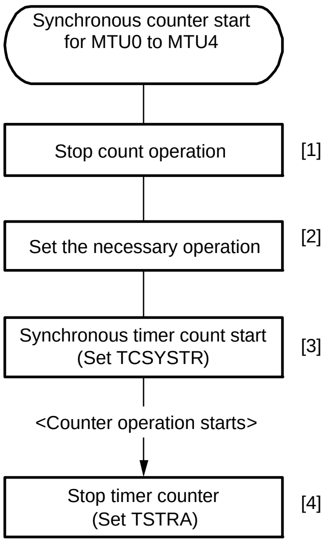
20.3.10 Synchronous Operation of MTU0 to MTU4

(1) Synchronous Counter Start for MTU0 to MTU4

The counters in MTU0 to MTU4 can be started synchronously by making the TCSYSTR settings.

(a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4

Figure 20.90 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4.



[1] Set TSTRA to stop the counters used for synchronous start operation.

[2] Specify the necessary operation with appropriate registers such as TCR, TMDR1, and TMDR2.

[3] In TCSYSTR, set the bits corresponding to the counters to be started synchronously to 1 at the same time. When TCSYSTR is set, TSTRA for the target timer counters are automatically set appropriately, the counters start synchronously, and TCSYSTR is automatically cleared.

[4] To stop a timer counter, set the bit in TSTRA that corresponds to the counter to 0.

Note: Even if a bit in TCSYSTR corresponding to an operating counter is cleared to 0, the counter will not stop.

Note: To start counters synchronously in reset-synchronized PWM mode or complementary PWM mode using MTU3 and MTU4, set the bits in TCSYSTR that correspond to MTU3 and MTU4 to 1 at the same time.

Figure 20.90 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4

(b) Examples of Synchronous Counter Start Operation

Figure 20.91 shows an examples of synchronous counter start operation for MTU0 to MTU4.

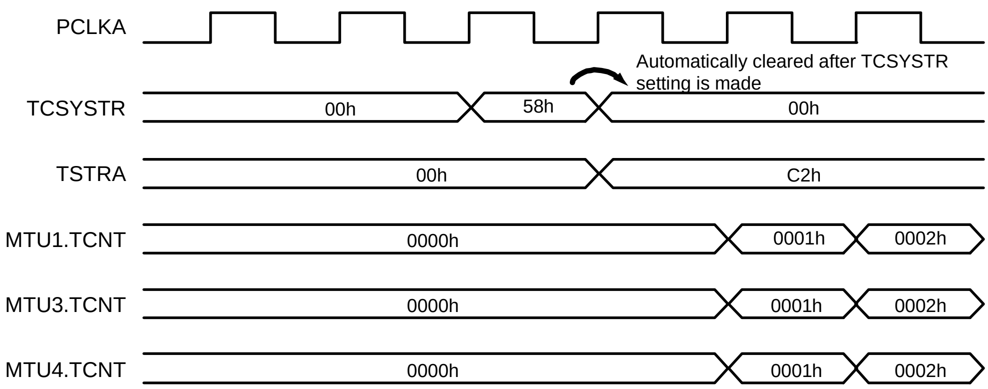


Figure 20.91 Examples of Synchronous Counter Start Operation for MTU0 to MTU4

20.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.

When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.

Figure 20.92 shows an example of setting external pulse width measurement, and Figure 20.93 an example of external pulse width measurement.

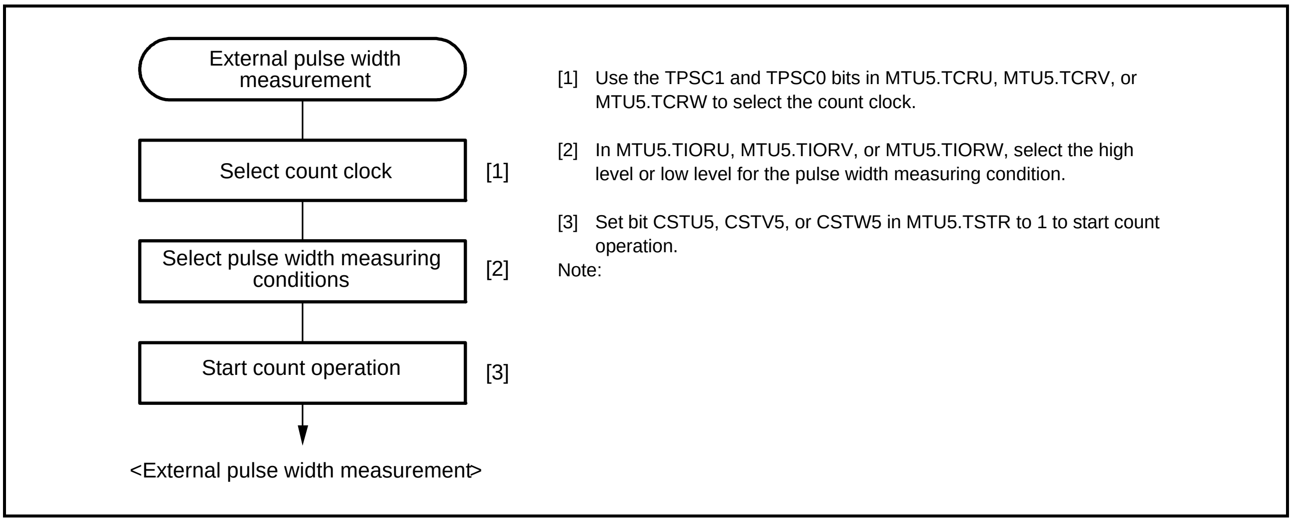


Figure 20.92 Example of External Pulse Width Measurement Setting Procedure

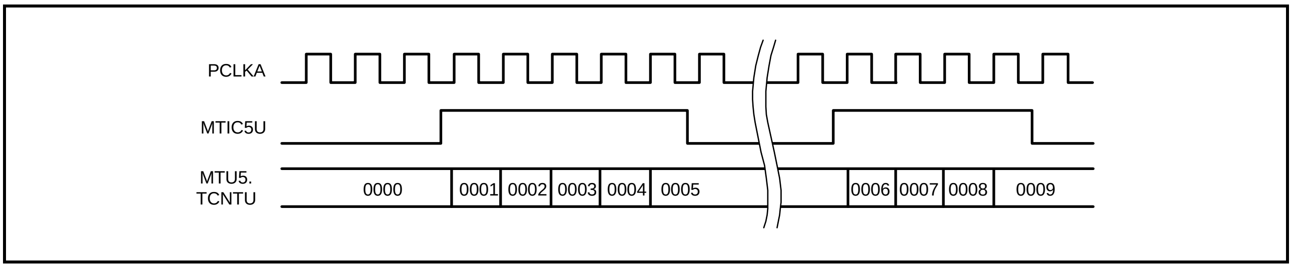


Figure 20.93 Example of External Pulse Width Measurement (Measuring High Pulse Width)

20.3.12 Dead Time Compensation

Figure 20.94 shows an example of the motor control circuit used to feed back a delay in the dead time (delay between complementary PWM output and inverter output) to MTU5. The MTU5 external pulse measurement function allows the delay between the complementary PWM output and inverter output to be measured and reflected in the duty ratio, which can be used as dead time compensation for the PWM output waveform in complementary PWM operation when MTU3 and MTU4 are used (Figure 20.95). Figure 20.96 shows the procedure for setting dead time compensation using MTU5. For details on MTU5 operation at this time, refer to section 20.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

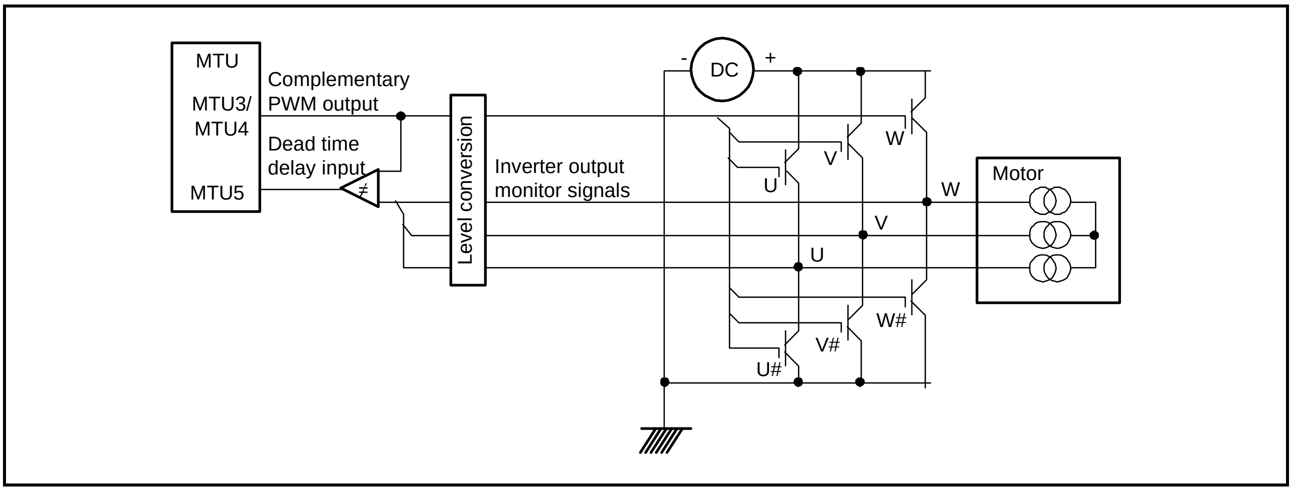


Figure 20.94 Motor Control Circuit Example

(1) Example of Dead Time Compensation Setting Procedure

Figure 20.96 shows an example of dead time compensation setting procedure by using three counters in MTU5.

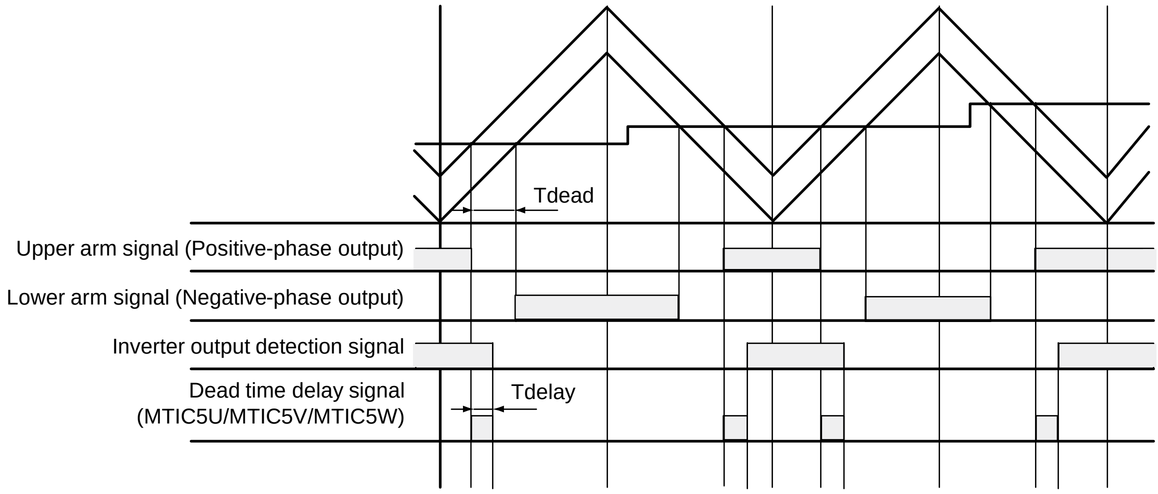
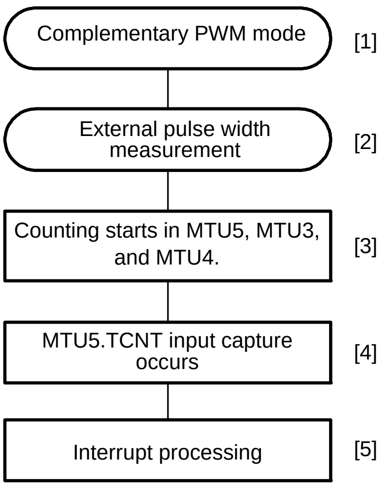


Figure 20.95 Delay in Dead Time in Complementary PWM Operation



[1] Set MTU3 and MTU4 to complementary PWM mode. For details, refer to section 20.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 20.3.11, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in TSTRA and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA3) or trough (TCIV4) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU3.TGRB and MTU5.TGRU, 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 TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note: The MPC and port mode registers (PMR) of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 20.96 Example of Dead Time Compensation Setting Procedure

20.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function can be used to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW are cleared to 0 at the transfer timing for TGRU, TGRV, and TGRW.
Figure 20.97 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest or trough in complementary PWM mode.

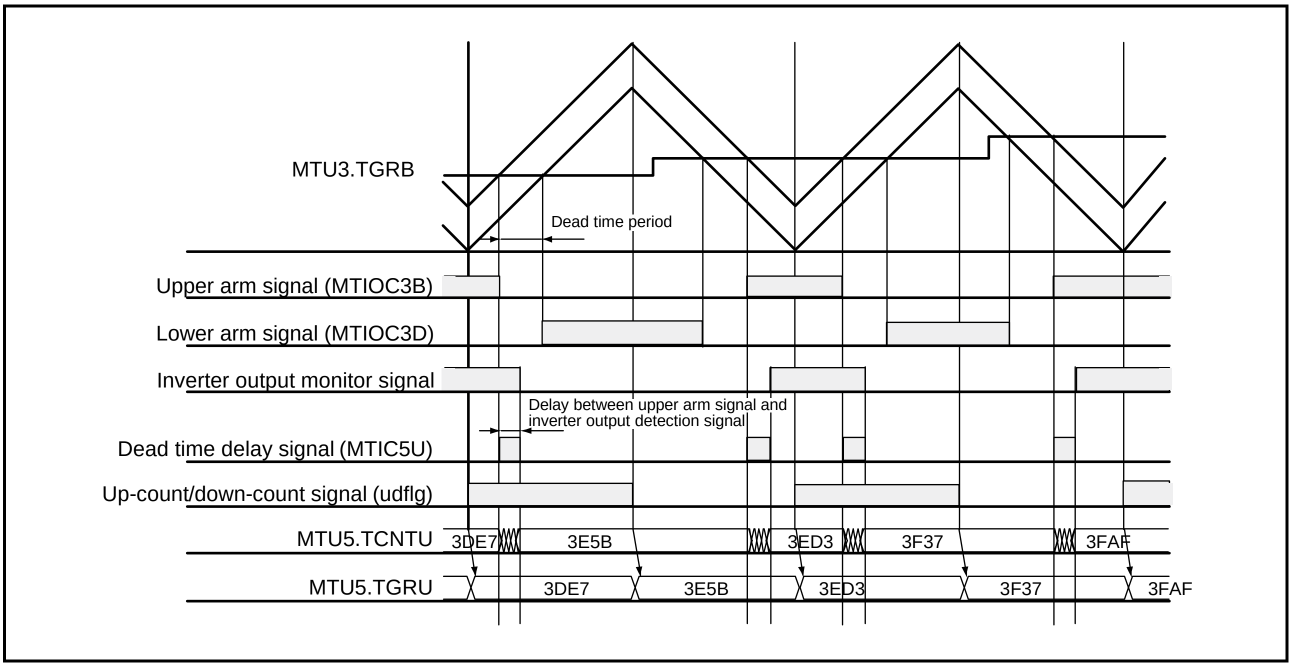


Figure 20.97 TCNTU Capture at Crest and/or Trough in Complementary PWM Operation

20.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.
Set the NFCRn register (n = 0 to 5, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.
Figure 20.98 shows the timing of noise filtering.

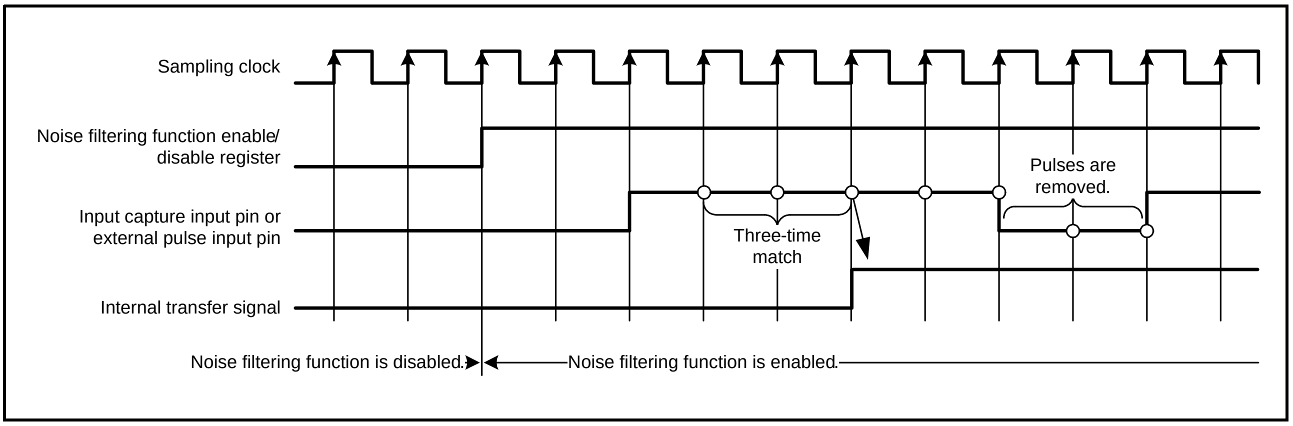


Figure 20.98 Timing of Noise Filtering

20.3.15 A/D Conversion Start Request Frame Synchronization Signal

This function can be used to monitor the generation timing of the A/D conversion start request signal using an external pin.

When the A/D conversion request signal to be monitored is selected by the TADSTRGR0 register, a pulse signal is output from the ADSM0 pin that is at the high level when the A/D conversion start request signal is generated, and at the low level in the timer cycle used to generate the A/D conversion start request signal.

Figure 20.99 shows an example of outputting the A/D conversion start request frame synchronization signal.

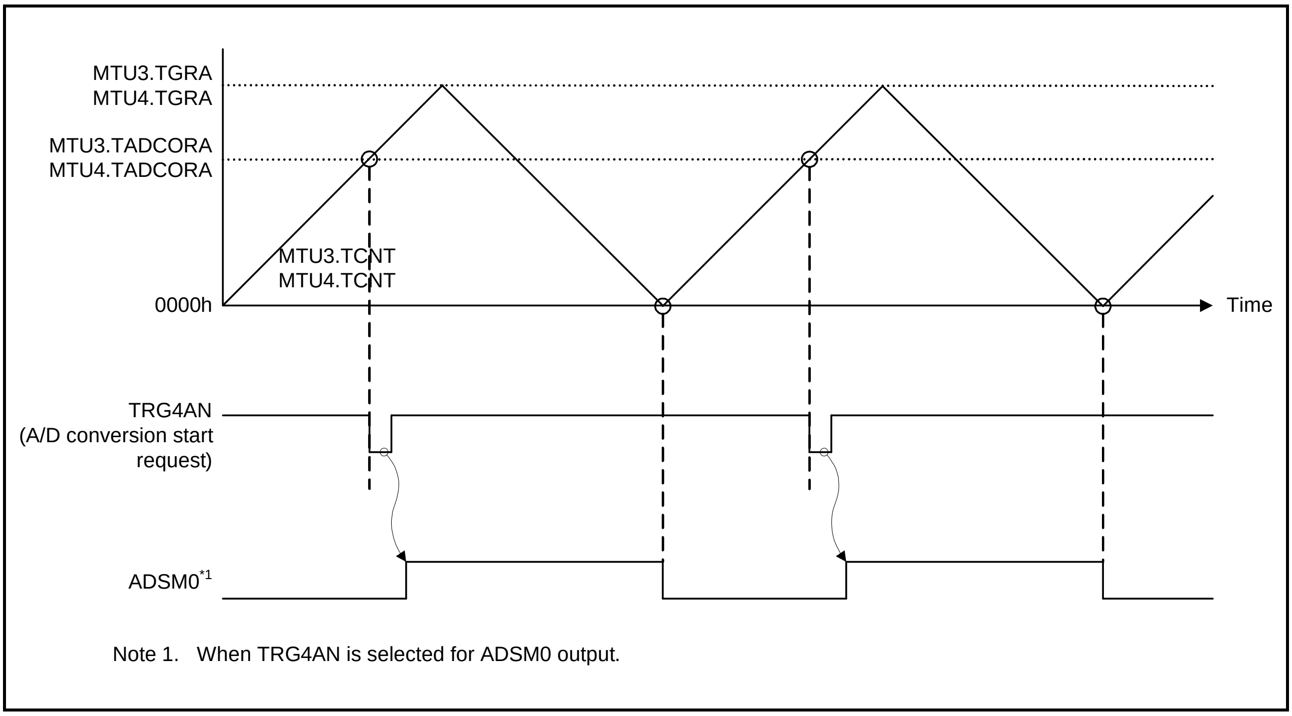


Figure 20.99 Example of Outputting A/D Conversion Start Request Frame Synchronization Signal

20.4 Interrupt Sources

20.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 14, Interrupt Controller (ICUb). Table 20.63 lists the MTU interrupt sources.

Table 20.63 MTU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	High
	TGIB0	MTU0.TGRB input capture/compare match	Possible	↑
	TGIC0	MTU0.TGRC input capture/compare match	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Possible	
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible	
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible	
	TGIV5	MTU5.TGRV input capture/compare match	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	Possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Note 1. Only in complementary PWM mode

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 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

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, clearing an interrupt is requested. The MTU has five overflow interrupts (one for each channel except MTU5).

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1, and MTU2).

20.4.2 DTC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, refer to section 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.

20.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU. Table 20.64 shows 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 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 TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTG bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER is set to 1

When either condition is satisfied, if A/D converter start signal TRGAnN 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/D converter start request signal TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.
When a compare match occurs between MTU0.TCNT and MTU0.TGRE while the TTGE2 bit in MTU0.TIER2 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TRG0N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) 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 MTU4.TADCORA or MTU4.TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (MTU4.TADCR) is set to 1. For details, refer to section 20.3.9, A/D Converter Start Request Delaying Function.
A/D conversion will start when TRG4AN is generated if A/D converter start signal TRG4AN from the MTU is selected as the trigger in the A/D converter, when TRG4BN is generated if TRG4BN from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN is generated if TRG4ABN from the MTU is selected as the trigger in the A/D converter.

Table 20.64 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*1		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA compare match not only with MTU4.TCNT but also with MTU3.TCNT and TCNTSA is detected. Accordingly, when compare match with MTU3.TCNT and TCNTSA occurs, TRGA4N is also generated.
When MTU3 and MTU 4 are made to operate in complementary PWM mode for generating an A/D converter start request, use the A/D converter start request by compare match between MTU4.TCNT and MTU4.TADCORA/B.

20.5 Operation Timing

20.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 20.100 and Figure 20.101 show the TCNT count timing in internal clock operation, Figure 20.102 shows the TCNT count timing in external clock operation (normal mode), and Figure 20.103 shows the TCNT count timing in external clock operation (phase counting mode).

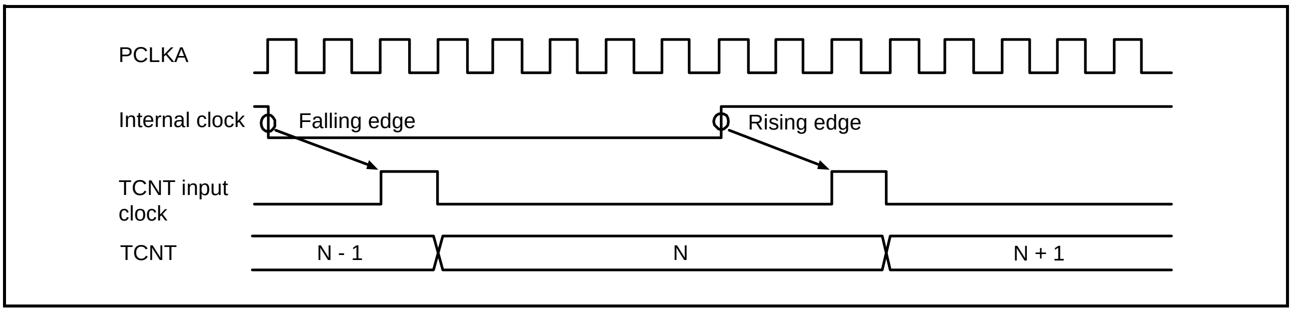


Figure 20.100 Count Timing in Internal Clock Operation (MTU0 to MTU4)

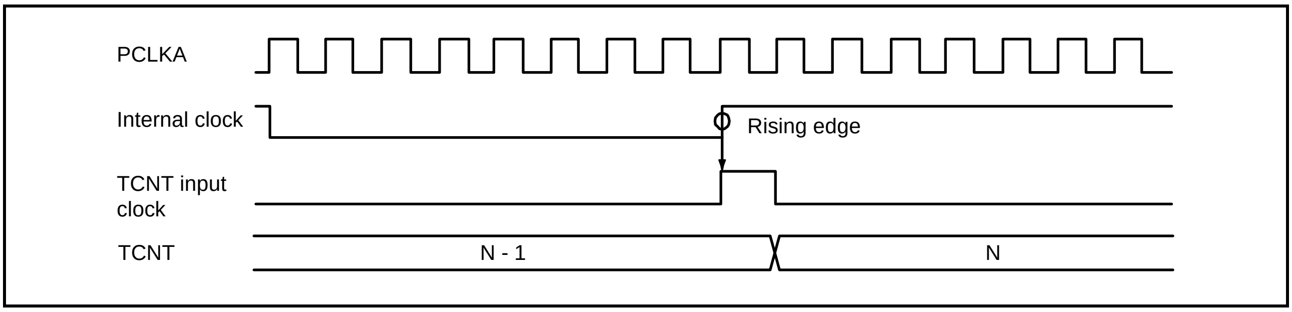


Figure 20.101 Count Timing in Internal Clock Operation (MTU5)

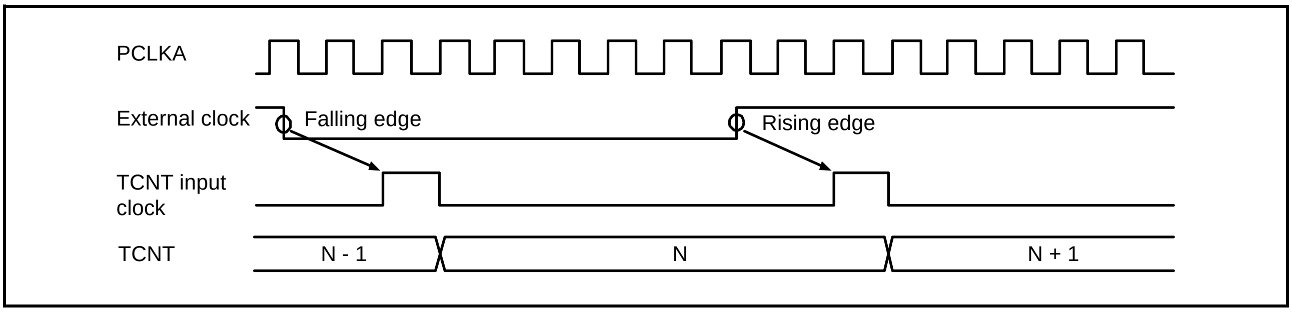


Figure 20.102 Count Timing in External Clock Operation (MTU0 to MTU4)

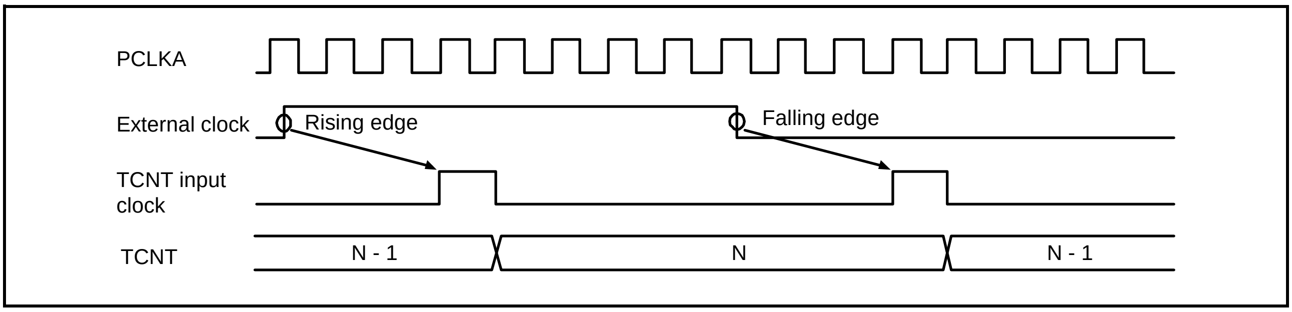


Figure 20.103 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 20.104 shows the output compare output timing (normal mode or PWM mode) and Figure 20.105 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

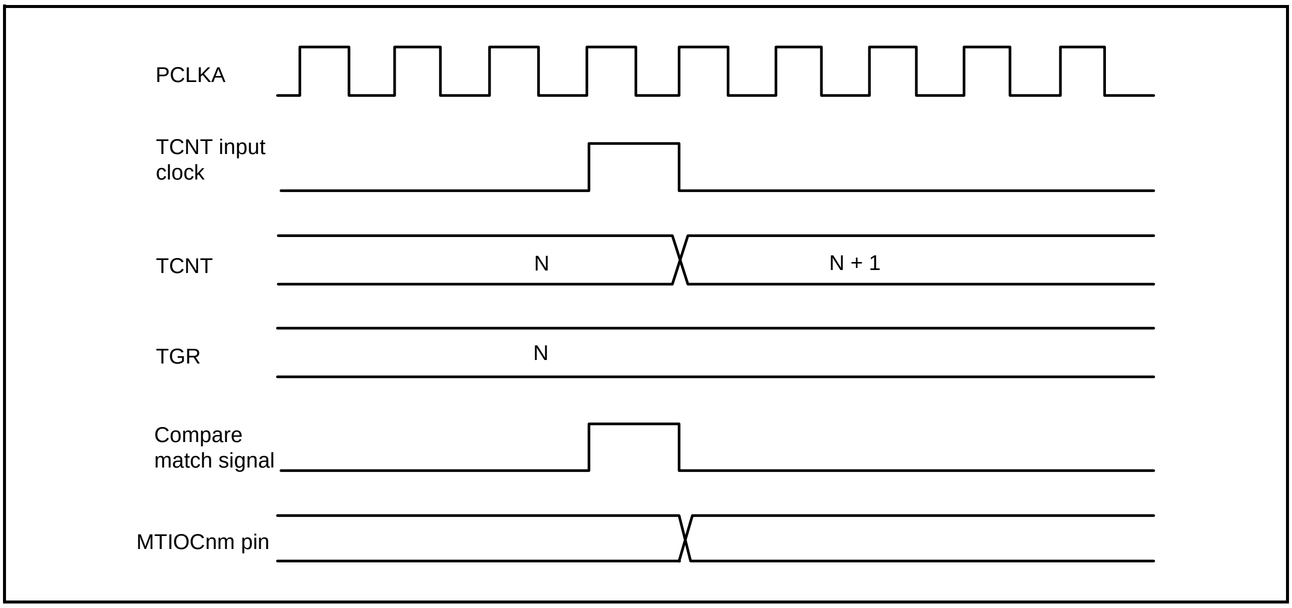


Figure 20.104 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4; m = A to D)

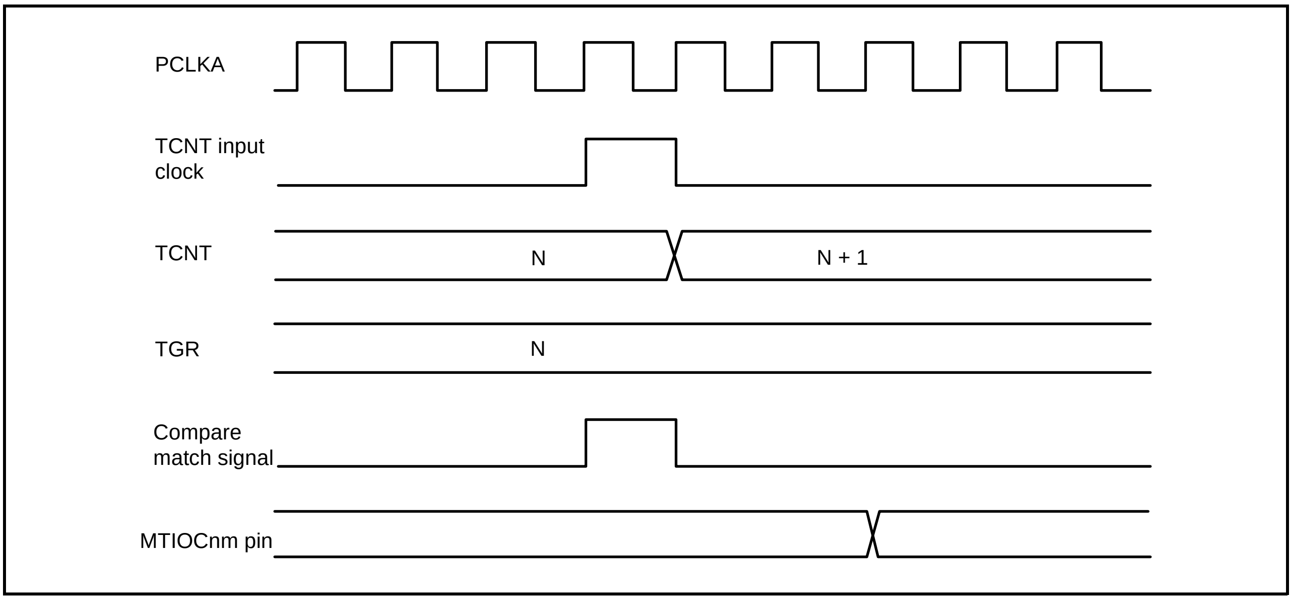


Figure 20.105 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4; m = A to D)

(3) Input Capture Signal Timing

Figure 20.106 shows the input capture signal timing.

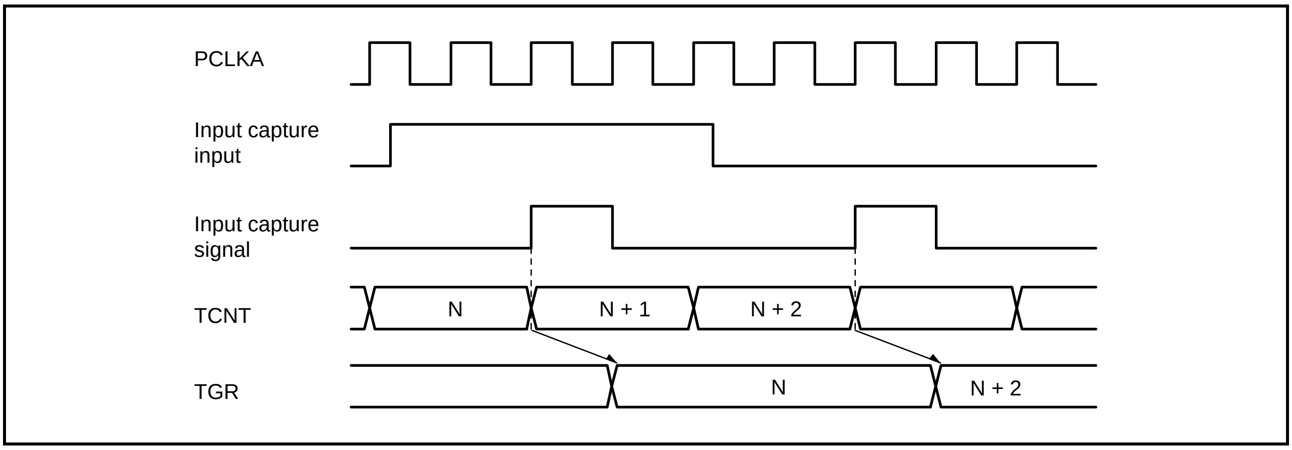


Figure 20.106 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

FFigure 20.107 and Figure 20.108 show the timing when counter clearing on compare match is specified, and Figure 20.109 shows the timing when counter clearing on input capture is specified.

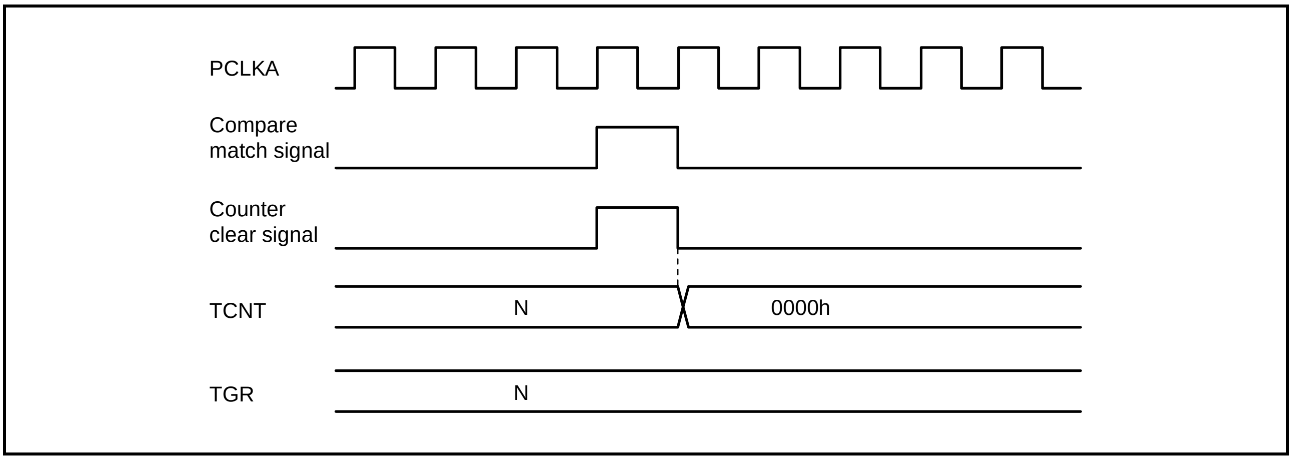


Figure 20.107 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

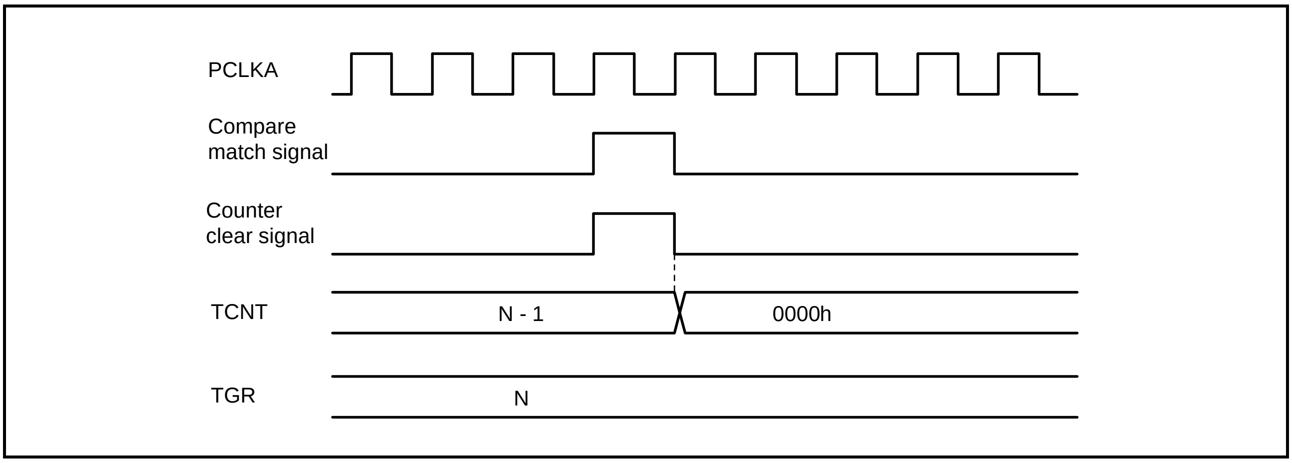


Figure 20.108 Counter Clear Timing (Compare Match) (MTU5)

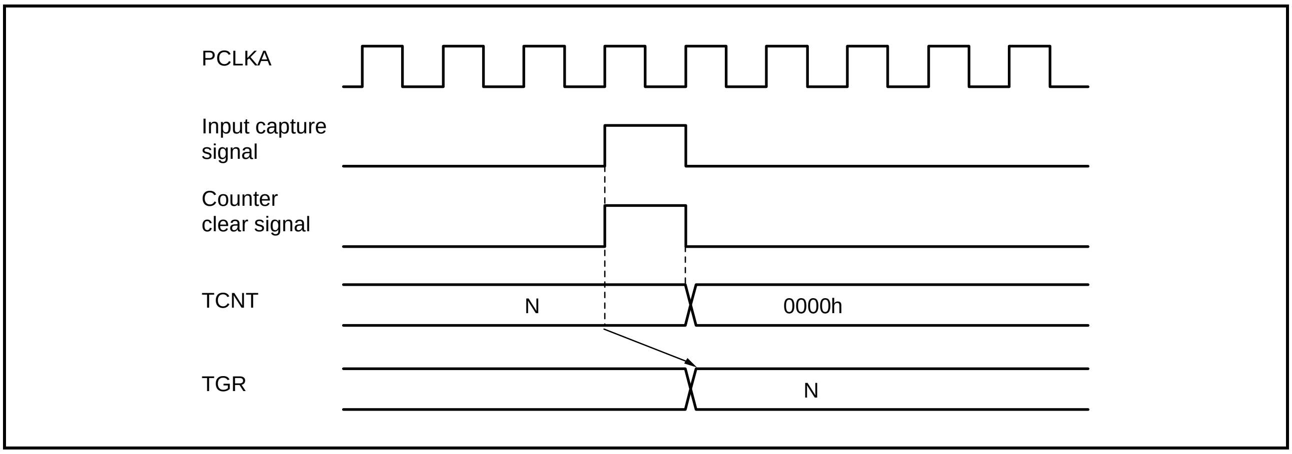


Figure 20.109 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 20.110 to Figure 20.112 show the timing in buffer operation.

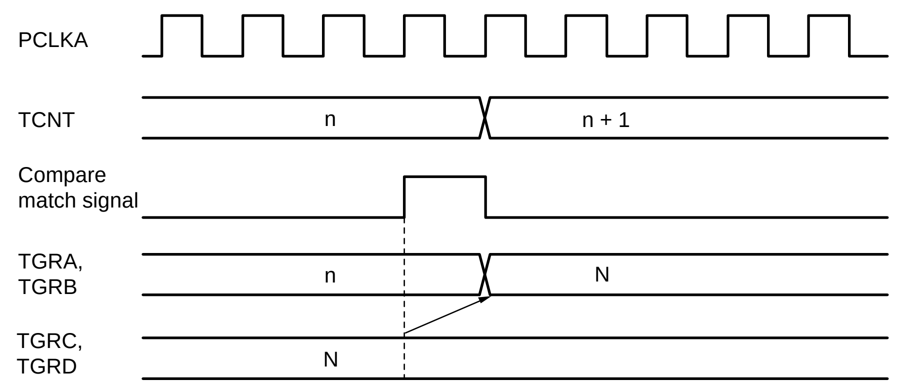


Figure 20.110 Buffer Operation Timing (Compare Match)

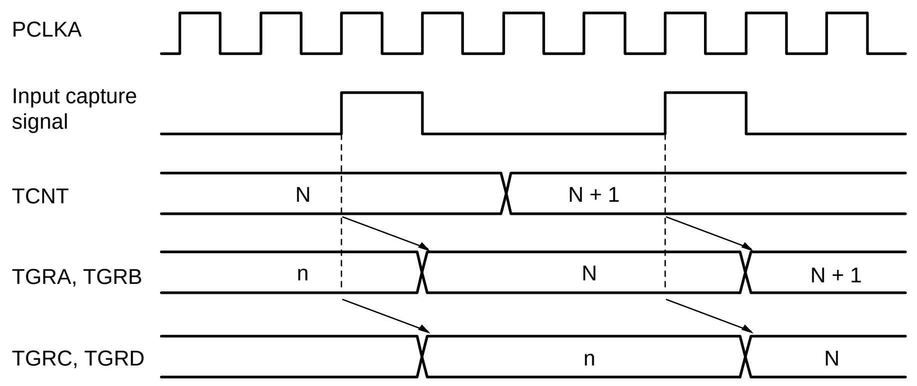


Figure 20.111 Buffer Operation Timing (Input Capture)

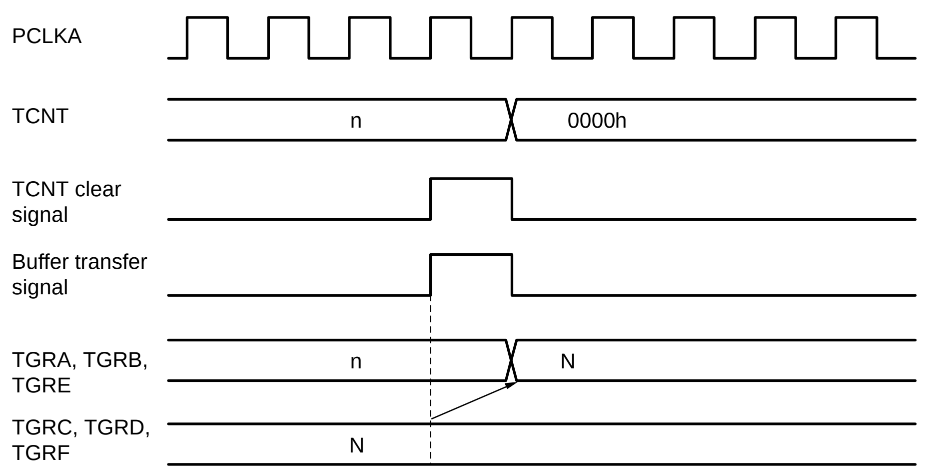


Figure 20.112 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 20.113 to Figure 20.115 show the buffer transfer timing in complementary PWM mode.

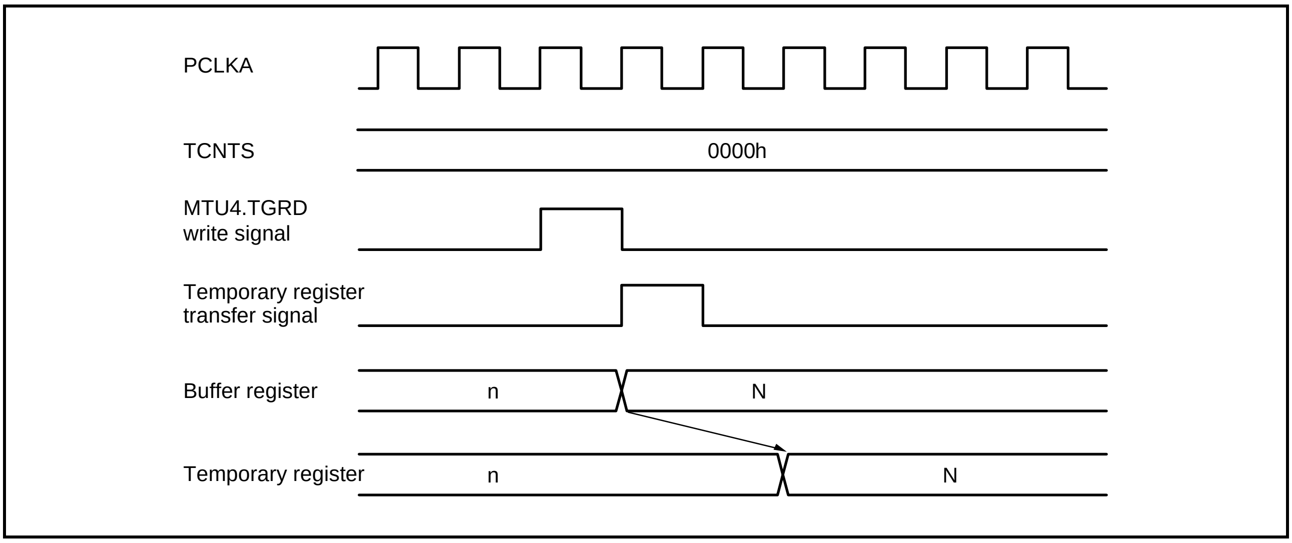


Figure 20.113 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

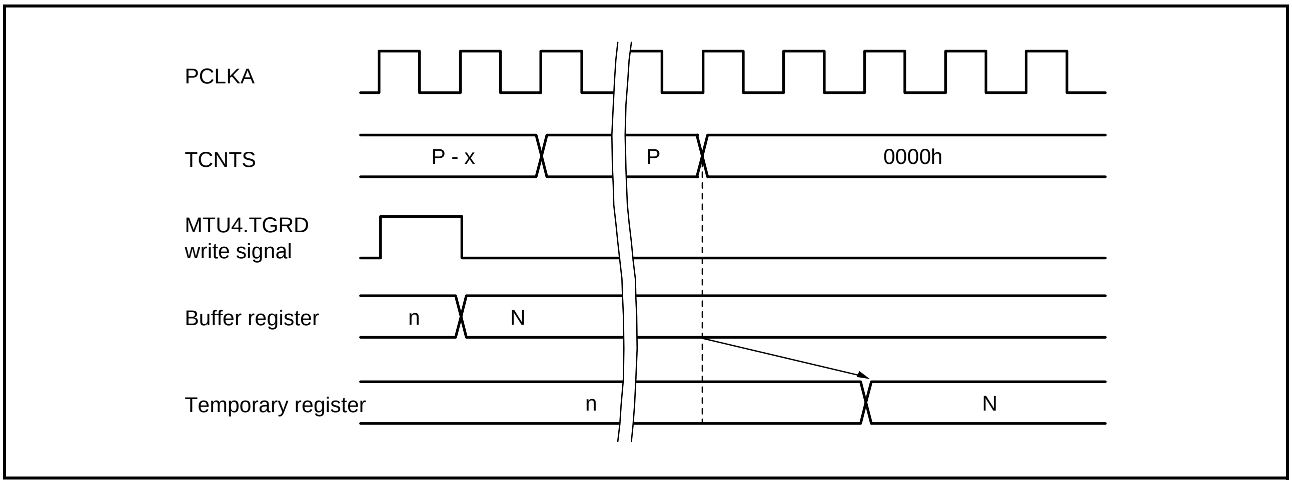


Figure 20.114 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

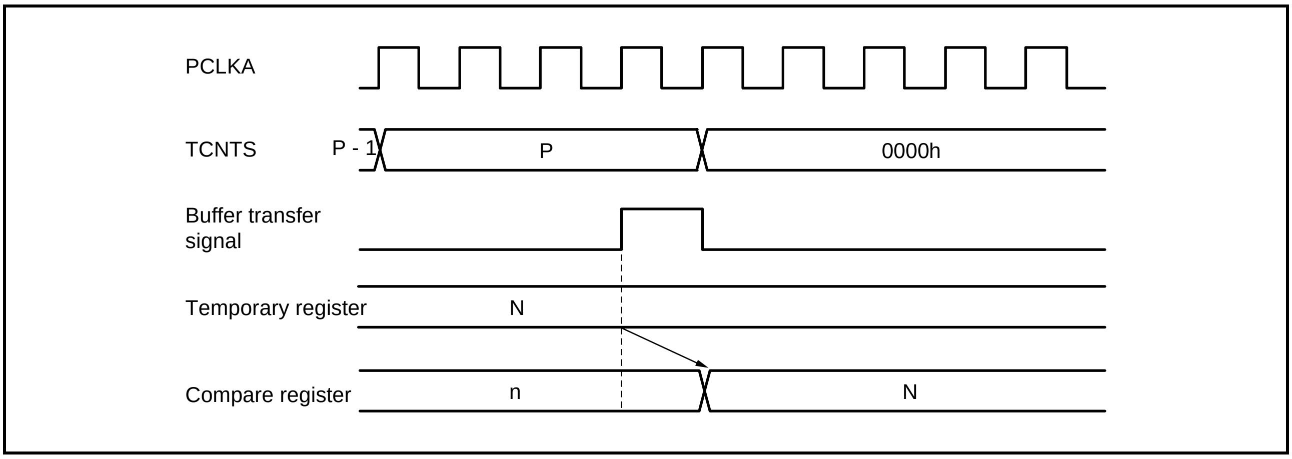


Figure 20.115 Transfer Timing from Temporary Register to Compare Register

20.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 20.116 and Figure 20.117 show the TGI interrupt request signal timing when a compare match occurs.

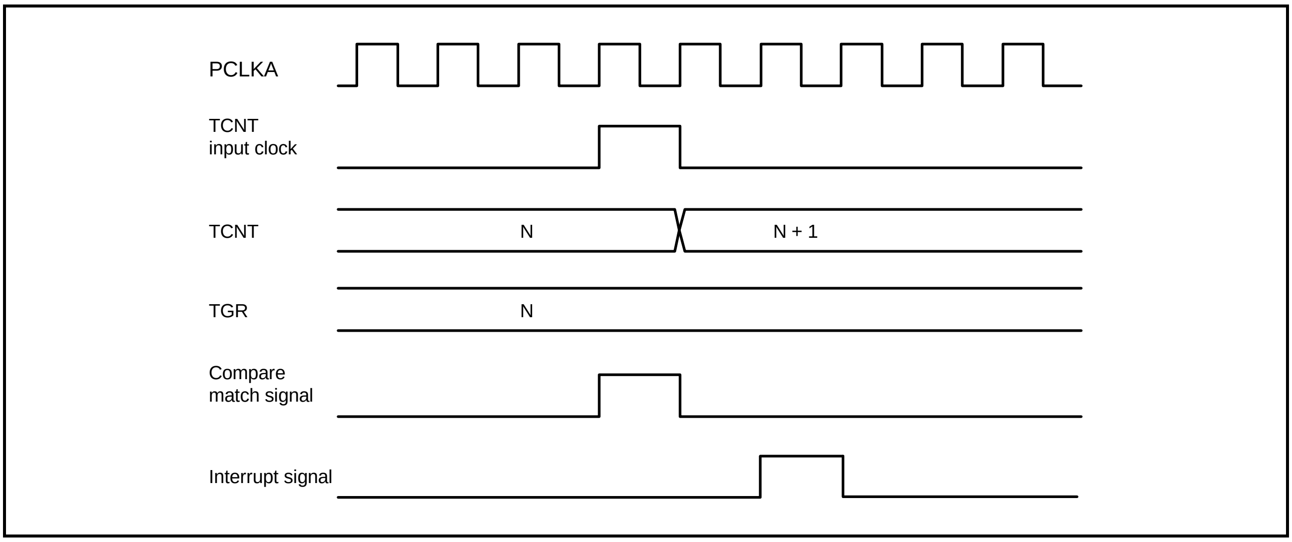


Figure 20.116 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

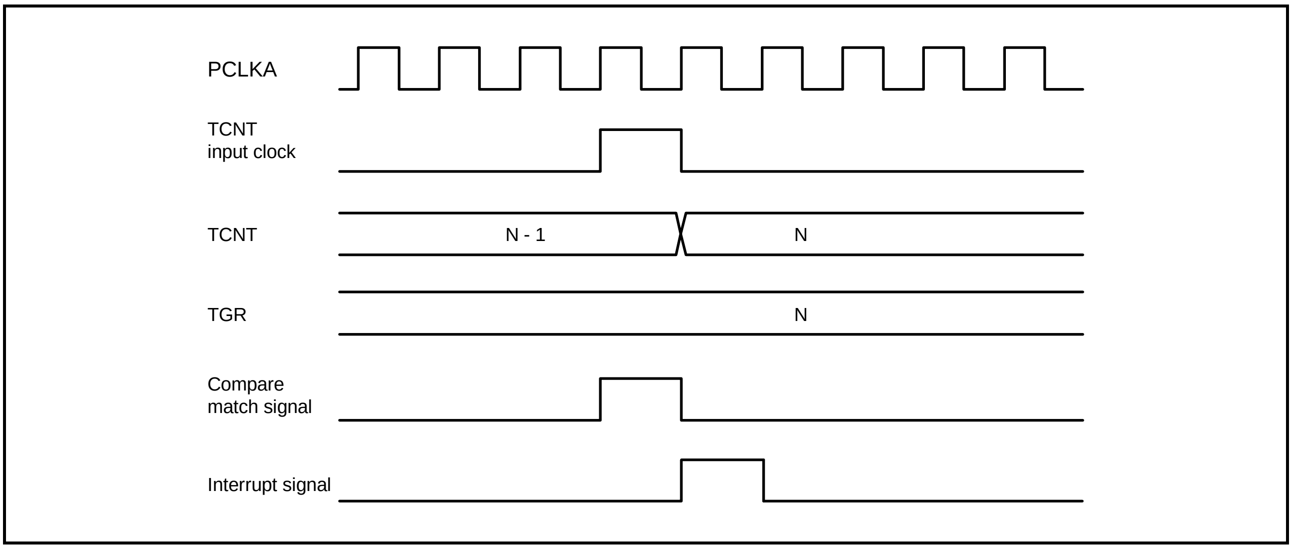


Figure 20.117 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 20.118 and Figure 20.119 show the TGI interrupt request signal timing when an input capture occurs.

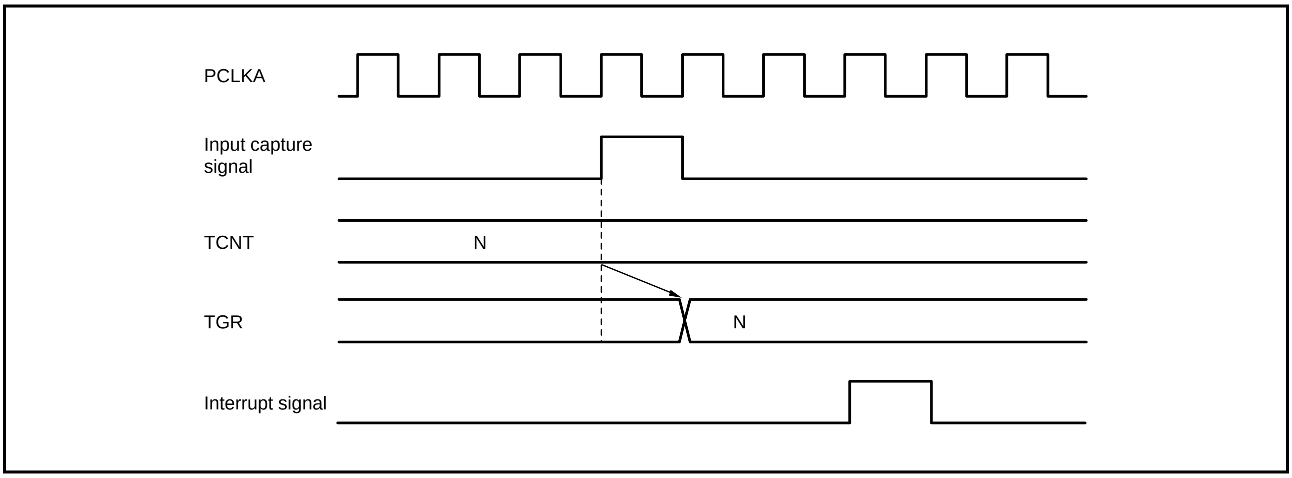


Figure 20.118 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

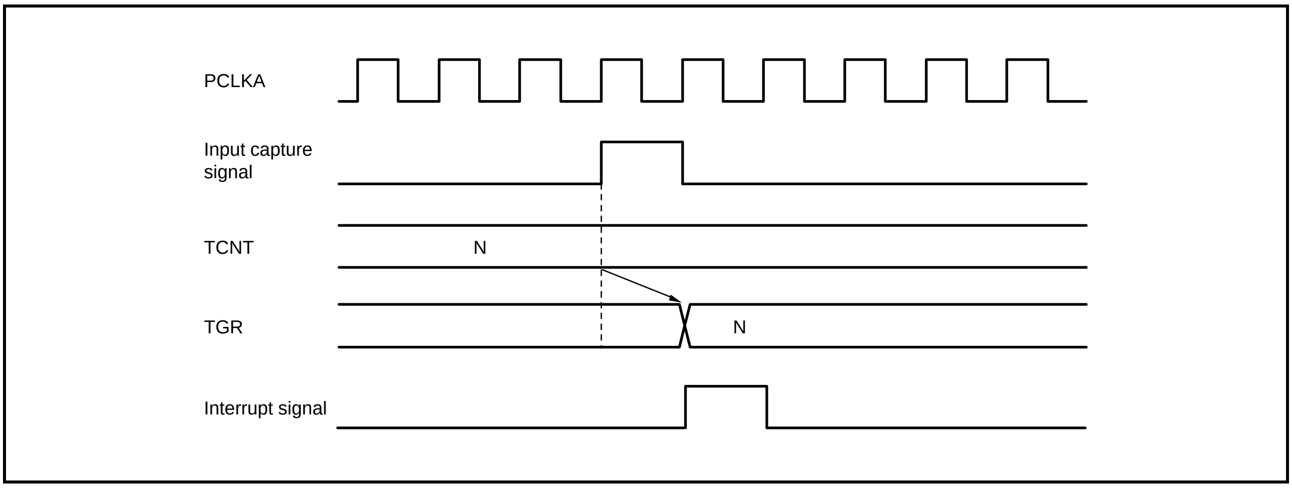


Figure 20.119 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 20.120 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 20.121 shows the TCIU interrupt request signal timing when an underflow is generated.

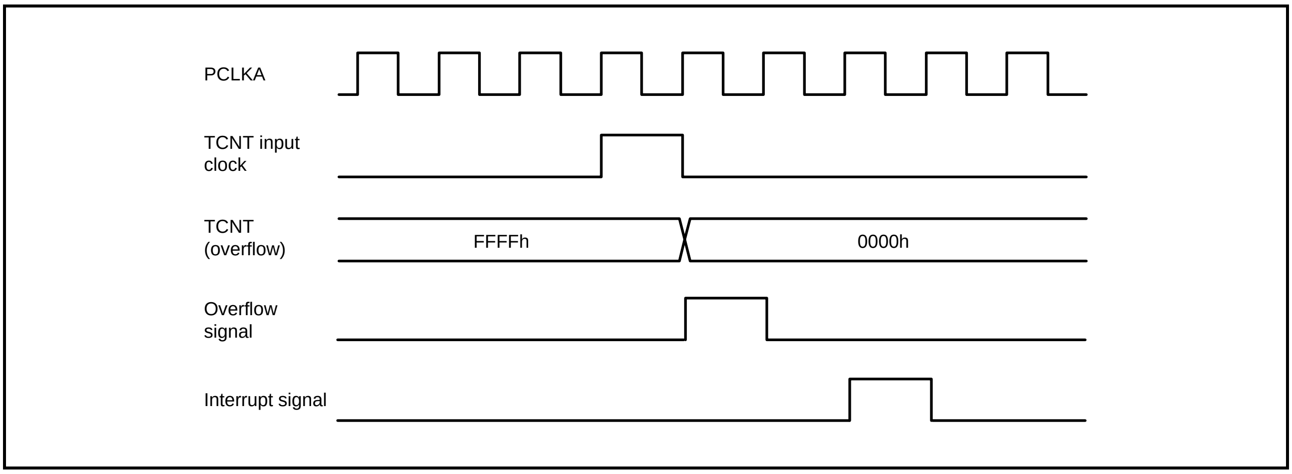


Figure 20.120 TCIV Interrupt Timing

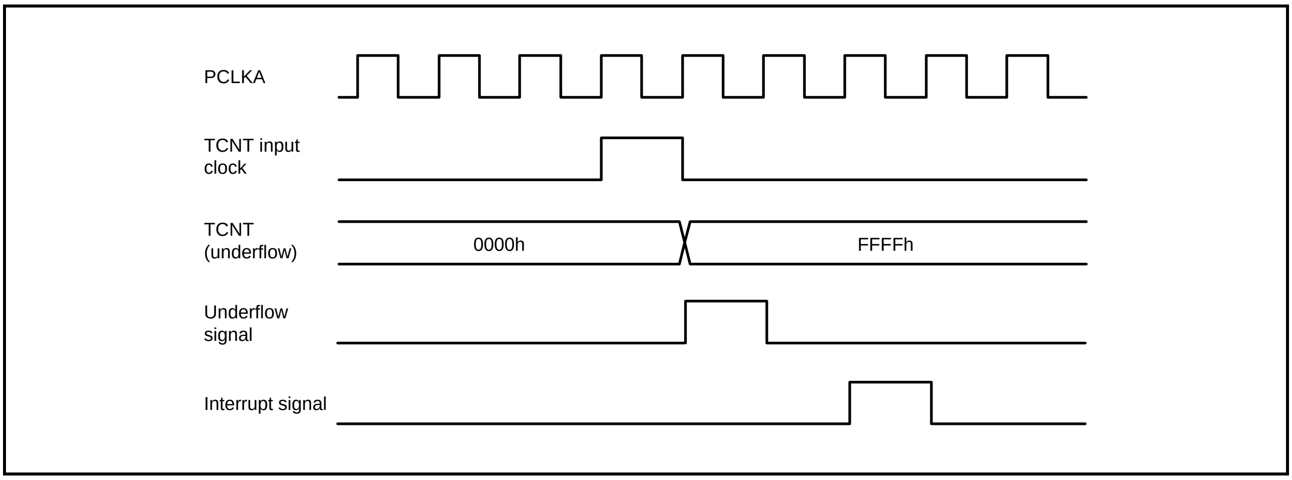


Figure 20.121 TCIU Interrupt Timing

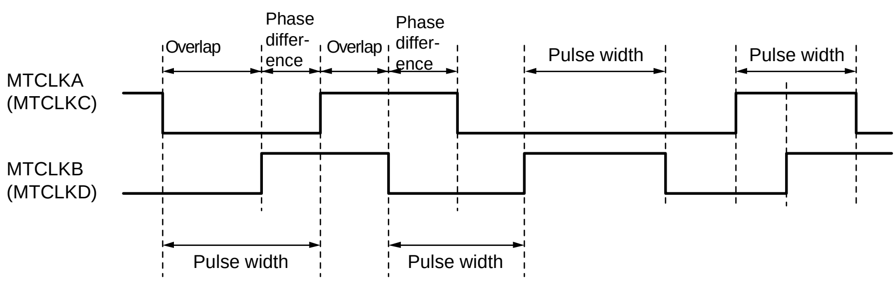
20.6 Usage Notes

20.6.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, refer to section 11, Low Power Consumption.

20.6.2 Input Clock Restrictions

The input clock pulse width must be at least three PCLKA clocks for single-edge detection, and at least five PCLKA clocks 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 three PCLKA clocks, and the pulse width must be at least five PCLKA clocks. Figure 20.122 shows the input clock conditions in phase counting mode.



Note: Phase difference and overlap: 3 PCLKA clock or more
Note: Pulse width: 5 PCLKA clock

Figure 20.122 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

20.6.3 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 TCNU updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]
N: TGR setting

20.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.

Figure 20.123 shows the timing in this case.

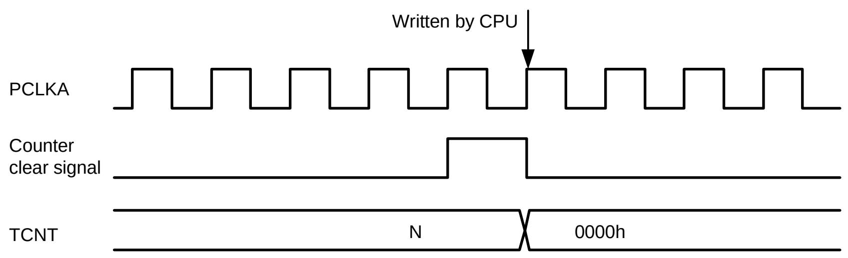


Figure 20.123 Contention between TCNT Write and Clear Operations

20.6.5 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 20.124 shows the timing in this case.

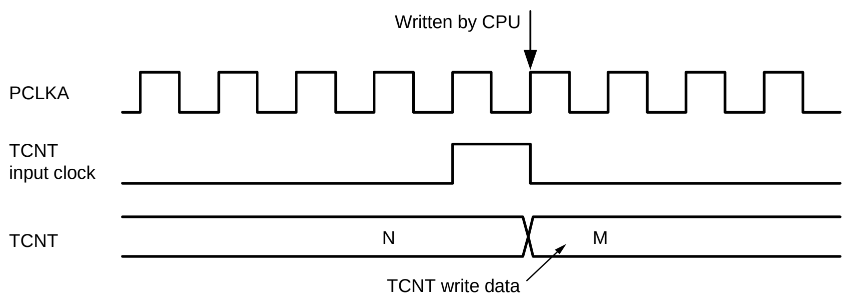


Figure 20.124 Contention between TCNT Write and Increment Operations

20.6.6 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 20.125 shows the timing in this case.

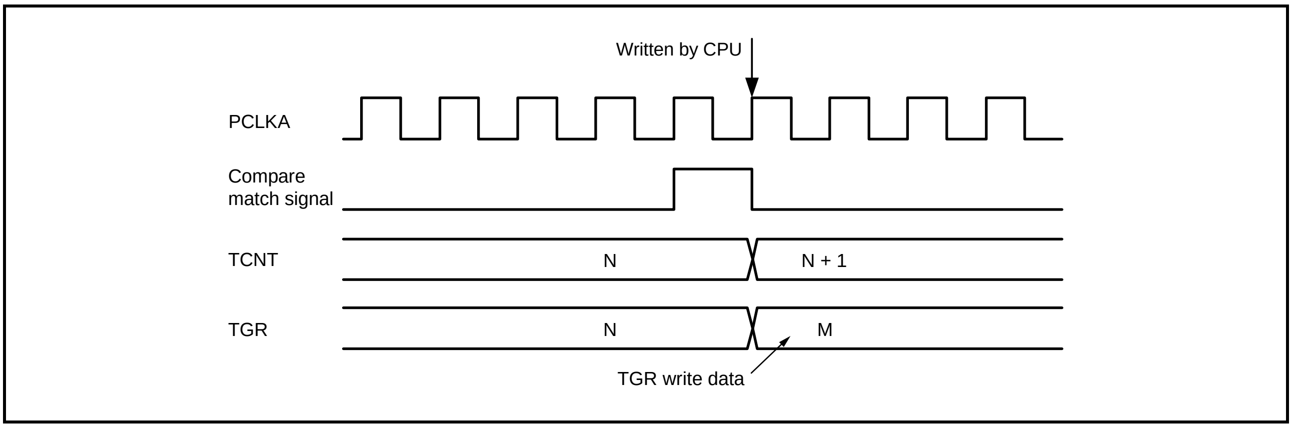


Figure 20.125 Contention between TGR Write Operation and Compare Match

20.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 20.126 shows the timing in this case.

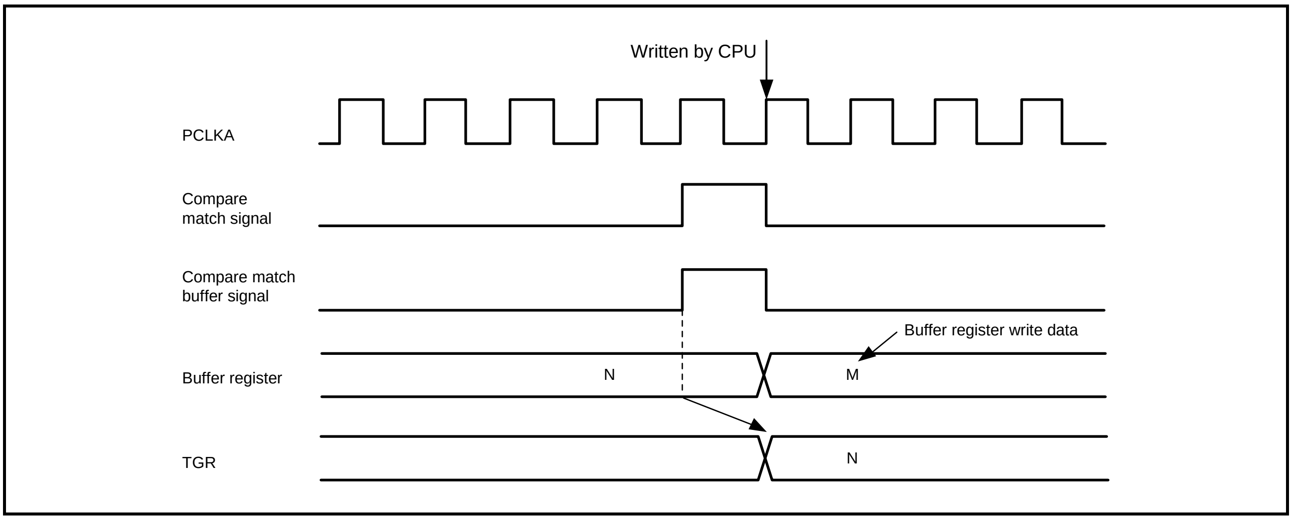


Figure 20.126 Contention between Buffer Register Write Operation and Compare Match

20.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 20.127 shows the timing in this case.

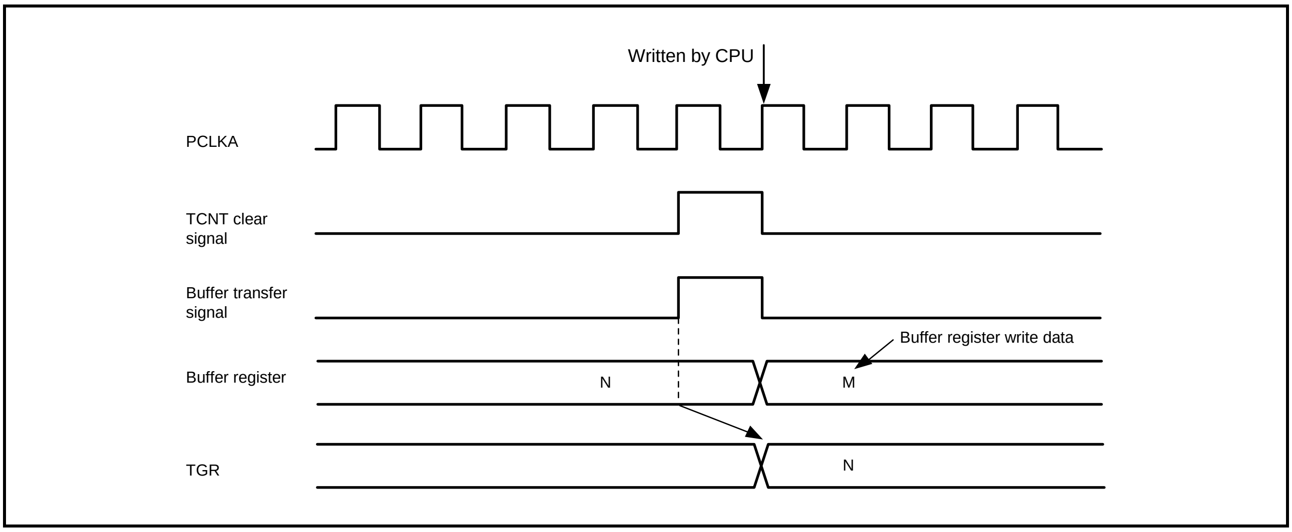


Figure 20.127 Contention between Buffer Register Write and TCNT Clear Operations

20.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 20.128 shows the timing in this case.

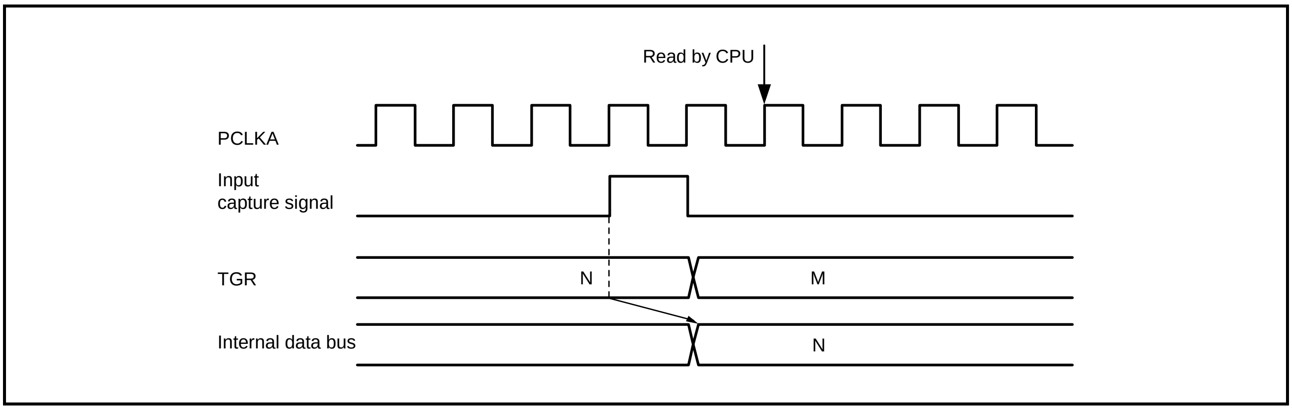


Figure 20.128 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

20.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.
Figure 20.129 and Figure 20.130 show the timing in this case.

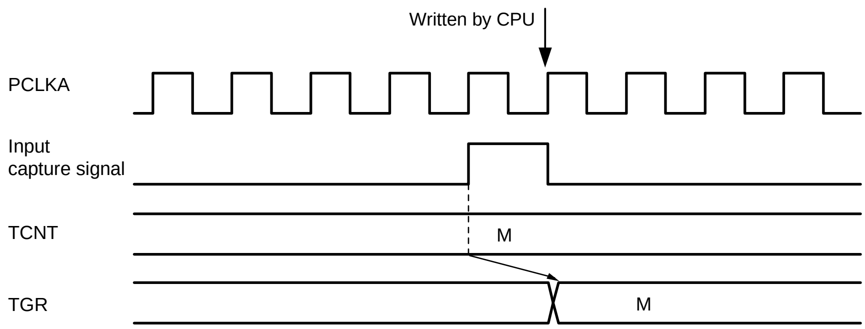


Figure 20.129 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

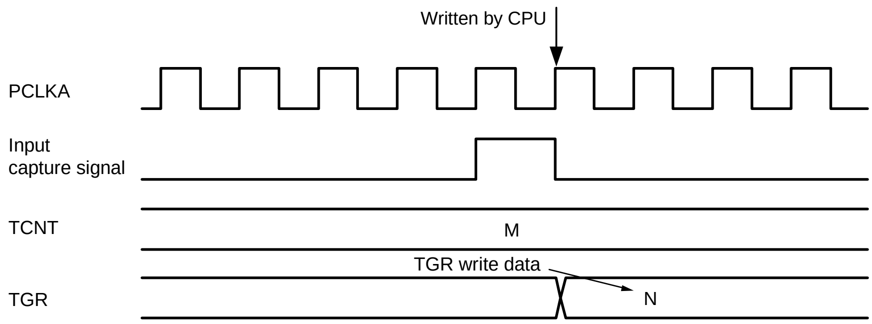


Figure 20.130 Contention between TGR Write Operation and Input Capture (MTU5)

20.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 20.131 shows the timing in this case.

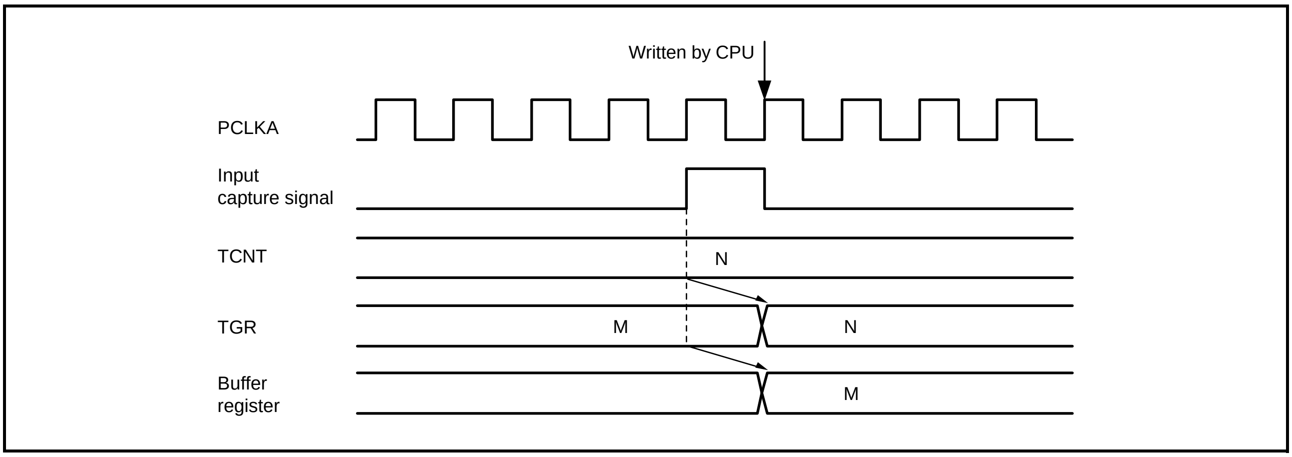


Figure 20.131 Contention between Buffer Register Write Operation and Input Capture

20.6.12 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 cycle, 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 20.132 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

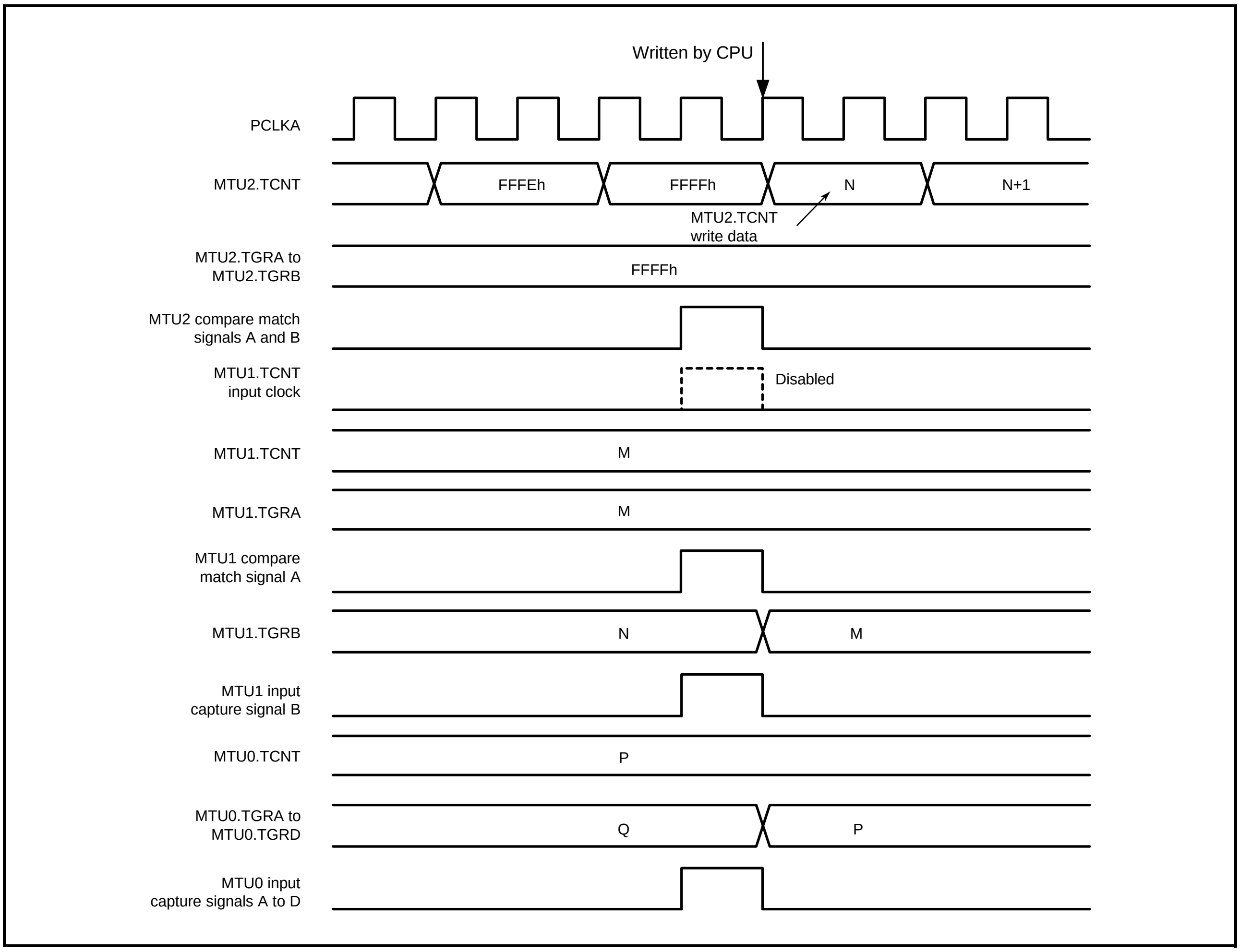


Figure 20.132 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

20.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, the MTU3.TCNT value is set to the timer dead time register (TDDRA) 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 20.133 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

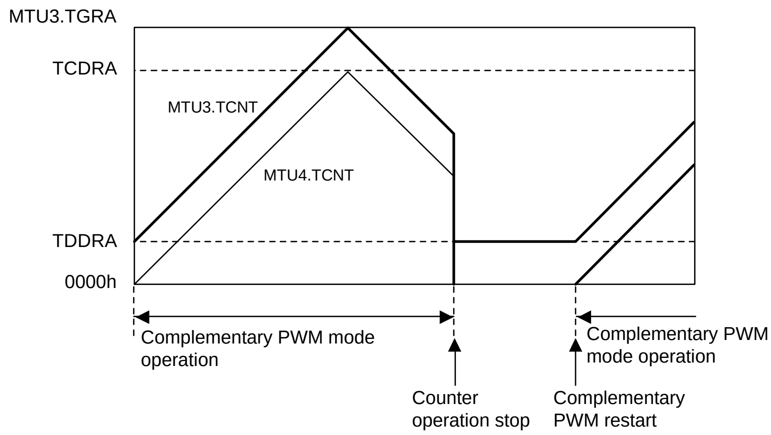


Figure 20.133 Counter Value When Stopped in Complementary PWM Mode

20.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDRA), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 to 0. The MTIOC4C pin cannot output waveforms if the BFA bit in MTU4.TMDR1 is set to 1. Likewise, the MTIOC4D pin cannot output waveforms if the BFB bit in MTU4.TMDR1 is set to 1.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR1. When the BFA bit in MTU3.TMDR1 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 TCBRA functions as a buffer register for TCDRA.

20.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 to 0. The MTIOC4C pin cannot output waveforms if the BFA bit in MTU4.TMDR1 is set to 1. Likewise, the MTIOC4D pin cannot output waveforms if the BFB bit in MTU4.TMDR1 is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA and BFB bits of MTU3.TMDR1. For example, if the BFA bit in MTU3.TMDR1 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, a TGImn interrupt (m = C, D; n = 3, 4) is not generated.

Figure 20.134 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3, and MTIOC4 operation with the BFA and BFB bits in MTU3.TMDR1 set to 1 and the BFA and BFB bits in MTU4.TMDR1 set to 0.

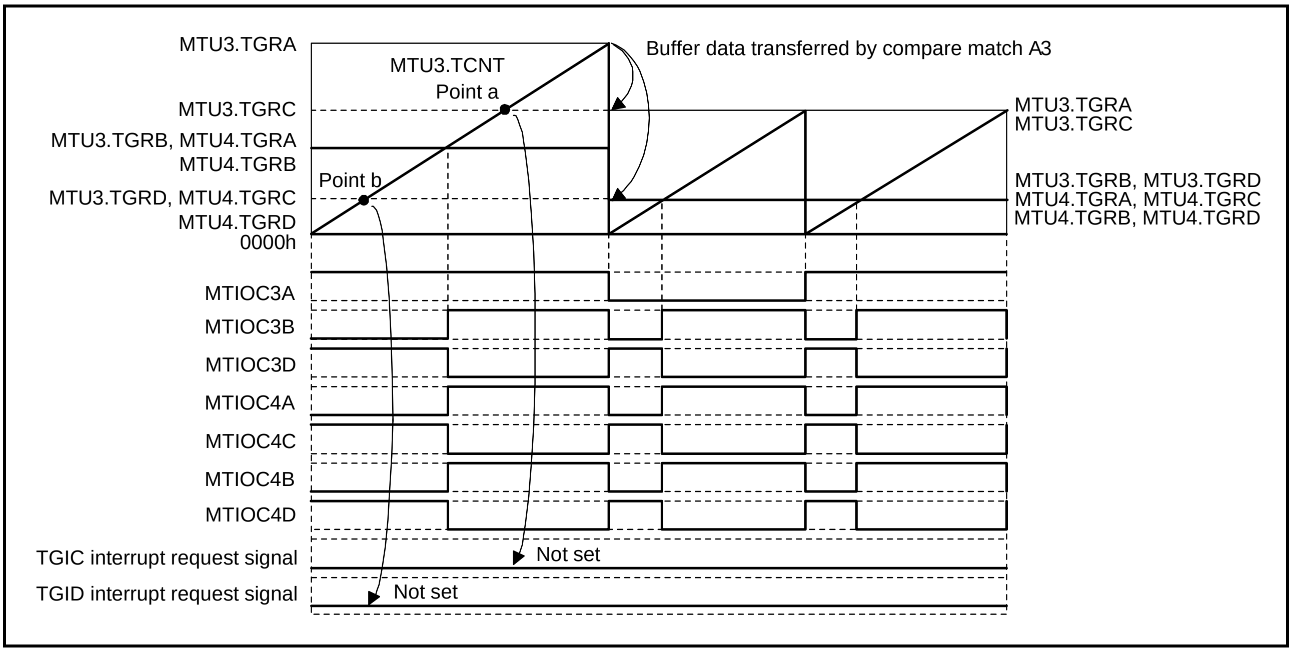


Figure 20.134 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

20.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the CST3 bit of TSTRA 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, a TCIVn interrupt (n = 3, 4) is not generated.

Figure 20.135 shows an example of operation in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source.

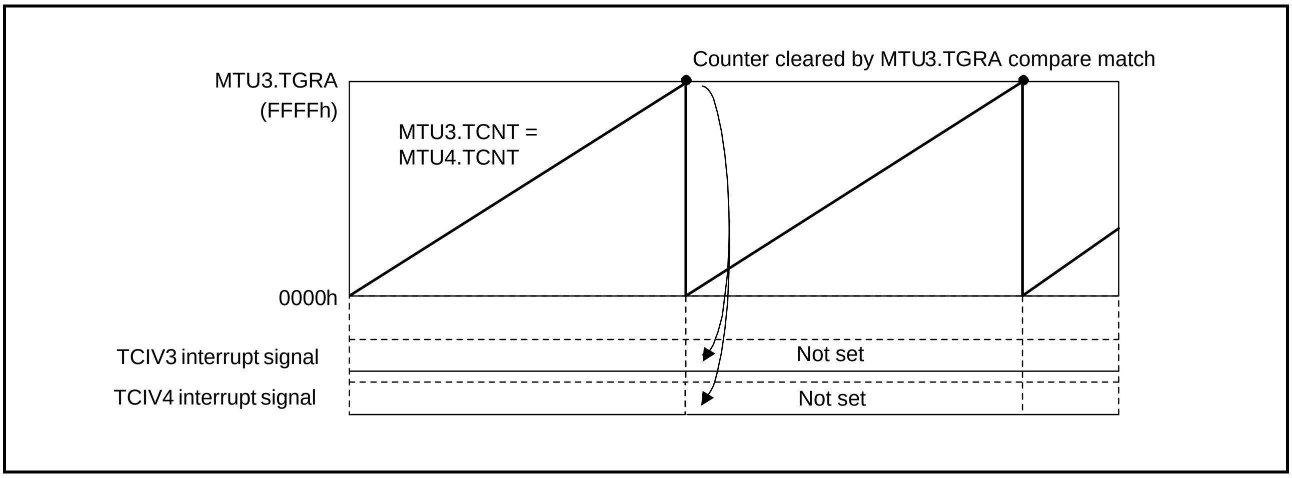


Figure 20.135 Overflow in Reset-Synchronized PWM Mode

20.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4) nor a TCIUn interrupt (n = 1, 2) is not generated and TCNT clearing takes precedence.
Figure 20.136 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

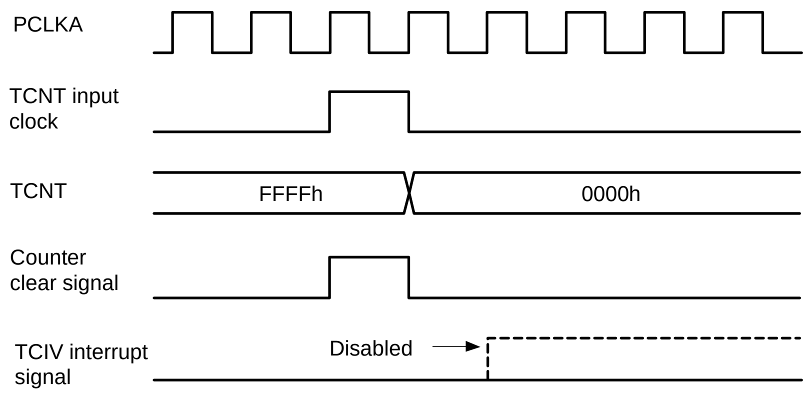


Figure 20.136 Contention between Overflow and Counter Clearing

20.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 20.137 shows the operation timing when there is contention between TCNT write operation and overflow.

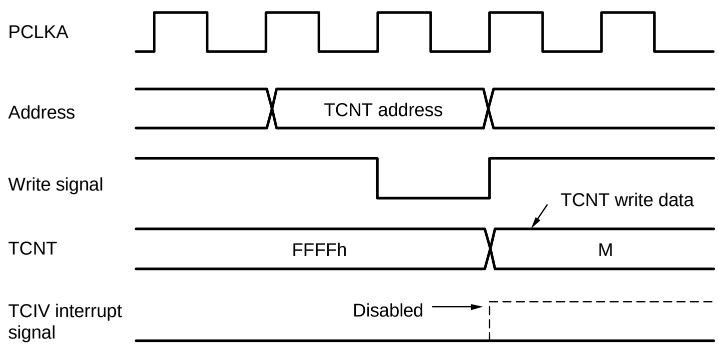


Figure 20.137 Contention between TCNT Write Operation and Overflow

20.6.19 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.

20.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 00h.
The output level in negative phase when the TDER.TDER bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1.OLSN bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1.OLSP bit.

20.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.
For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.
The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 20.2.11, Timer Input Capture Control Register (TICCR).

20.6.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.

(1) When the number skipped is zero for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
- The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKA clock (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
- The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKA clock (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).

(2) When the number skipped is one or more for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
- The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKA (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

20.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to section 21, Port Output Enable 3 (POE3b).

20.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRj (j = U, V, W) bit to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) register is stopped. If an MTU5.TGRj (j = U, V, W) bit is set to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.
In this case, if the compare match enable bit (MTU5.TIER.TGIE5j (j = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTj (j = U, V, W) are enabled or disabled.

20.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 20.138, 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 20.139, 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 ≤ TDDR) set to at least double the value of the TDDRA register.

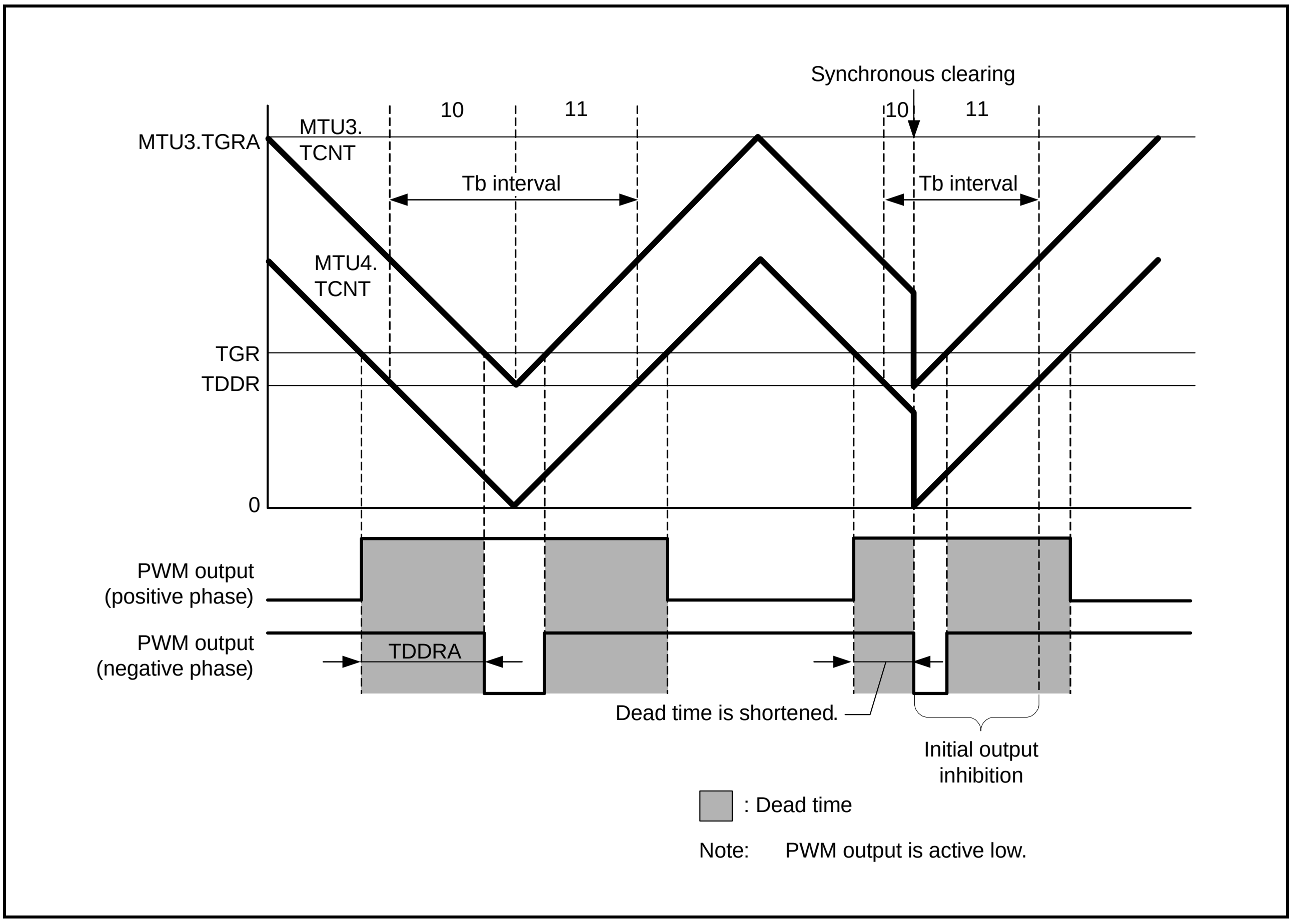


Figure 20.138 Example of Synchronous Clearing (When Condition 1 Applies)

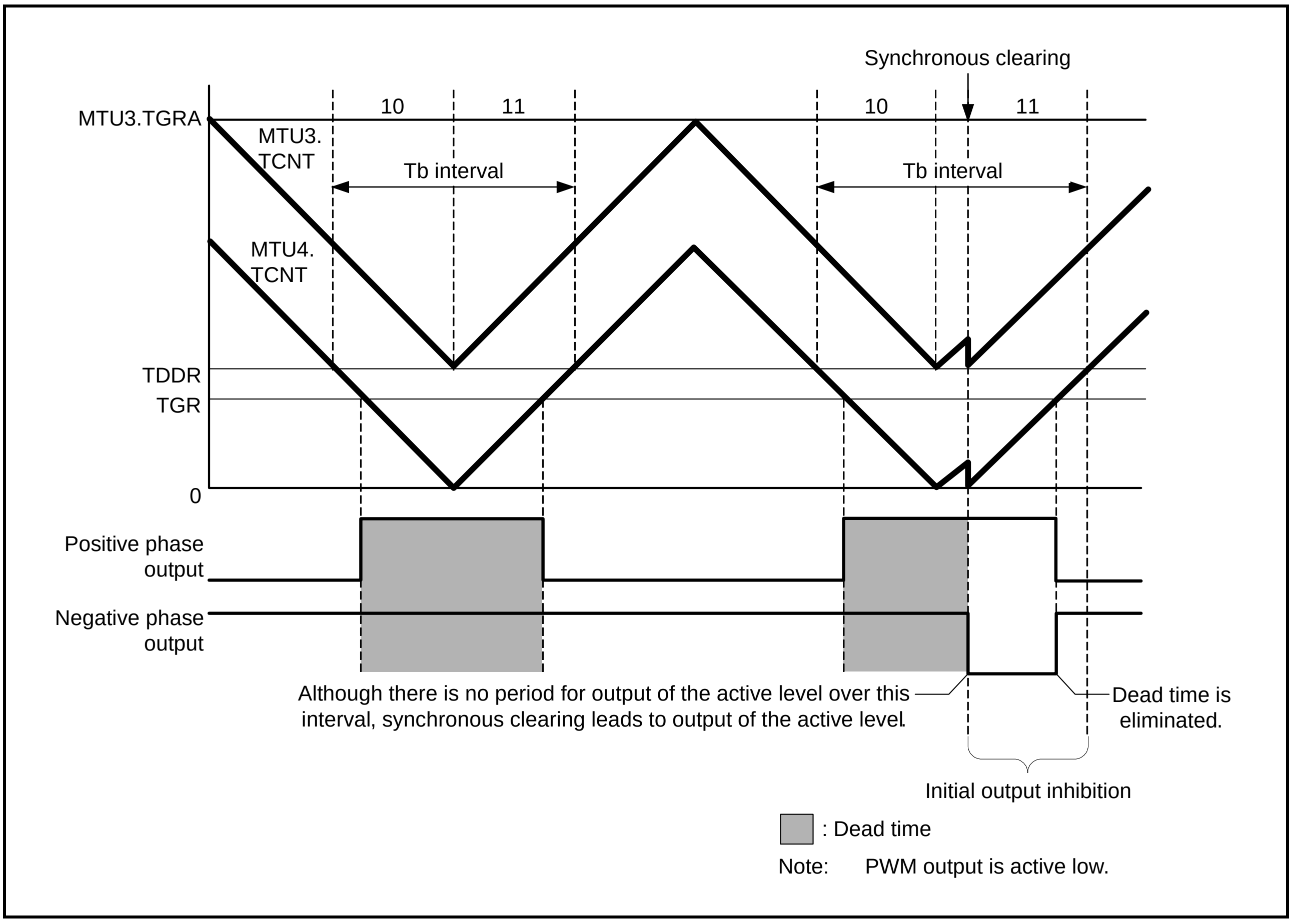


Figure 20.139 Example of Synchronous Clearing (When Condition 2 Applies)

20.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, the PCLKA/1 clock is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the 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 20.140 shows the timing for continuous output of the interrupt signal in response to a compare match.

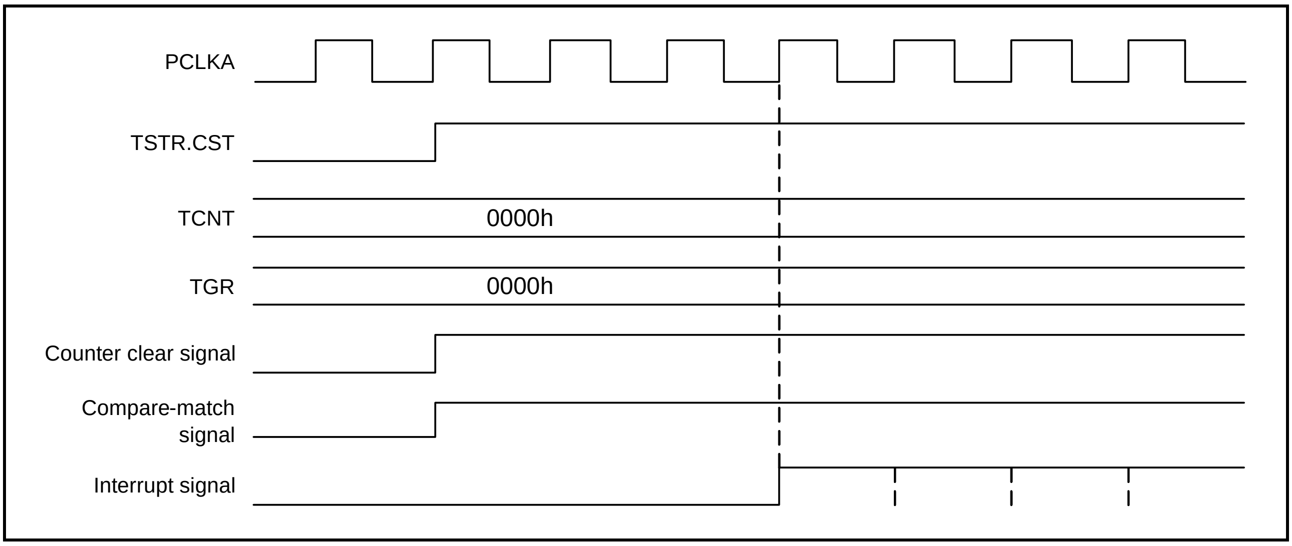


Figure 20.140 Continuous Output of Interrupt Signal in Response to a Compare Match

20.6.27 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to 0 and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during up-counting immediately after transfer. Refer to Figure 20.141.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to the same value as the TRCR and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D converter start request is issued during down-counting immediately after transfer. Refer to Figure 20.142.
- To issue an A/D converter start request linked with the interrupt skipping function, set the MTU4.TADCORA and MTU4.TADCORB registers so that $2 \leq$ MTUn.TADCORA/TADCORB $\leq$ TCDR - 2 is satisfied (n = 4).

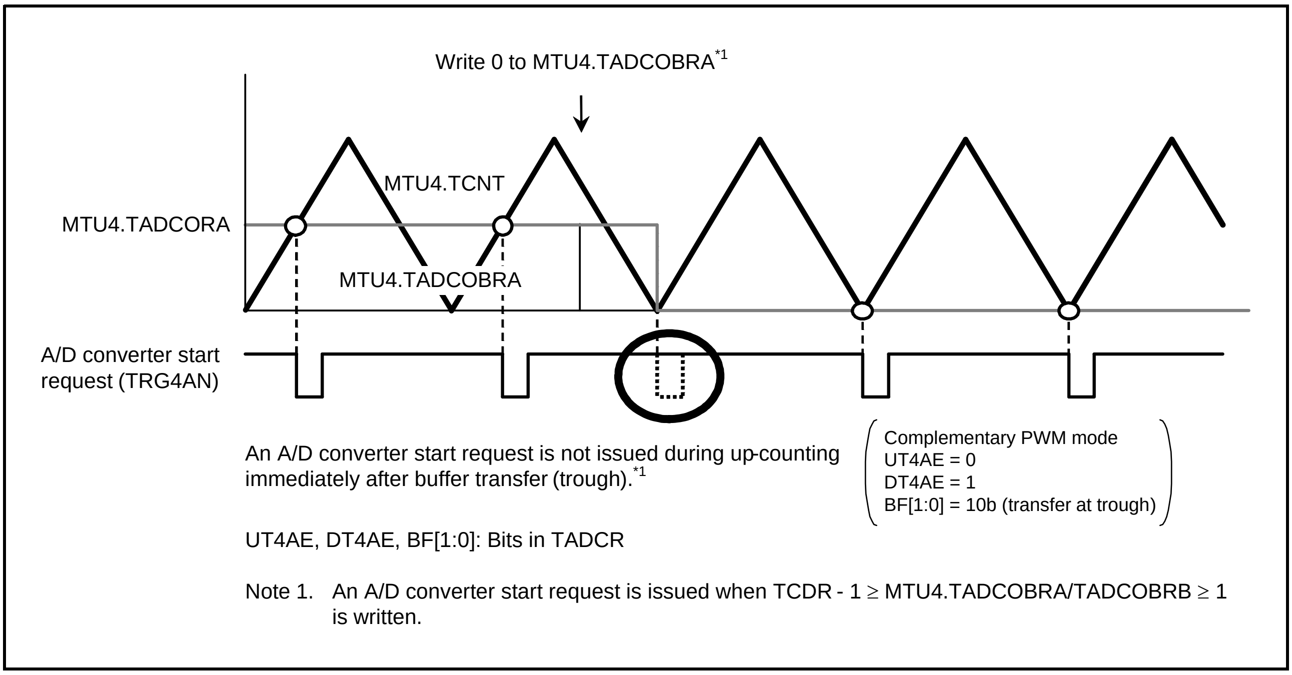


Figure 20.141 A/D Converter Start Request When 0 is Written to MTU4.TADCOBRA

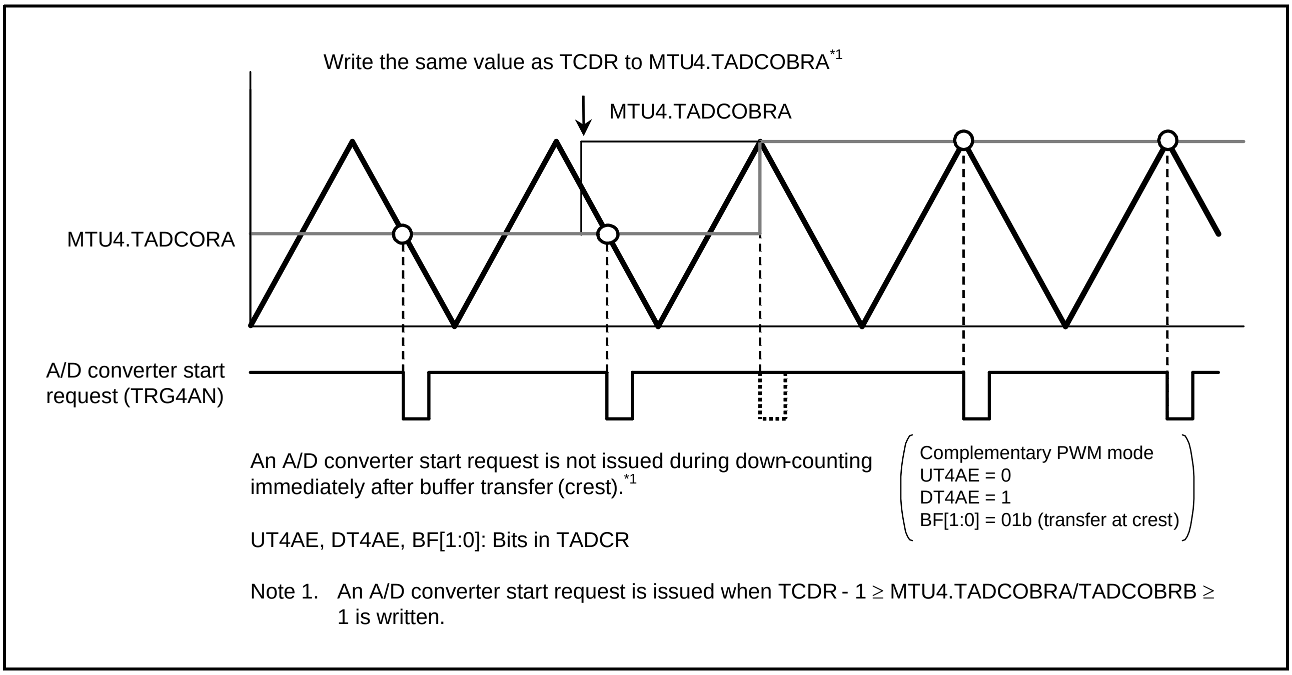


Figure 20.142 A/D Converter Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

20.7 MTU Output Pin Initialization

20.7.1 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 5 (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.

20.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C and MTIOC4D) should be specified through TOERA setting. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.
The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 20.65.

Table 20.65 Mode Transition Combinations

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

20.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4) pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
- In PWM mode 2, waveforms are not output to the cycle register pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR1A or TOCR2A) setting, temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOERA). At this time, if no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 20.65. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 20.143 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

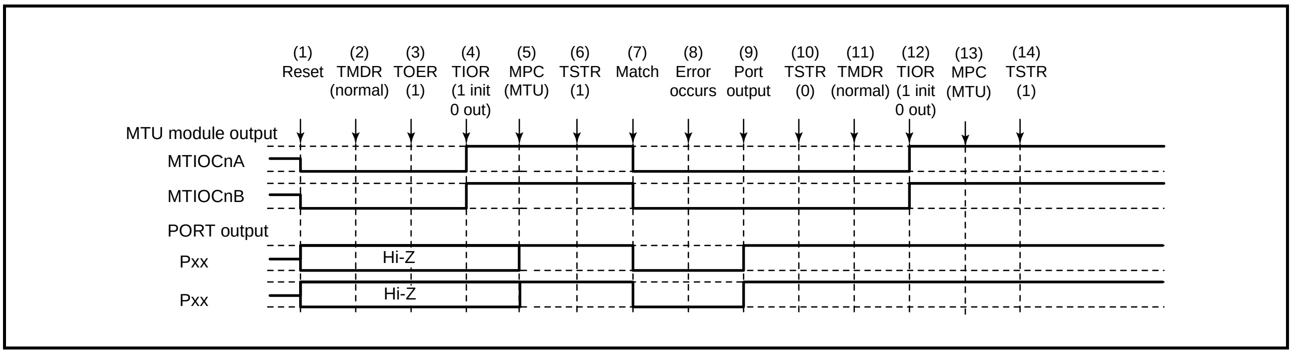


Figure 20.143 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR1 setting is for normal mode.
(3) For MTU3 and MTU4, enable output with TOERA 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) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting TSTRA.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting TSTRA.
(11) This step is not necessary when restarting in normal mode.
(12) Initialize the pins with TIOR.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 20.144 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

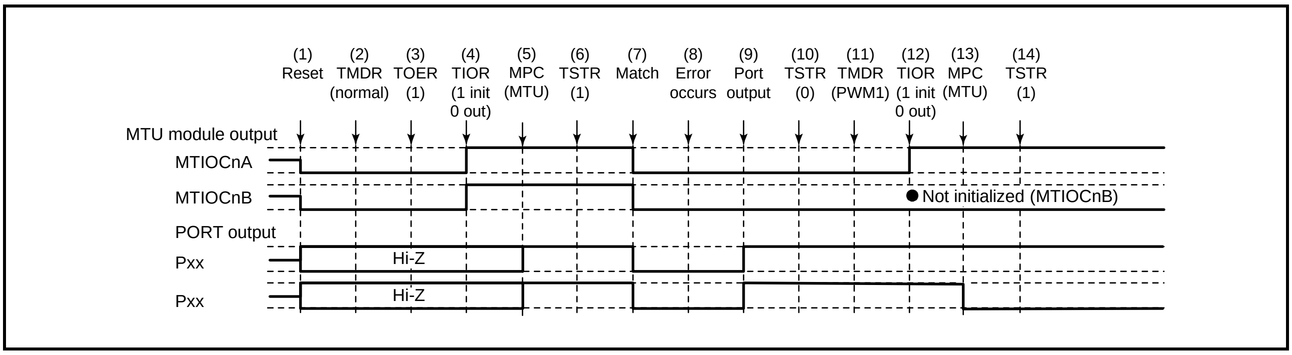


Figure 20.144 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 20.143.
(11) Set PWM mode 1.
(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 20.145 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

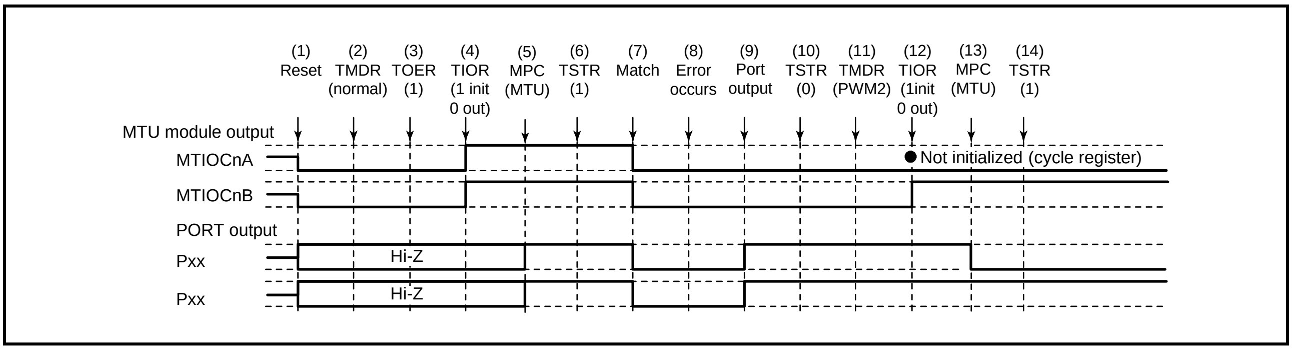


Figure 20.145 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 20.143.
(11) Set PWM mode 2.
(12) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOERA setting is not necessary.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 20.146 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

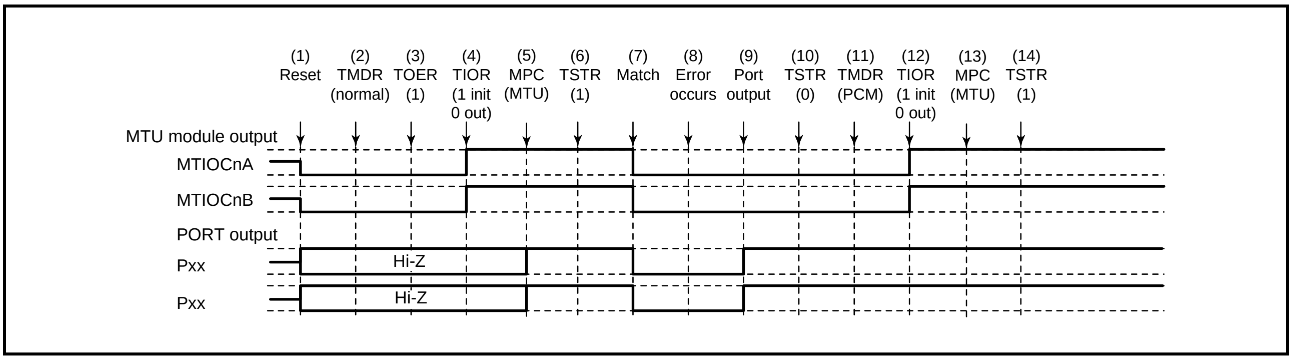


Figure 20.146 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 20.143.
(11) Set the phase counting mode.
(12) Initialize the pins with TIOR.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOERA setting is not necessary.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.147 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

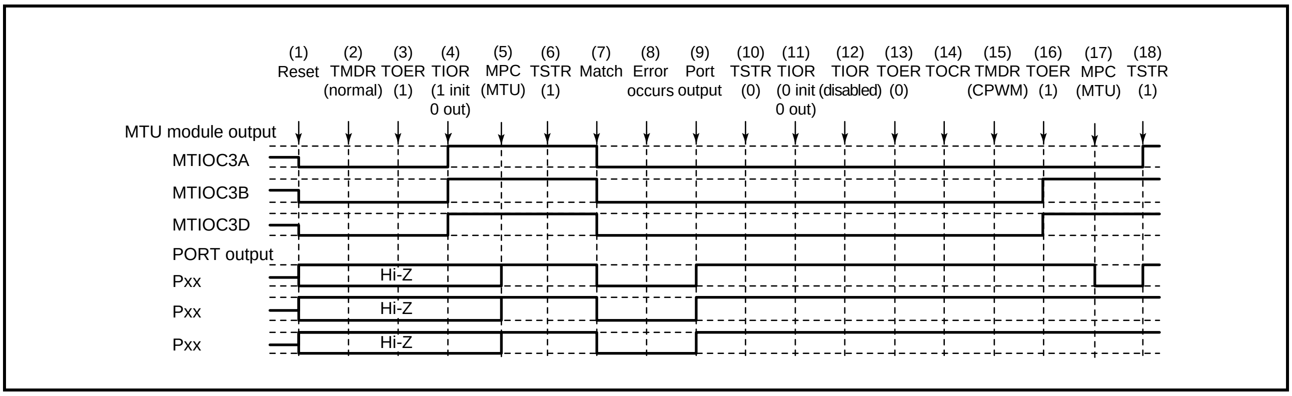


Figure 20.147 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 20.143.

(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 TOERA.

(14) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.

(15) Set complementary PWM mode.

(16) Enable output in MTU3 and MTU4 with TOERA.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting TSTRA.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.148 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

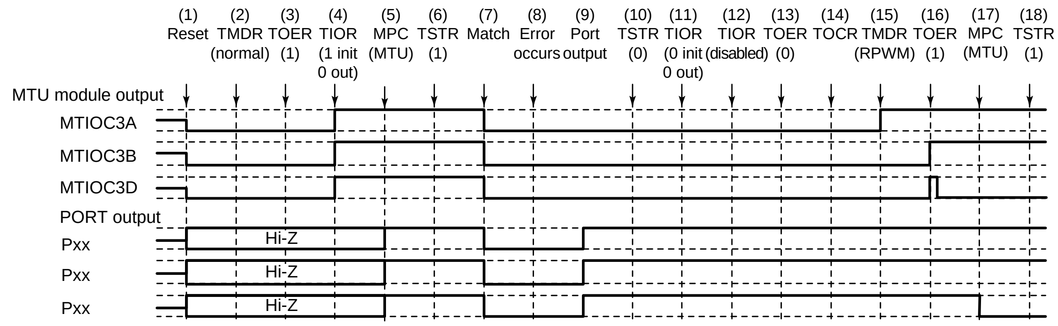


Figure 20.148 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 20.143.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.

(15) Set reset-synchronized PWM mode.

(16) Enable output in channels 3 and 4 with TOERA.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting TSTRA.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 20.149 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

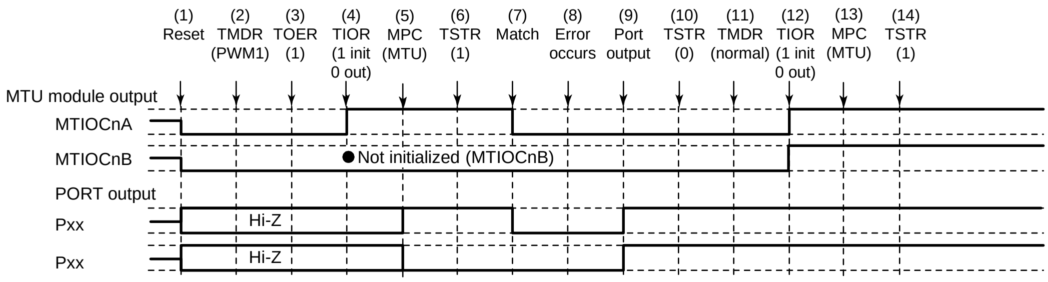


Figure 20.149 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(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 TOERA 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) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting TSTRA.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting TSTRA.
(11) Set normal mode.
(12) Initialize the pins with TIOR.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 20.150 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

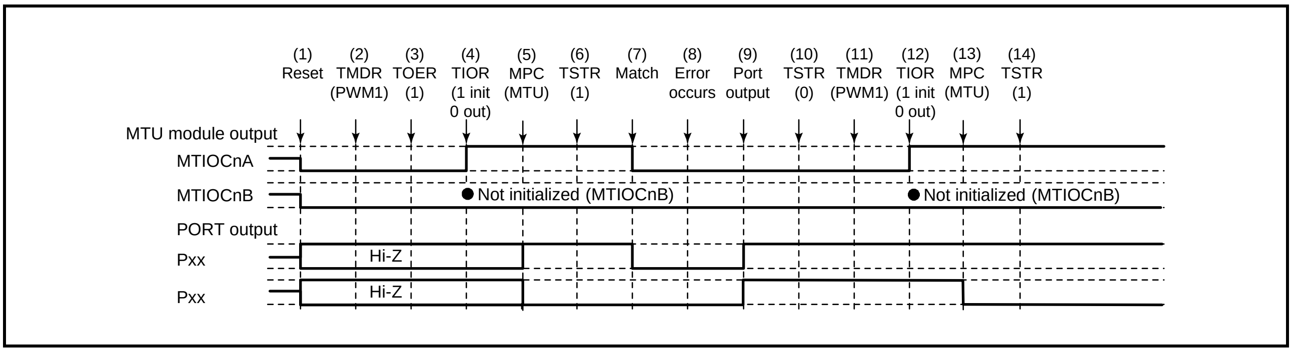


Figure 20.150 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 20.149.

(11) This step is not necessary when restarting in PWM mode 1.

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting TSTRA.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 20.151 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

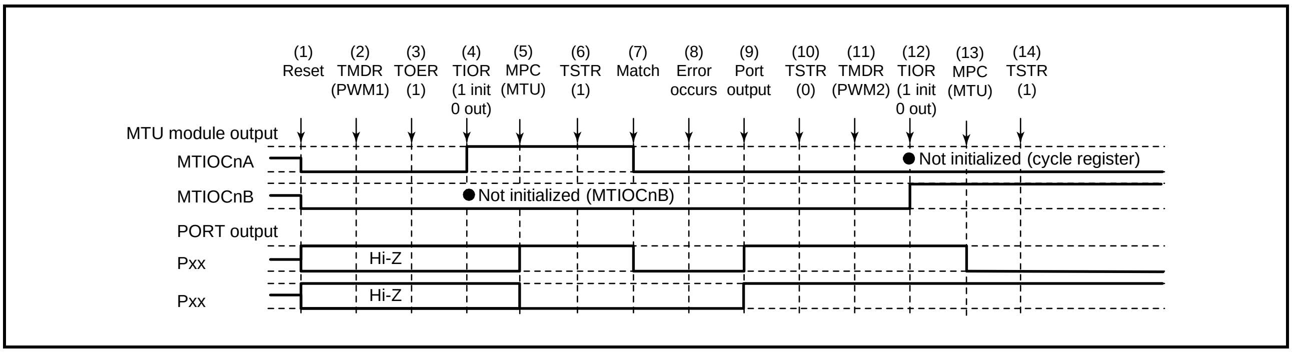


Figure 20.151 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 20.149.

(11) Set PWM mode 2.

(12) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting TSTRA.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOERA setting is not necessary.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 20.152 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

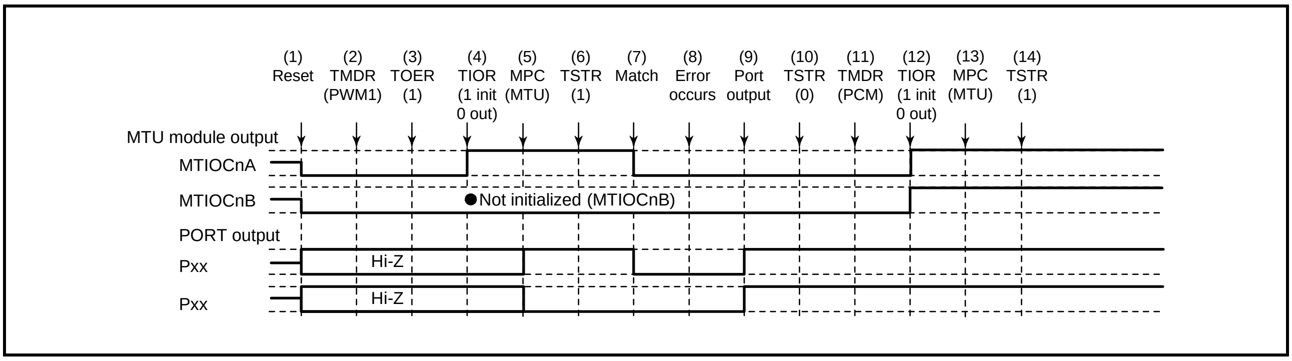


Figure 20.152 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 20.149.
(11) Set the phase counting mode.
(12) Initialize the pins with TIOR.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOERA setting is not necessary.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 20.153 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

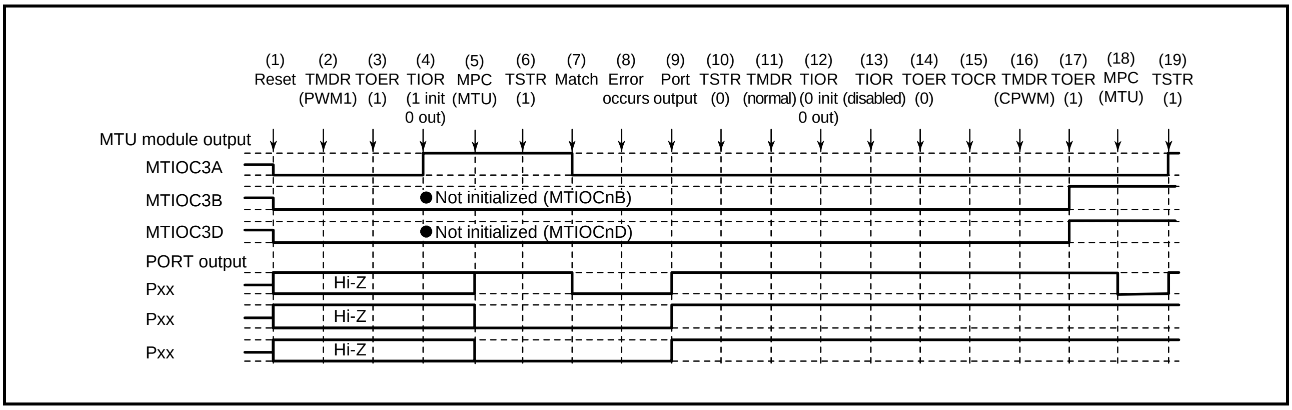


Figure 20.153 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 20.149.

(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 TOERA.

(15) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.

(16) Set complementary PWM mode.

(17) Enable output in MTU3 and MTU4 with TOERA.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting TSTRA.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.154 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

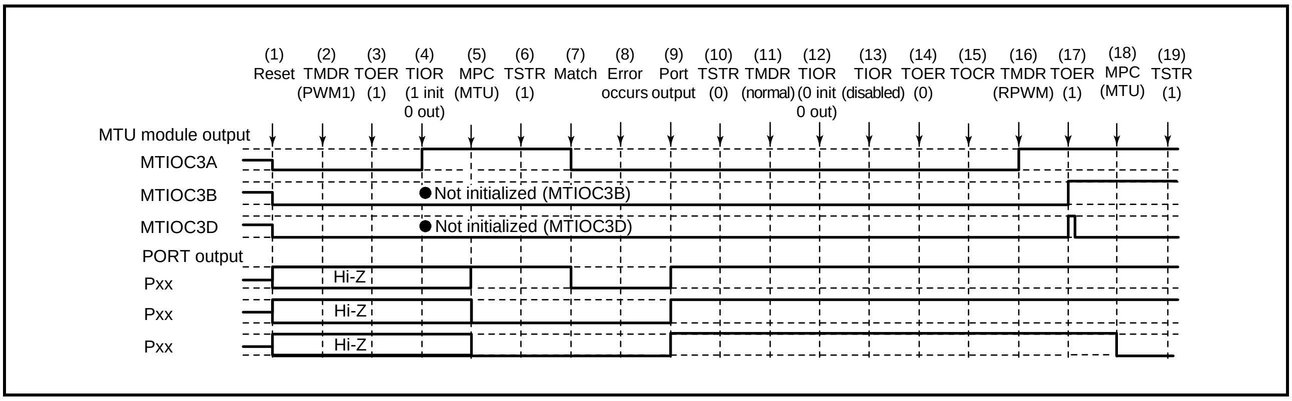


Figure 20.154 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 20.153.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 with TOERA.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting TSTRA.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 20.155 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

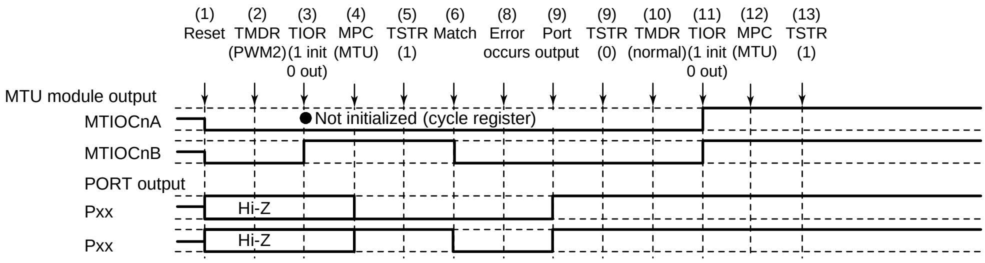


Figure 20.155 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(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) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting TSTRA.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting TSTRA.
(10) Set normal mode.
(11) Initialize the pins with TIOR.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting TSTRA.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 20.156 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

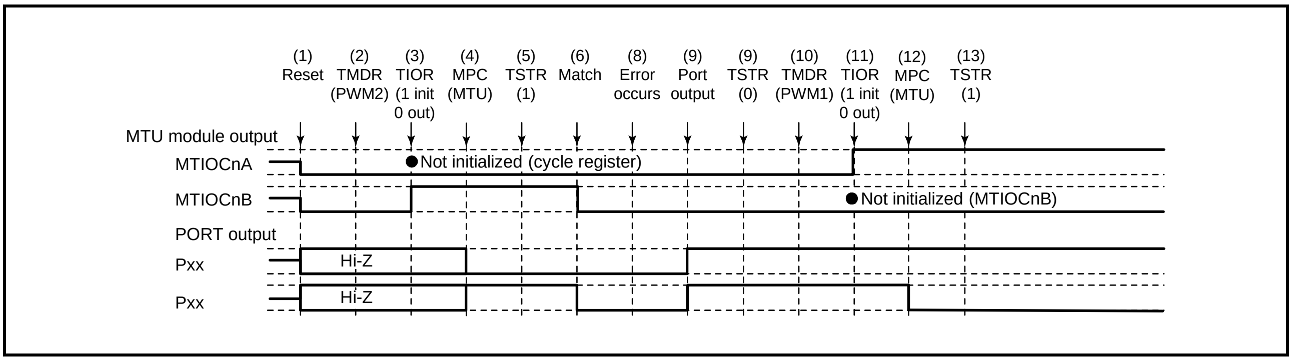


Figure 20.156 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 20.155.

(10) Set PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(13) Restart operation by setting TSTRA.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 20.157 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

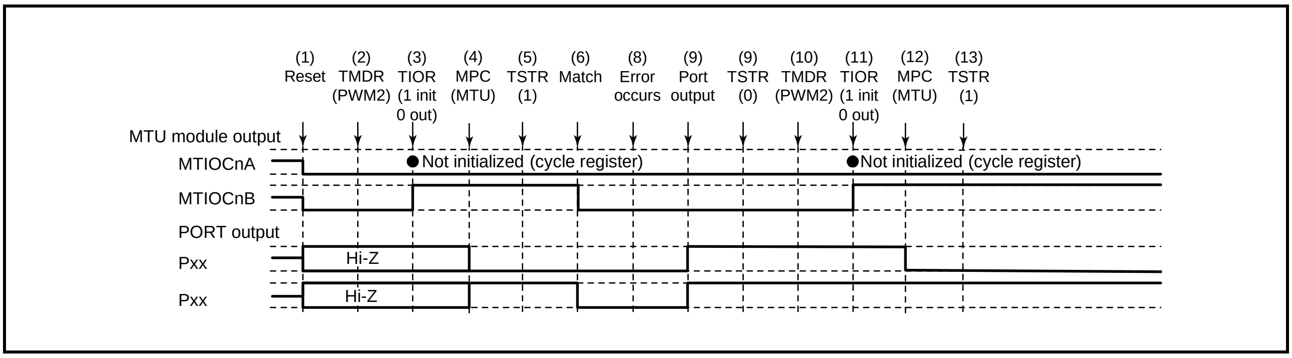


Figure 20.157 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 20.155.

(10) This step is not necessary when restarting in PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(13) Restart operation by setting TSTRA.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 20.158 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

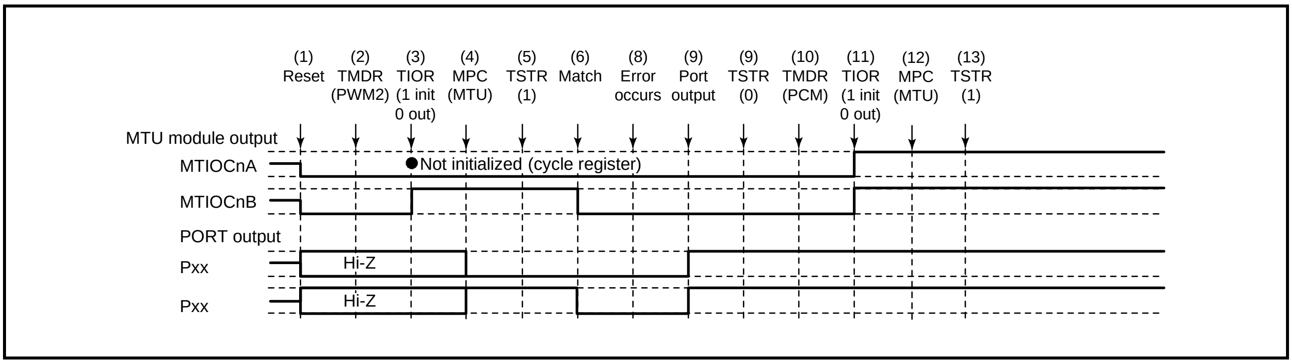


Figure 20.158 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 20.155.

(10) Set the phase counting mode.

(11) Initialize the pins with TIOR.

(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(13) Restart operation by setting TSTRA.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 20.159 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

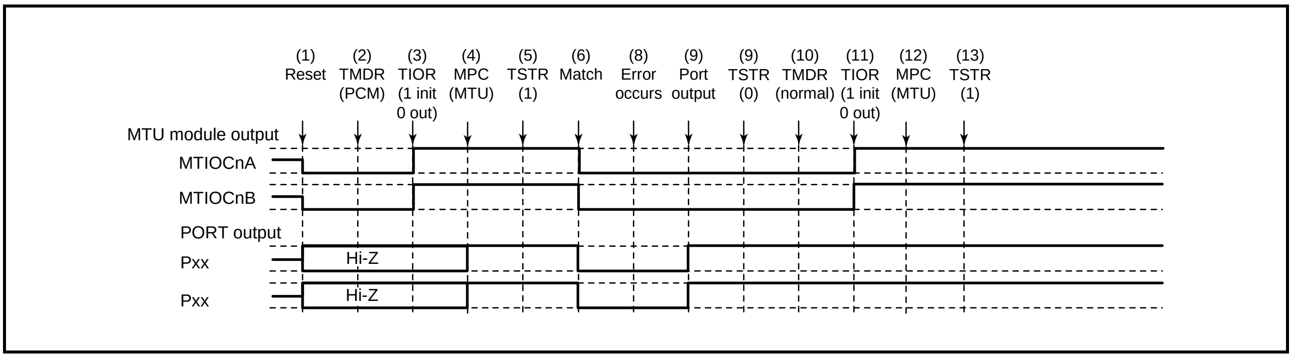


Figure 20.159 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(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) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting TSTRA.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting TSTRA.
(10) Set normal mode.
(11) Initialize the pins with TIOR.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting TSTRA.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 20.160 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

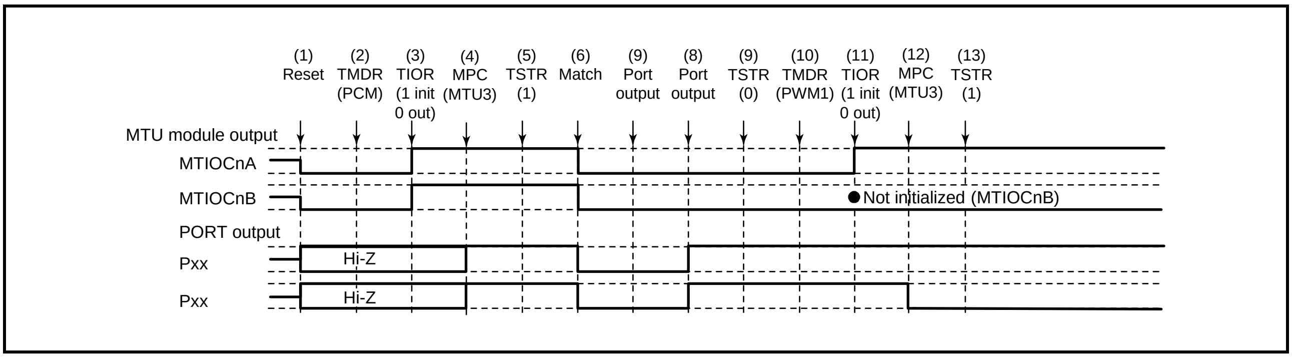


Figure 20.160 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 20.159.

(10) Set PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(13) Restart operation by setting TSTRA.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 20.161 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

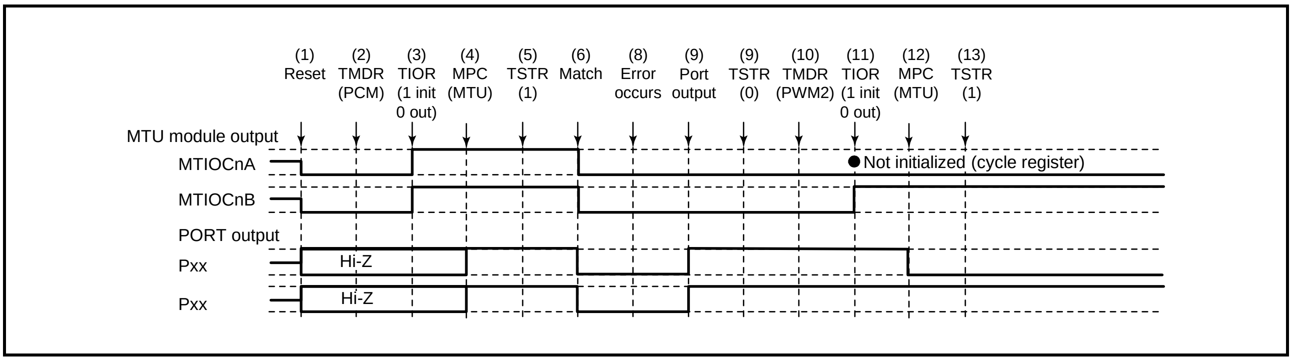


Figure 20.161 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 20.159.
(10) Set PWM mode 2.
(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting TSTRA.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 20.162 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

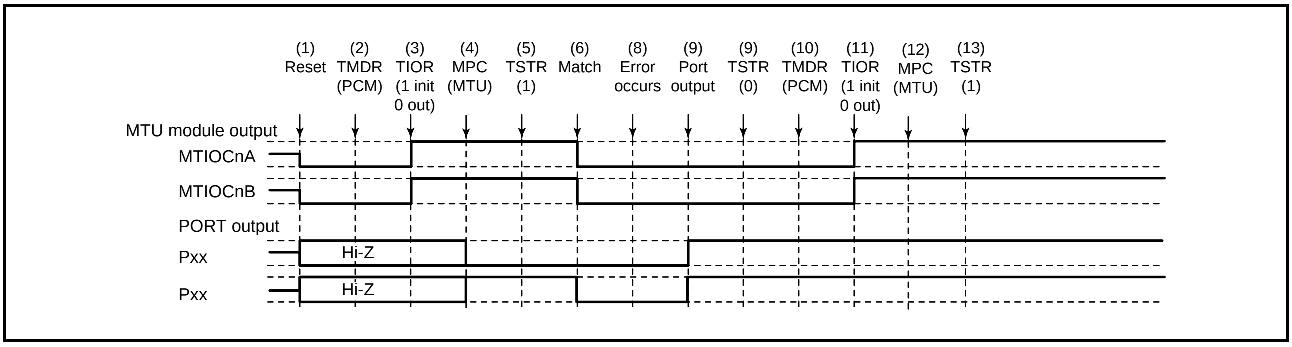


Figure 20.162 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 20.159.
(10) This step is not necessary when restarting in phase counting mode.
(11) Initialize the pins with TIOR.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting TSTRA.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 20.163 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

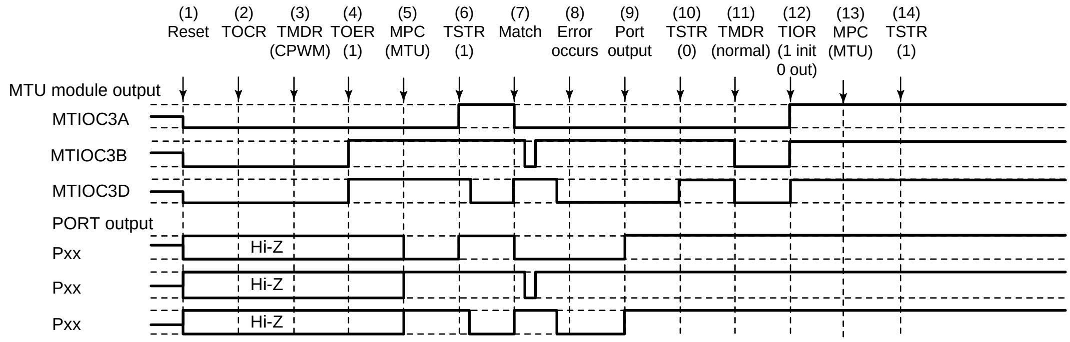


Figure 20.163 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 with TOERA.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting TSTRA.
(7) The complementary PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting TSTRA. (MTU output becomes the initial complementary PWM output value).
(11) Set normal mode (MTU output goes low).
(12) Initialize the pins with TIOR.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 20.164 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

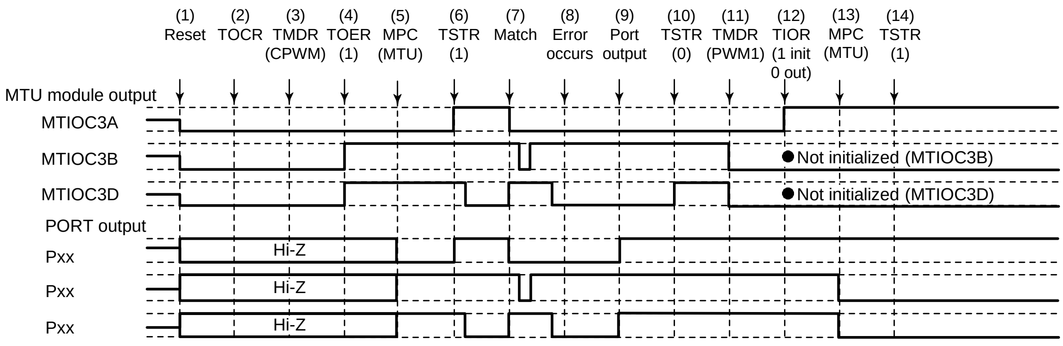


Figure 20.164 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 20.163.

(11) Set PWM mode 1 (MTU output goes low).

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting TSTRA.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.165 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).

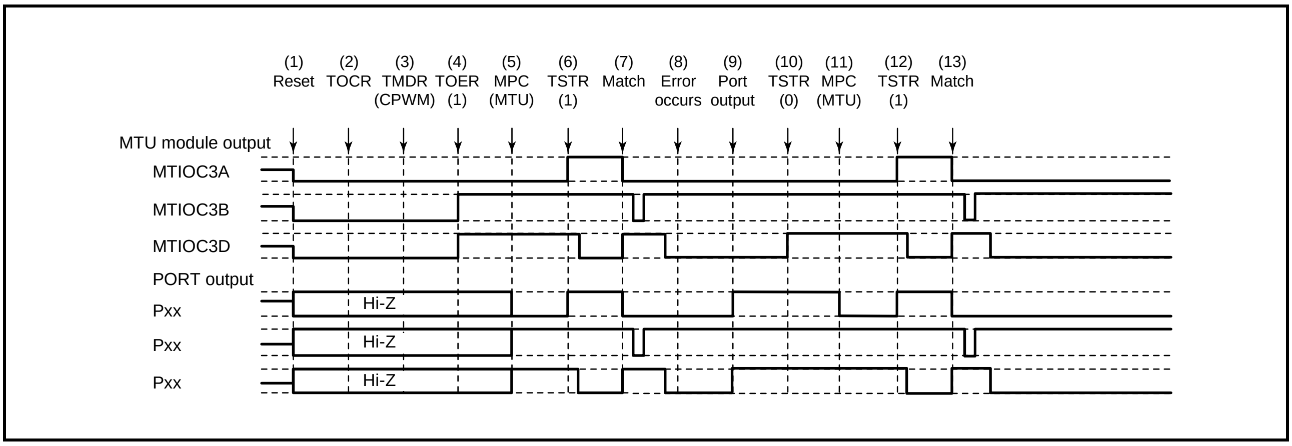


Figure 20.165 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 20.163.

(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(12) Restart operation by setting TSTRA.

(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 20.166 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 ratio settings).

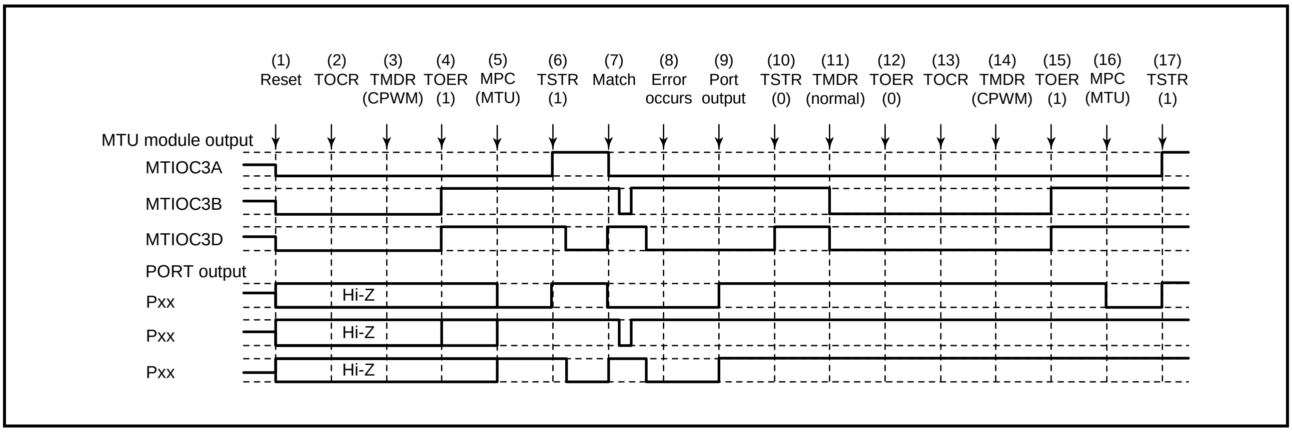


Figure 20.166 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 20.163.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with TOERA.
(13) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 with TOERA.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting TSTRA.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.167 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

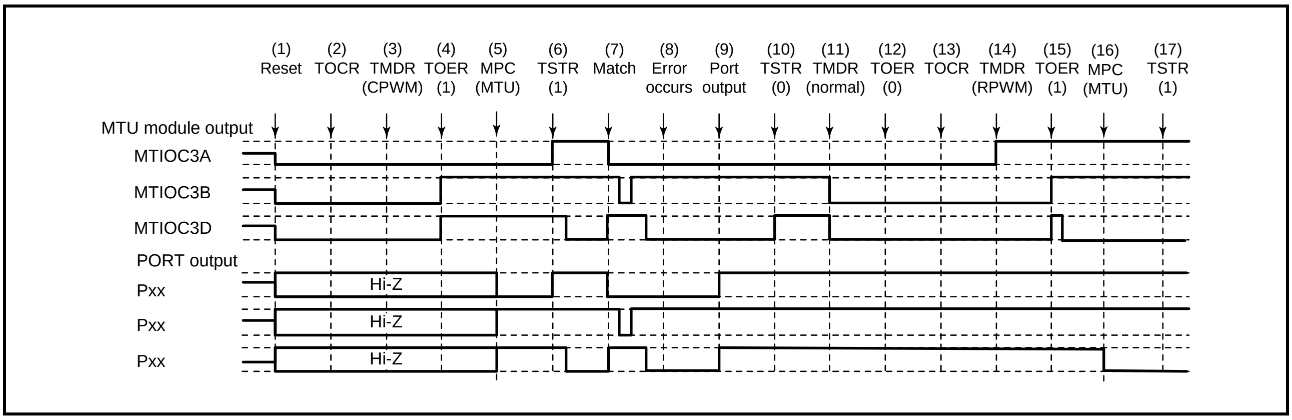


Figure 20.167 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (9) are the same as in Figure 20.163.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with TOERA.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 with TOERA.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting TSTRA.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 20.168 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

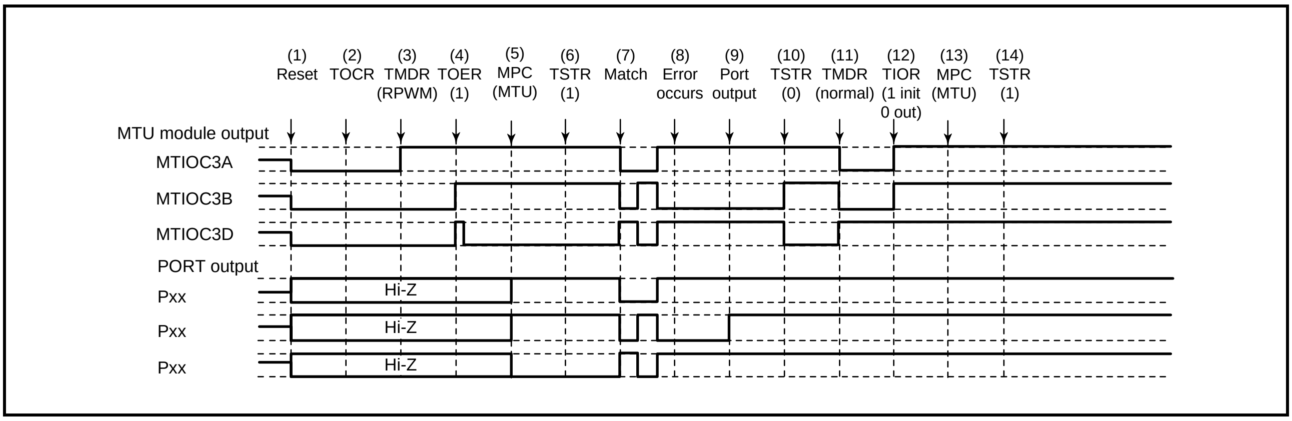


Figure 20.168 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(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 TOCR1A and TOCR2A.
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 with TOERA.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting TSTRA.
(7) The reset-synchronized PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting TSTRA. (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) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting TSTRA.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 20.169 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

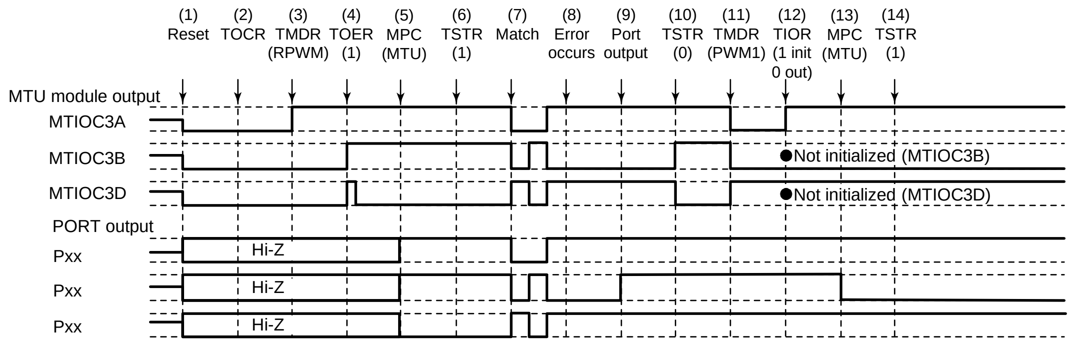


Figure 20.169 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 20.168.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting TSTRA.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.170 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

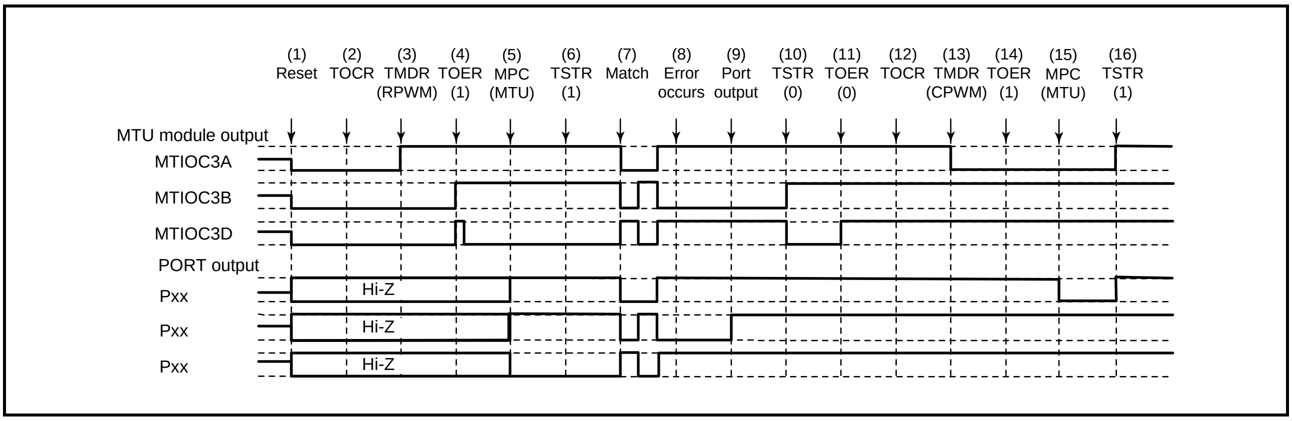


Figure 20.170 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 20.168.
(11) Disable output in MTU3 and MTU4 with TOERA.
(12) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A.
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 with TOERA.
(15) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(16) Restart operation by setting TSTRA.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.171 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.

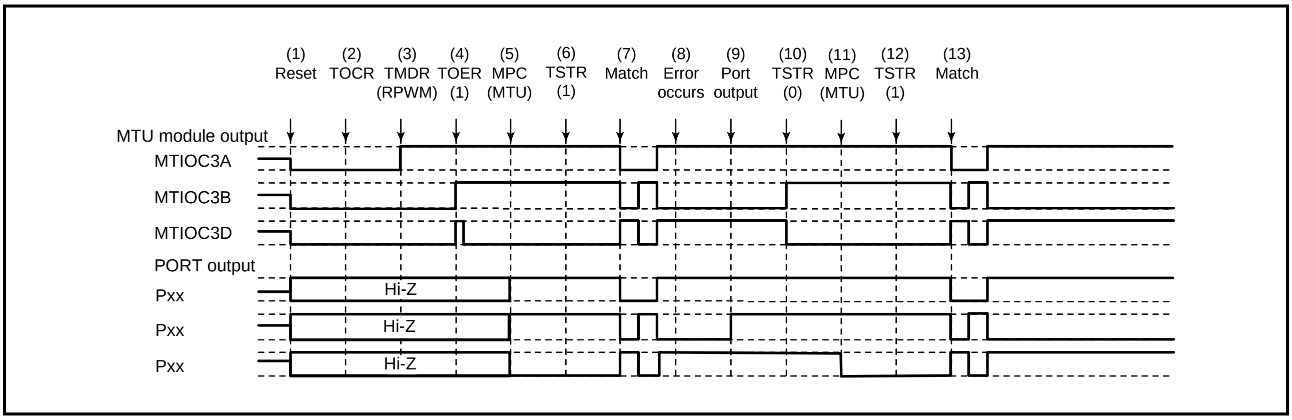


Figure 20.171 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 20.168.

(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(12) Restart operation by setting TSTRA.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

21. Port Output Enable 3 (POE3b)

The port output enable 3 (POE3b) register can be used to place output pins for the MTU in the high-impedance state under various conditions.

In this section, “PCLK” is used to refer to PCLKB.

21.1 Overview

Table 21.1 lists the specifications of the POE, and Figure 21.1 shows a block diagram of the POE.

Table 21.1 POE Specifications

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Target pins to be placed in the high-impedance state</td><td>• MTU output pins
MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D)
MTU3 pins (MTIOC3B, MTIOC3D)
MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D)</td></tr>
<tr><td>Conditions for the high-impedance state</td><td>• Setting pins as inputs: Setting the POE0#, POE8#, and POE10# pins as inputs.
• Short-circuits between output pins: A match (short circuit) between the output signal levels at the active level over one or more cycles on the following combination of pins
<table>
<tr><th></th><th>MTU Complementary PWM Output Pins</th></tr>
<tr><td>1</td><td>MTIOC3B and MTIOC3D</td></tr>
<tr><td>2</td><td>MTIOC4A and MTIOC4C</td></tr>
<tr><td>3</td><td>MTIOC4B and MTIOC4D</td></tr>
</table>
• Register setting for high-impedance being made
• Detection that the clock generation circuit had stopped oscillating
• Comparator detection in the comparator (CMPC)</td></tr>
<tr><td>Function</td><td>• Each of the POE0#, POE8#, and POE10# input pins can be set for falling edge, PCLK/8 × 16, PCLK/16 × 16, or PCLK/128 × 16 low-level sampling.
• Pins for the MTU complementary PWM output and MTU0 pins can be placed in high-impedance state by POE0#, POE8#, and POE10# pins falling-edge or low-level sampling.
• Pins for the MTU complementary PWM output and MTU0 pins can be placed in high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generator.
• Pins for the MTU complementary PWM output can be placed in high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more.
• Pins for the MTU complementary PWM output and MTU0 pins can be placed in the high-impedance state in response to comparator detection in the comparator (CMPC).
• Pins for the MTU complementary PWM output and MTU0 pins can be placed in the high-impedance state by modifying the settings of the POE registers.
• Interrupts can be generated by input-level sampling or output-level comparison results.</td></tr>
</table>

The POE has input-level detection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 21.1.

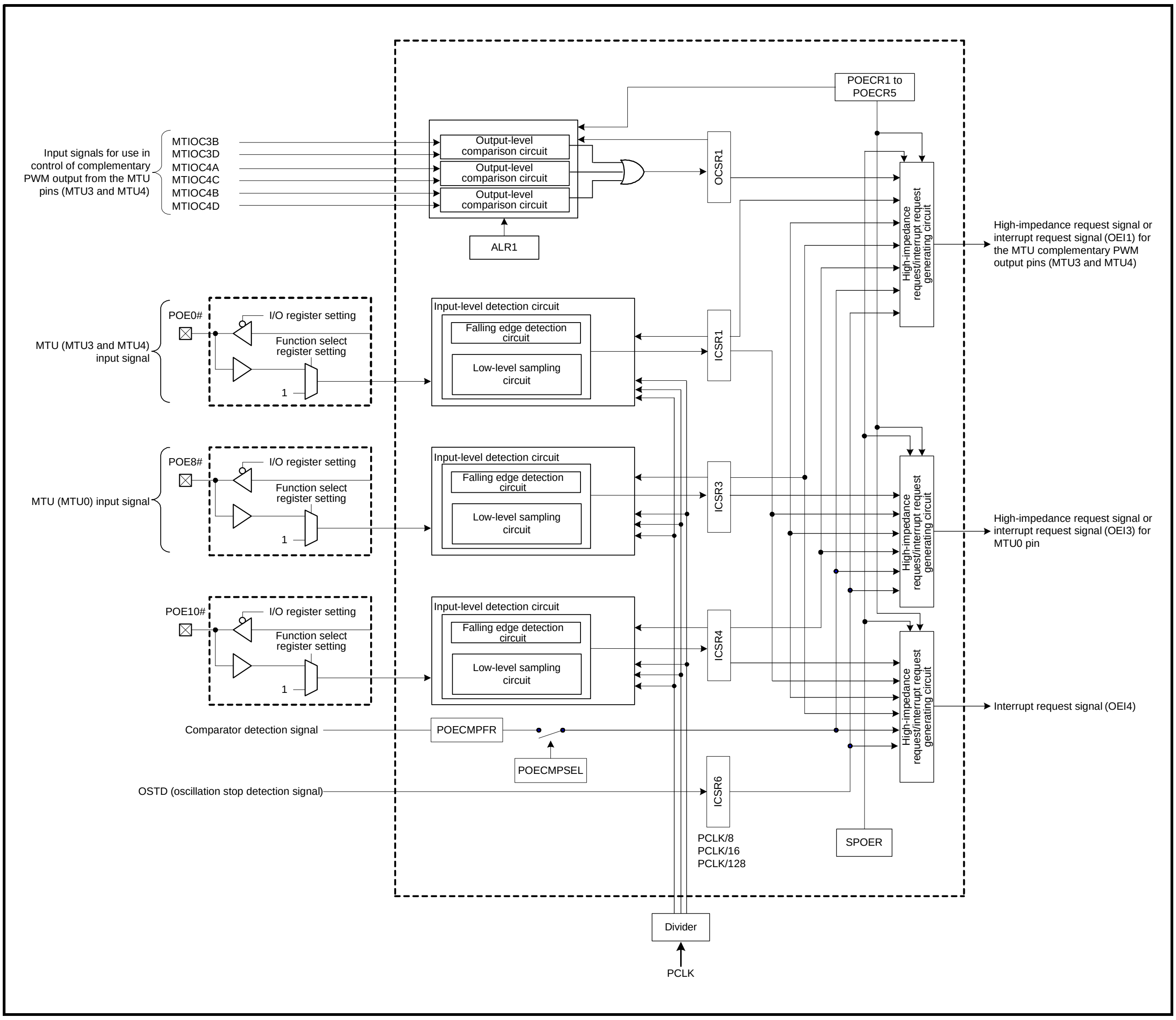


Figure 21.1 POE Block Diagram

Table 21.2 shows I/O pins to be used by the POE.

Table 21.2 POE I/O Pins

Pin Name	I/O	Description
POE0#	Input	Request signal to place the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state, and in accordance with register settings, is also capable of placing the MTU0 pins in the high-impedance state.
POE8#	Input	Request signal to place the pins for MTU0 in high-impedance state, and in accordance with register settings, is also capable of placing the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state.
POE10#	Input	In accordance with register settings, is capable of placing the MTU3 and MTU4 pins for MTU complementary PWM output, and MTU0 pins in high-impedance state.

Table 21.3 shows output-level comparisons with pin combinations.

Table 21.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU3 and MTU4 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level (low level when the MTUn.TOCR1A.OLSP bit is 0 with the MTUn.TOCR1A.TOCS bit cleared to 0 or high level when the OLSP bit is 1, or low level when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2A register of MTUn are 0 with the MTUn.TOCR1A.TOCS bit set to 1 and high level when these bits are 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison and high-impedance control can be selected by registers of POE.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

21.2 Register Descriptions

The POE registers are initialized by a reset.

21.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 C4C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	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# pin input. 0 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# pin 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	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE0F	POE0 Flag	0: Indicates that 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
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.

ICSR1 register selects the input modes for the POE0# pin, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] Bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1
  When low-level sampling is set by the POE0M[1:0] bits, the high level needs to be input to the POE0# pin to write 0 to this flag.
  For details, refer to section 21.3.7, Release from High-Impedance State.

21.2.2 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 C4C8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	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# pin input. 0 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# pin 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	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	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.

ICSR3 register selects the input mode for the POE8# pin, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] Bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.
[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

[Clearing condition]

- By writing 0 to POE8F after reading POE8F = 1
  When low-level sampling is set by the POE8M[1:0] bits, the high level needs to be input to the POE8# pin to write 0 to this flag.
  For details, refer to section 21.3.7, Release from High-Impedance State.

21.2.3 Input Level Control/Status Register 4 (ICSR4)

Address(es): 0008 C4D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	POE10M[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	POE10M[1:0]	POE10 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE10# pin input. 0 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE10# pin 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	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE10E	POE10 High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE10F	POE10 Flag	0: Indicates that a high-impedance request has not been input to the POE10# pin. 1: Indicates that a high-impedance request has been input to the POE10# 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.

ICSR4 register selects the POE10# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE10M[1:0] Bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag is set to 1.

POE10E Bit (POE10 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE10F flag is set to 1.

POE10F Flag (POE10 Flag)

This flag indicates that a request for the high-impedance state has been input to the POE10# pin.
[Setting condition]

- When the input set by POE10M[1:0] occurs at the POE10# pin

[Clearing condition]

- By writing 0 to POE10F after reading POE10F = 1
  When low-level sampling is set by the POE10M[1:0] bits, the high level needs to be input to the POE10# pin to write 0 to this flag.
  For details, refer to section 21.3.7, Release from High-Impedance State.

21.2.4 Input Level Control/Status Register 6 (ICSR6)

Address(es): 0008 C4DCh

	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 MTU complementary PWM output pins, or MTU0 pins in high-impedance state. 1: Places the MTU complementary PWM output pins, and MTU0 pins in high-impedance state.	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: Indicates that a oscillation stop high-impedance request has not been generated. 1: Indicates that a oscillation stop high-impedance request has been generated.	R/W*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

ICSR6 register controls the oscillation stop high-impedance and indicates status.

OSTSTE Bit (OSTST High-Impedance Enable)

This bit enables/disables the MTU complementary PWM output pins and MTU0 pins to be placed in the high-impedance state when oscillation stop is detected.

OSTSTF Flag (OSTST High-Impedance Flag)

This flag indicates that a oscillation stop high-impedance request has been generated.
When oscillation stop is detected, this flag is set to 1. To clear this flag, wait for at least 10 cycles of PCLKB after this flag becomes 1 and write 0 to this flag while the OSTDSR.OSTDF flag is 0. Writing 0 to this flag while the OSTDSR.OSTDF flag is 1 cannot clear this flag. After clearing this flag, confirm that the flag has actually been modified to 0.

[Setting condition]

- When oscillation stop is detected

[Clearing condition]

- By writing 0 to OSTSTF after reading OSTSTF = 1

21.2.5 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 C4C2h

	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: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	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.

OCSR1 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables 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 pins in high-impedance state 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 MTU3 and MTU4 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1
  To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins.
  For details, refer to section 21.3.7, Release from High-Impedance State.

21.2.6 Active Level Setting Register 1 (ALR1)

Address(es): 0008 C4DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	MTIOC3B Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	MTIOC3D Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	MTIOC4A Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	MTIOC4C Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	MTIOC4B Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	MTIOC4D Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

ALR1 register specifies the active levels of the MTU outputs for detection of short circuits of those outputs as reflected in OCSR1.

OLSG0A Bit (MTIOC3B Active Level Setting)

This bit sets the active level of the MTIOC3B output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG0B Bit (MTIOC3D Active Level Setting)

This bit sets the active level of the MTIOC3D output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1A Bit (MTIOC4A Active Level Setting)

This bit sets the active level of the MTIOC4A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1B Bit (MTIOC4C Active Level Setting)

This bit sets the active level of the MTIOC4C output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2A Bit (MTIOC4B Active Level Setting)

This bit sets the active level of the MTIOC4B output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2B Bit (MTIOC4D Active Level Setting)

This bit sets the active level of the MTIOC4D output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSEN Bit (Active Level Setting Enable)

This bit enables or disables of the active-level settings in the OLSGnm bits (n = 0 to 2; m = A, B). Clearing the OLSEN bit to 0 disables the OLSGnm bits, in which case the active levels of the MTU output are determined by the MTU.TOCR1m and MTU.TOCR2m registers. Setting the OLSEN bit to 1 enables the OLSGnm bits, in which case the active levels of the MTU output are as selected by the OLSGnm bits in this register.

21.2.7 Software Port Output Enable Register (SPOER)

Address(es): 0008 C4CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTUC H0HIZ	—	MTUC H34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTUCH34HIZ	MTU3 and MTU4 Output High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W
b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	MTUCH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPOER register controls high-impedance state of the pins.

MTUCH34HIZ 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, and MTIOC4D) in high-impedance state.

[Setting condition]

- By writing 1 to MTUCH34HIZ

[Clearing conditions]

- Reset
- By writing 0 to MTUCH34HIZ after reading MTUCH34HIZ = 1

MTUCH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to place the MTU pins in high-impedance state.

[Setting condition]

- By writing 1 to MTUCH0HIZ

[Clearing conditions]

- Reset
- By writing 0 to MTUCH0HIZ after reading MTUCH0HIZ = 1

21.2.8 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 C4CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU0C 1ZE	MTU0B 2ZE	MTU0B 1ZE	MTU0A 1ZE	MTU0D ZE	MTU0C ZE	MTU0B ZE	MTU0A ZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AZE	MTIOC0A PB3 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b1	MTU0BZE	MTIOC0B PB2 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b2	MTU0CZE	MTIOC0C PB1 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b3	MTU0DZE	MTIOC0D PB0 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b4	MTU0A1ZE	MTIOC0A P31 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b5	MTU0B1ZE	MTIOC0B P30 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b6	MTU0B2ZE	MTIOC0B P93 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1
b7	MTU0C1ZE	MTIOC0C P94 Pin High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*1

Note 1. Can be modified only once after a reset.

POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE Bit (MTIOC0A PB3 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0A output of PB3 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0BZE Bit (MTIOC0B PB2 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output of PB2 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0CZE Bit (MTIOC0C PB1 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0C output of PB1 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0DZE Bit (MTIOC0D PB0 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0D output of PB0 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0A1ZE Bit (MTIOC0A P31 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0A output of P31 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0B1ZE Bit (MTIOC0B P30 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output of P30 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0B2ZE Bit (MTIOC0B P93 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output of P93 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

MTU0C1ZE Bit (MTIOC0C P94 Pin High-Impedance Enable)

This bit specifies whether to place the MTIOC0C output of P94 in high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in POECR5, the ICSRn.POEmF flag (n = 1, 4; m = 0, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag, is set to 1.

21.2.9 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 C4CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 1.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BDZE	MTIOC4B/4D High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W*1
b9	MTU4ACZE	MTIOC4A/4C High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W*1
b10	MTU3BDZE	MTIOC3B/3D High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3 and MTU4 pins).

MTU4BDZE Bit (MTIOC4B/4D High-Impedance Enable)

This bit specifies whether to place the MTIOC4B output and MTIOC4D output in high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 3, 4; m = 8, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag is set to 1.

MTU4ACZE Bit (MTIOC4A/4C High-Impedance Enable)

This bit specifies whether to place the MTIOC4A output and MTIOC4C output in high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 3, 4; m = 8, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag is set to 1.

MTU3BDZE Bit (MTIOC3B/3D High-Impedance Enable)

This bit specifies whether to place the MTIOC3B output and MTIOC3D output in high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 3, 4; m = 8, 10), or POECMPFR.CnFLAG (n = 0 to 2) flag is set to 1.

21.2.10 Port Output Enable Control Register 4 (POECR4)

Address(es): 0008 C4D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IC4ADDMT34ZE	IC3ADDMT34ZE	—	—	CMADDMT34ZE
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT34ZE	MTU3 and MTU4 High-Impedance CFLAG Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance POE8F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b9 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

POECR4 register is used to extend the control conditions of the high-impedance state for the MTU3 and MTU4 pins for the MTU complementary PWM output.

CMADDMT34ZE Bit (MTU3 and MTU4 High-Impedance CFLAG Add)

Adds the POECMPFR.CnFLAG flag (n = 0 to 2) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).
However, when this flag is placed in the high-impedance, the OEIn interrupt (n = 1, 3, 4) will not occur.

IC3ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).

IC4ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).

21.2.11 Port Output Enable Control Register 5 (POECR5)

Address(es): 0008 C4D2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IC4ADD MT0ZE	—	—	IC1ADD MT0ZE	CMADD MT0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT0ZE	MTU0 High-Impedance CFLAG Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b1	IC1ADDMT0ZE	MTU0 High-Impedance POE0F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b2	—	Reserved	This bit is 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*1
b4	IC4ADDMT0ZE	MTU0 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b15 to b5	—	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.

POECR5 register is used to extend the control conditions of the high-impedance for the MTU0 pins.

CMADDMT0ZE Bit (MTU0 High-Impedance CFLAG Add)

Adds the POECMPFR.CnFLAG flag (n = 0 to 2) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).
However, when this flag is placed in the high-impedance, the OEIn interrupt (n = 1, 3, 4) will not occur.

IC1ADDMT0ZE Bit (MTU0 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE Bit (MTU0 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

21.2.12 Port Output Enable Comparator Detection Flag Register (POECMPFR)

Address(es): 0008 C4E6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C2FLAG	C1FLAG	C0FLAG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C0FLAG	Comparator Channel 0 Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b1	C1FLAG	Comparator Channel 1 Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b2	C2FLAG	Comparator Channel 2 Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The flag can only be set to 0 by writing 0 after reading 1.

CnFLAG Flag (Comparator Channel n Detection Flag) (n = 0 to 2)

This flag indicates whether each comparator output is detected or not detected.

[Setting condition]

- A change from low level to high level in the comparator output is detected.
  - When the comparator is set to non-inverted output, the input voltage changes from lower to higher than the reference voltage
  - When the comparator is set to inverted output, the input voltage changes from higher to lower than the reference voltage

[Clearing condition]

- By writing 0 to CnFLAG after reading CnFLAG = 1

21.2.13 Port Output Enable Comparator Request Select Register (POECMPSEL)

Address(es): 0008 C4E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	POEREQ2	POEREQ1	POEREQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POEREQ0	Comparator Channel 0 POE Request Enable	0: Disables POE request generation upon comparator detection. 1: Enables POE request generation upon comparator detection.	R/W*1
b1	POEREQ1	Comparator Channel 1 POE Request Enable	0: Disables POE request generation upon comparator detection. 1: Enables POE request generation upon comparator detection.	R/W*1
b2	POEREQ2	Comparator Channel 2 POE Request Enable	0: Disables POE request generation upon comparator detection. 1: Enables POE request generation upon comparator detection.	R/W*1
b15 to b3	—	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.

POECMPSEL register sets a comparator detection flag to use as a POE source.

POEREQn Bit (Comparator Channel n POE Request Enable) (n = 0 to 2)

This bit disables or enables POE request generation in response to each comparator output detection. A POE request is generated when one of the comparator outputs is detected.

21.3 Operation

The following shows the target pins and conditions for high-impedance control.

(1) MTU3 pins (MTIOC3B, MTIOC3D)

When one of the following conditions is satisfied while the POECR2.MTU3BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins
  When the OCSR1.OCF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(2) MTU4 pins (MTIOC4A, MTIOC4C)

When one of the following conditions is satisfied while the POECR2.MTU4ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins
  When the OCSR1.OCF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop

When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(3) MTU4 pins (MTIOC4B, MTIOC4D)

When one of the following conditions is satisfied while the POECR2.MTU4BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins
  When the OCSR1.OCF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(4) MTU0 pin PB3 (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0AZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(5) MTU0 pin P31 (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0A1ZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(6) MTU0 pin PB2 (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0BZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(7) MTU0 pin P30 (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0B1ZE bit is 1, the pins are placed in high-impedance state.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5

When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.

- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(8) MTU0 pin P93 (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0B2ZE bit is 1, the pins are placed in high-impedance state.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(9) MTU0 pin PB1 (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0CZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.

When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(10) MTU0 pin P94 (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0C1ZE bit is 1, the pins are placed in high-impedance state.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(11) MTU0 pin PB0 (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0DZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ0 bit are 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ1 bit are 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit and the POECMPSEL.POEREQ2 bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

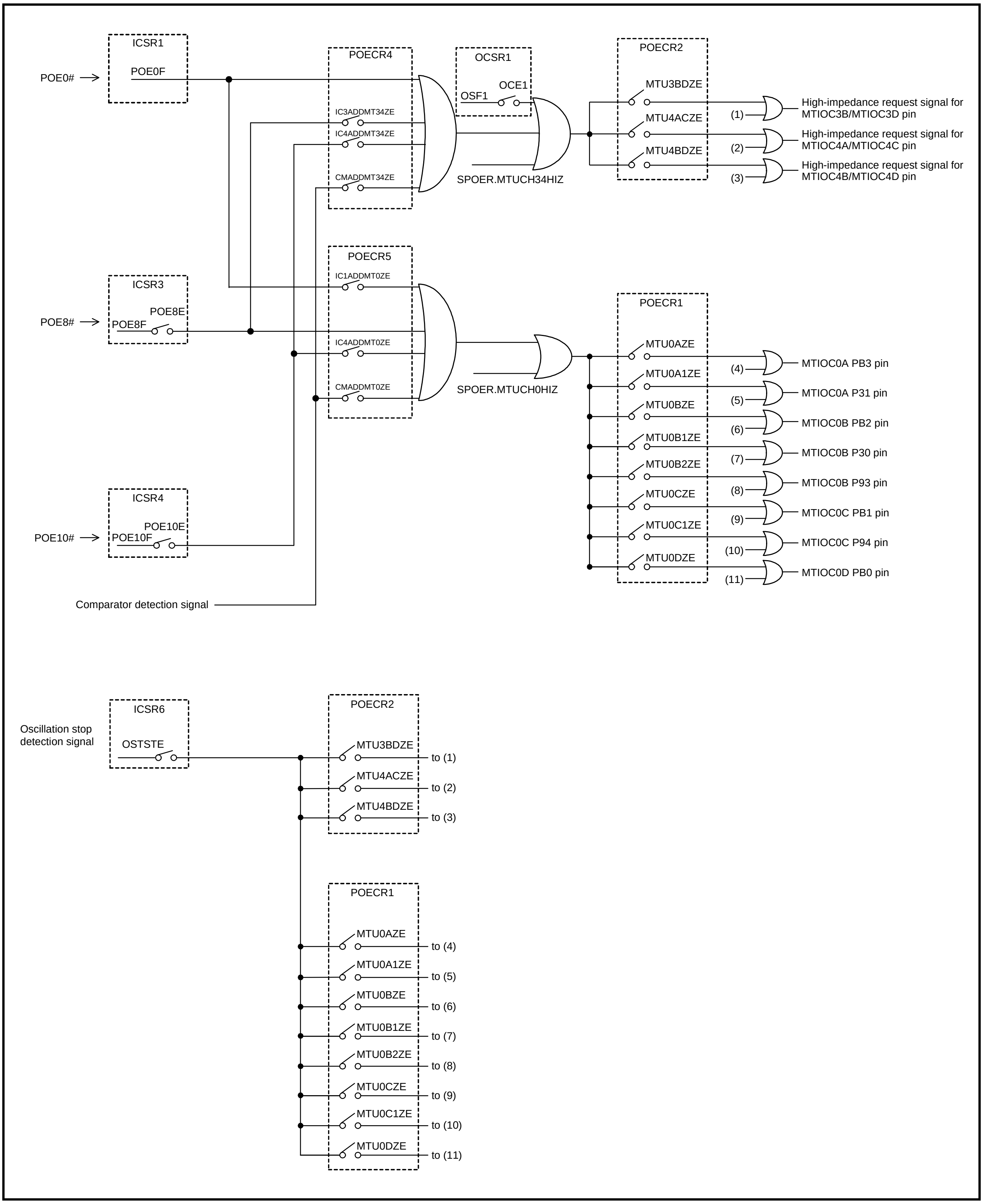


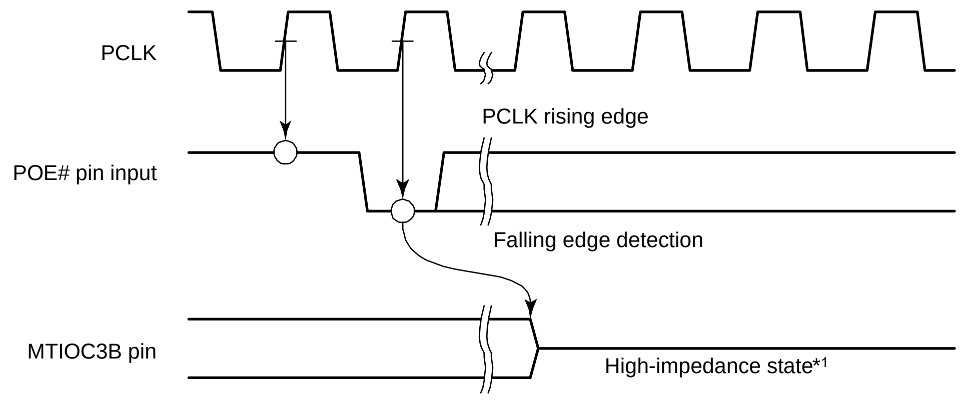
Figure 21.2 Target Pins and Conditions for High-Impedance Control

21.3.1 Input-Level Detection Operation

If the input conditions set by ICSR1 to ICSR4 occur on the POE0#, POE8#, and POE10# pins, the MTU3 and MTU4 pins for the MTU complementary PWM output, and MTU0 pins are placed in high-impedance state. Note however, that these pins are still placed in the high-impedance state even when the MTU functions are not selected for the pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE8#, and POE10# pins, the pins multiplexed with MTU complementary PWM output pins, and MTU0 pins are placed in high-impedance state.
The falling edge is detected after the level is sampled with PCLK. Input a low level for at least one PCLK clock to the POE0#, POE8#, and POE10# pins.
Figure 21.3 shows a sample timing after the level changes in input to the POE0#, POE8#, and POE10# pins until the respective pins enter high-impedance state.

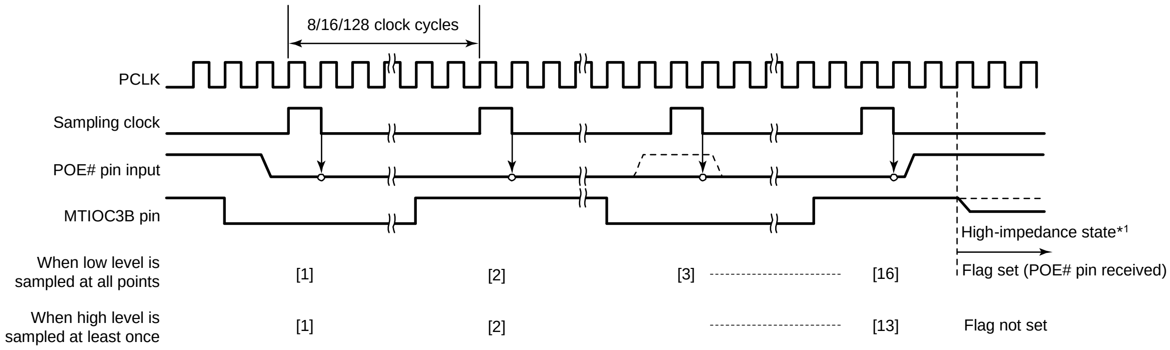


Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 21.3 Falling Edge Detection

(2) Low-Level Detection

Figure 21.4 shows the low-level detection operation. When 16 continuous low levels are sampled with the sampling clock selected by the ICSR1 to ICSR4 registers, the low level is recognized and the MTU complementary PWM output pins, and MTU0 pins are placed in high-impedance state. If even one high level is detected during this interval, the low level is not recognized.
The timing when pins for the MTU complementary PWM output, and MTU0 pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

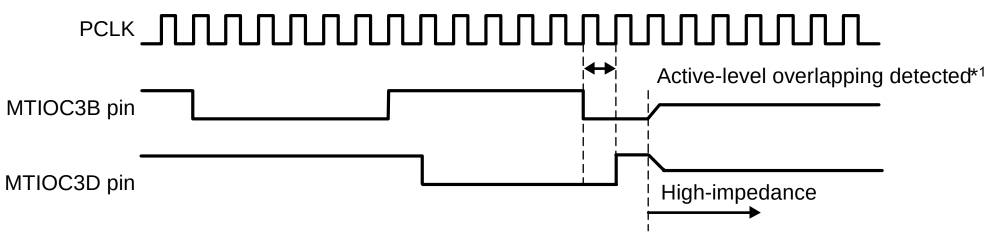


Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 21.4 Low-Level Detection Operation

21.3.2 Output-Level Compare Operation

Figure 21.5 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.



Note 1. When the active level of both MTIOC3B and MTIOC3D pins is set to low.

Figure 21.5 Output-Level Compare Operation

21.3.3 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU3, and MTU4) can be directly controlled by using the SPOER register.
For instance, setting the SPOER.MTUCH34HIZ bit to 1 places the MTU3 and MTU4 pins specified by the POECR2 register in the high-impedance state.
The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

21.3.4 High-Impedance Control through Detection of Oscillation Stop

When oscillation stop is detected by the oscillation stop detection function of the clock generator while the ICSR6.OSTSTE bit is 1, the following pins are placed in the high-impedance state:
The MTU complementary PWM output pins specified by the POECR2 register and the MTU0 pins specified by the POECR1 register.

21.3.5 High-Impedance Control through Detection of the Comparator

The MTU complementary PWM output pins and MTU0 pins can be placed in the high-impedance state in response to detection by the on-chip comparator in the comparator.
For instance, when the POECMPFR.CnFLAG (n = 0 to 2) flag is added to the high-impedance control conditions for the MTU3 and MTU4 pins by setting the POECR4.CMADDMT34ZE bit to 1, the MTU3 and MTU4 pins specified by the POECR2 register are placed in the high-impedance state on comparator detection.
The high-impedance state of other pins can be controlled by the POECR1 to POECR5 registers.

21.3.6 Additional Functions for Controlling High-Impedance States

High-impedance control conditions for the MTU complementary PWM output pins, and MTU0 pins can be added by setting the POECR4 and POECR5 registers.
For instance, the settings listed below can be added as high-impedance control conditions for the MTU3 and MTU4 pins.

- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE10# pin

The high-impedance state of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5.

21.3.7 Release from High-Impedance State

MTU pins which have entered high-impedance state due to input-level detection can be released from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR3.POE8F, and ICSR4.POE10F flags. However, note that when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR3.POE8M[1:0], and ICSR4.POE10M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE8#, and POE10# pins and is detected.
MTU pins which have entered high-impedance state due to output-level detection can be released from the state either by returning them to their initial state with a reset, or by setting the OCSR1.OSF1 flag to 0. However, note that just writing 0 to a flag is ignored (the flag is not set to 0); the flags can be cleared by writing 0 to it only after setting the inactive level to be output from the pin. The inactive level is output by setting the MTU and ALR1 registers.
MTU pins which have entered high-impedance state due to comparator detection can be released from the state either by returning them to their initial state with a reset or by setting the POECMPFR.CnFLAG (n = 0 to 2) flag to 0.

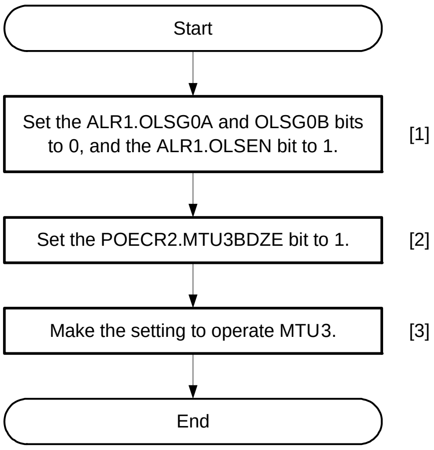
When setting the POECMPFR.CnFLAG flag to 0, be sure to confirm that the analog input signal that triggered comparator detection has returned to a normal value by performing A/D conversion and so on.
Note that the above comparator detection flag POECMPFR.CnFLAG is not set to 1 again in the following cases:
- This flag is cleared without confirmation that the analog input signal has returned to a normal value, and

1. the analog input signal remains above the reference voltage when the comparator is set to non-inverted output, or
2. the analog input signal remains below the reference voltage when the comparator is set to inverted output.

MTU pins which have entered high-impedance state due to oscillation stop detection can be released from the state either by returning them to their initial state with a reset or by setting the SYSTEM.OSTDSR.OSTDF flag to 0 to set the ICSR6.OSTSTF flag to 0.

21.4 POE Setting Procedure

Figure 21.6 shows the procedure for setting the POE. It illustrates an example of high-impedance control in response to comparison of the output levels on the MTU3 pins (MTIOC3B/MTIOC3D). In the figure, P71 is used as the MTIOC3B pin and P74 is used as the MTIOC3D pin.



Set the MTU3 pins (P71 and P74) for the peripheral function by the PMR register of the I/O port before making the following settings.

[1] Set the ALR1.OLSG0A and OLSG0B bits to 0 and the ALR1.OLSEN bit to 1 to set MTIOC3B and MTIOC3D pins as active low.

[2] Set the POECR2.MTU3BDZE bit to 1 to enable high-impedance control on the MTIOC3B and MTIOC3D pins.

[3] Make the setting to operate MTU3 pin.

After the above settings, if MTIOC3B (P71) and MTIOC3D (P74) pins are simultaneously at the low level, then the pins are placed in high-impedance state and the OCSR1.OSF1 flag is set to 1.
For the method for releasing high-impedance, refer to section 21.3.7, Release from High-Impedance State.

Figure 21.6 Procedure for Setting the POE

21.5 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 21.4 shows the interrupt sources and their conditions.

Table 21.4 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F and OSF1	When the ICSR1.POE0F flag is set to 1 while the ICSR1.PIE1 bit is 1 or when the OCSR1.OSF1 flag is set to 1 while the OCSR1.OIE1 bit is 1
OEI3	Output enable interrupt 3	POE8F	When the ICSR3.POE8F flag is set to 1 while the ICSR3.PIE3 bit is 1
OEI4	Output enable interrupt 4	POE10F	When the ICSR4.POE10F flag is set to 1 while the ICSR4.PIE4 bit is 1

21.6 Usage Notes

21.6.1 Transition to Low Power Consumption Mode

When the POE is used, do not make a transition to software standby mode. In this mode, the POE stops and thus the high-impedance state of pins cannot be controlled.

21.6.2 High-Impedance Control When the MTU is Not Selected

If high-impedance control for a pin having a multiplexed MTU pin function is enabled by setting the POECR1 and POECR2 registers and the high-impedance condition is satisfied, the pin is placed in the high-impedance state even if the MTU function is not selected for the pin on which it is multiplexed.
To avoid unintended high-impedance states, ensure that there are no differences between the settings for MTU pin selection in the PmnPFS registers of the MPC and for MTU pin selection in the pin select register of the POE.

21.6.3 When the POE is Not Used

The high-impedance of pins can be controlled using the POE after a reset. When the POE is not used, write 0 to the target bits in the POECR1 and POECR2 registers.

22. 8-Bit Timer (TMR)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a baud rate clock for the SCI.
In this section, “PCLK” is used to refer to PCLKB.

22.1 Overview

Table 22.1 lists the specifications of the TMR. Table 22.2 lists the TMR functions.
Figure 22.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 22.2 shows that of the 8-bit timer module (unit 1).

Table 22.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 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 counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits) • 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
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
A/D conversion start trigger of the A/D converter	Compare match A of TMR0 and TMR2
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*1
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, see section 25, Serial Communications Interface (SCIg).

Table 22.2 TMR Functions

Item		Unit 0			Unit 1		
Counter mode		8 Bits		16 Bits	8 Bits		16 Bits
Channel		TMR0	TMR1	TMR0 + TMR1	TMR2	TMR3	TMR2 + TMR3
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB TMRI0	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
Timer output	Low output	○	○	○	○	○	○
	High output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
DTC activation	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
A/D conversion start trigger of the A/D converter*1		○	—	○	○	—	○
SCI baud rate clock generation*2		○		—	○		—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

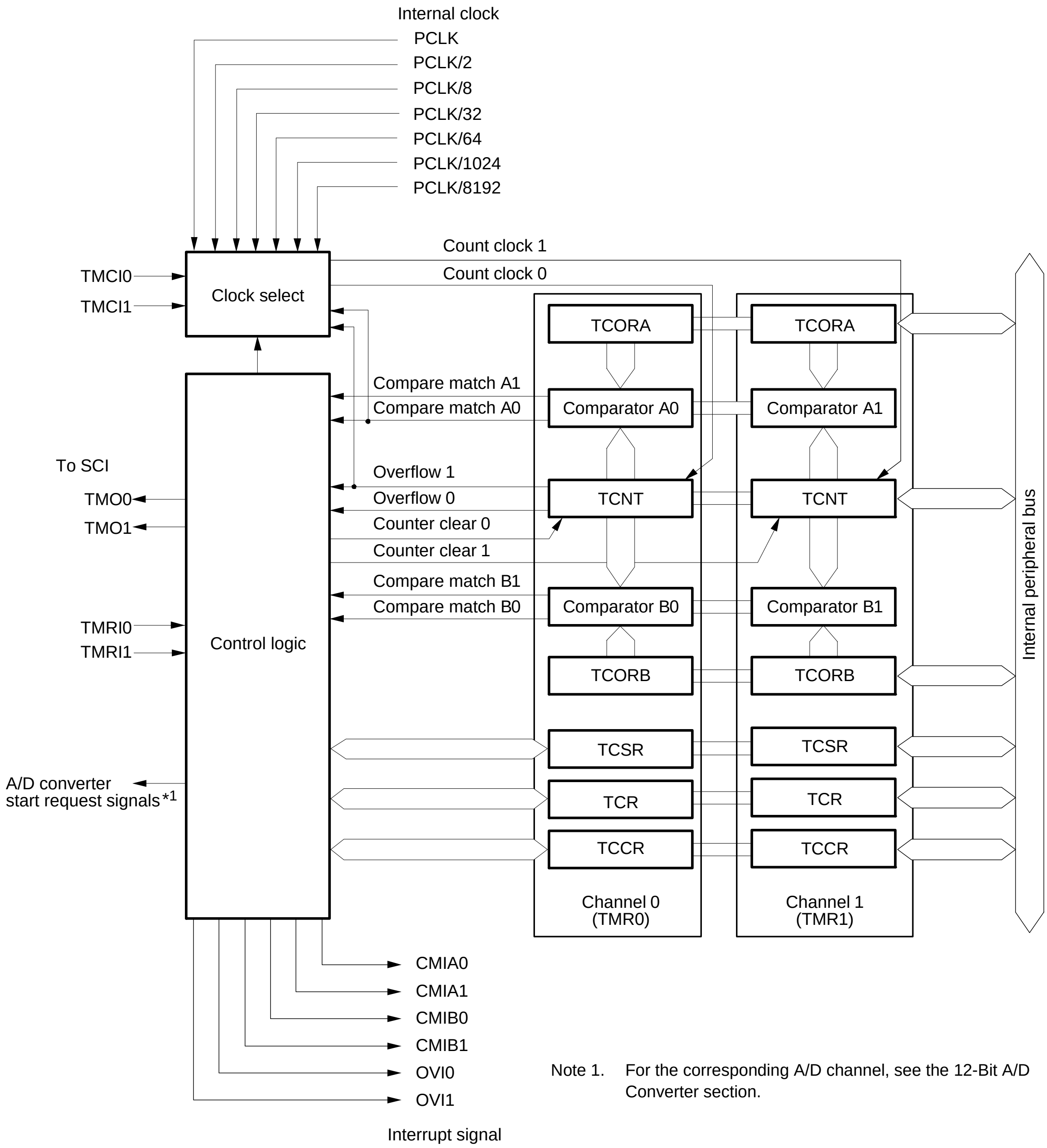
○: Possible

—: Impossible

Note 1. For details, see section 29, 12-Bit A/D Converter (S12ADE).

Note 2. For details, see section 25, Serial Communications Interface (SCIg).

Note 3. For details, see section 11, Low Power Consumption.



Note 1. For the corresponding A/D channel, see the 12-Bit A/D Converter section.

TCORA:	Time constant register A	TCSR:	Timer control/status register
TCNT:	Timer counter	TCR:	Timer control register
TCORB:	Time constant register B	TCCR:	Timer counter control register

Figure 22.1 Block Diagram of TMR (Unit 0)

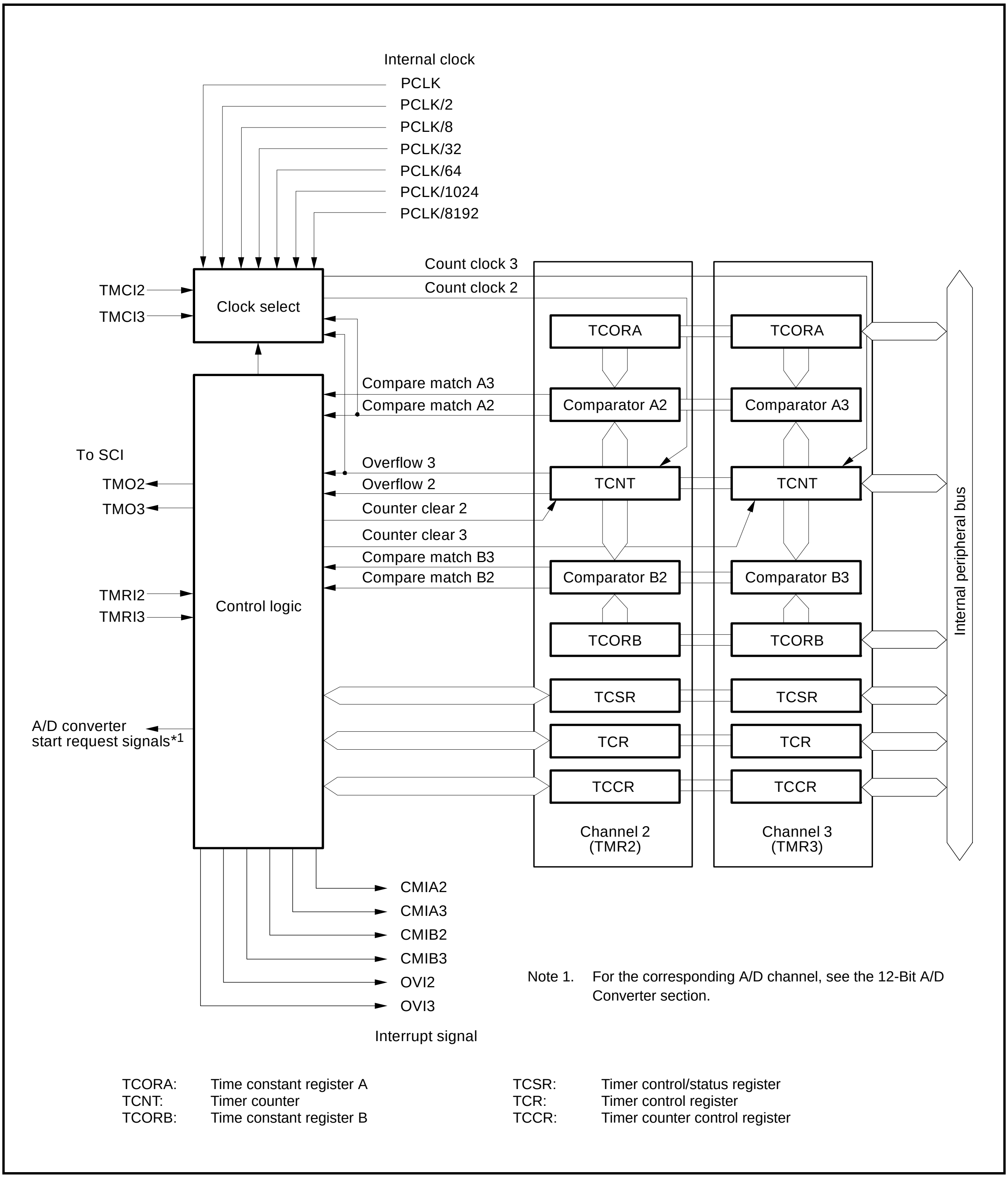


Figure 22.2 Block Diagram of TMR (Unit 1)

Table 22.3 lists the I/O pins of the TMR.

Table 22.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
		TMRI0	Input	Inputs external counter reset
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset

22.2 Register Descriptions

Table 22.4 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

22.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h

	TMR0.TCNT(TMR2.TCNT)								TMR1.TCNT(TMR3.TCNT)							
	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	0	0	0	0	0	0	0	0

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 count clock.

TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt (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 22.6, TMR Interrupt Sources.

22.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h

	TMR0.TCORA(TMR2.TCORA)								TMR1.TCORA(TMR3.TCORA)							
	b7	b6	b5	b4	b3	b2	b1	b0	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

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.

22.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h

	TMR0.TCORB(TMR2.TCORB)								TMR1.TCORB(TMR3.TCORB)							
	b7	b6	b5	b4	b3	b2	b1	b0	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

TCORB is an 8-bit readable/writable register.
TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.
The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEB bit.
However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

22.2.4 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 counter reset signal (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 counter reset signal, set the corresponding pin function. For details, refer to section 18, I/O Ports and section 19, Multi-Function Pin Controller (MPC).

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.

22.2.5 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 22.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 22.5.	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 counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 18, I/O Ports and section 19, Multi-Function Pin Controller (MPC).

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 22.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Table 22.5 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 count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, see section 18, I/O Ports and section 19, Multi-Function Pin Controller (MPC).

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 TCNT count clock is generated. Do not use this setting.

22.2.6 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
	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B *1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	ADTE	A/D Trigger Enable*2	0: A/D conversion start request in response to compare match A is disabled. 1: A/D conversion start request in response to compare match A is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When 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 pin 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.

Note 2. For the corresponding A/D channel, see section 29, 12-Bit A/D Converter (S12ADE).

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B *1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When 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 pin 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.

22.3 Operation

22.3.1 Pulse Output

Figure 22.3 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 pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

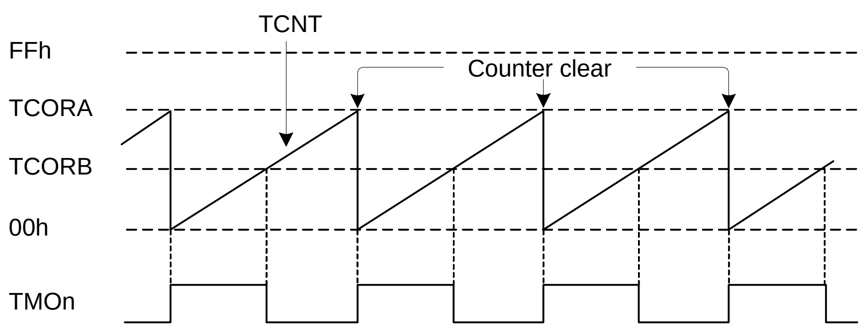


Figure 22.3 Example of Pulse Output (n = 0 to 3)

22.3.2 External Counter Reset Input

Figure 22.4 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 counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

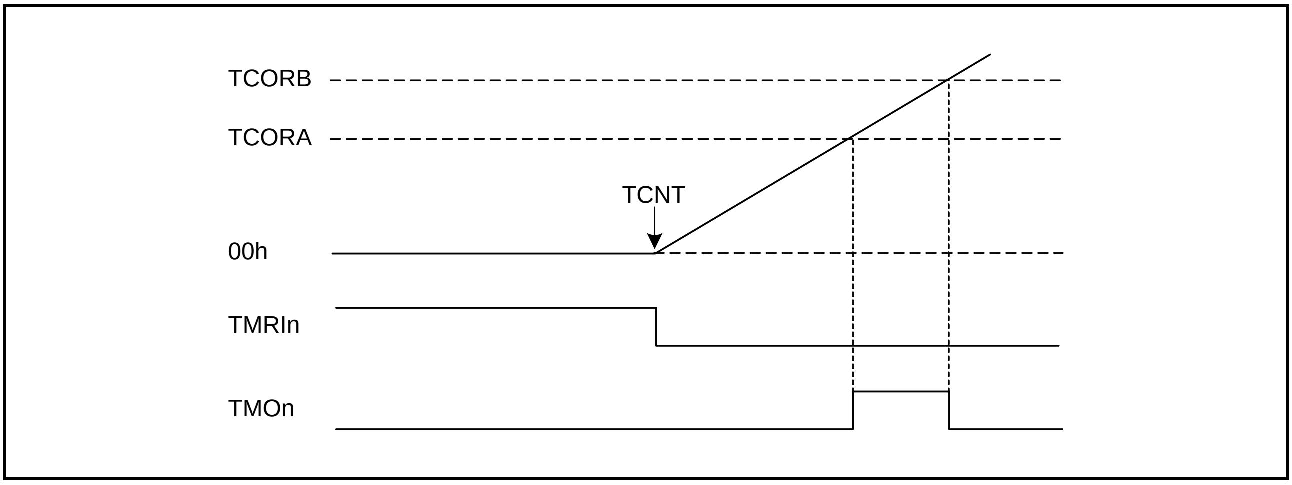


Figure 22.4 Example of External Counter Reset Signal Input (n = 0 to 3)

22.4 Operation Timing

22.4.1 TCNT Count Timing

Figure 22.5 shows the count timing of TCNT for internal clock. Figure 22.6 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

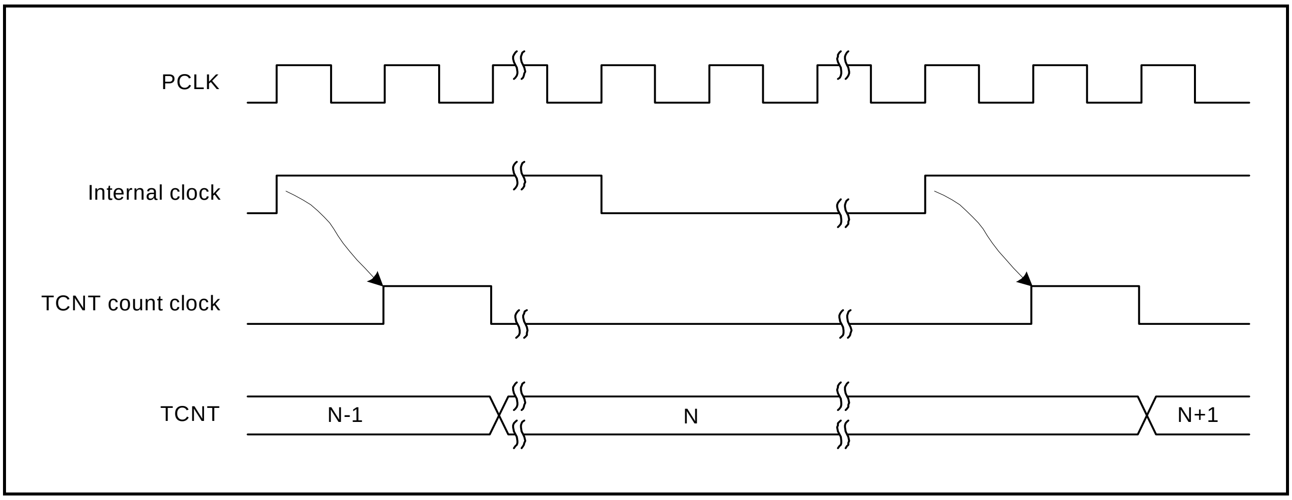


Figure 22.5 Count Timing for Internal Clock

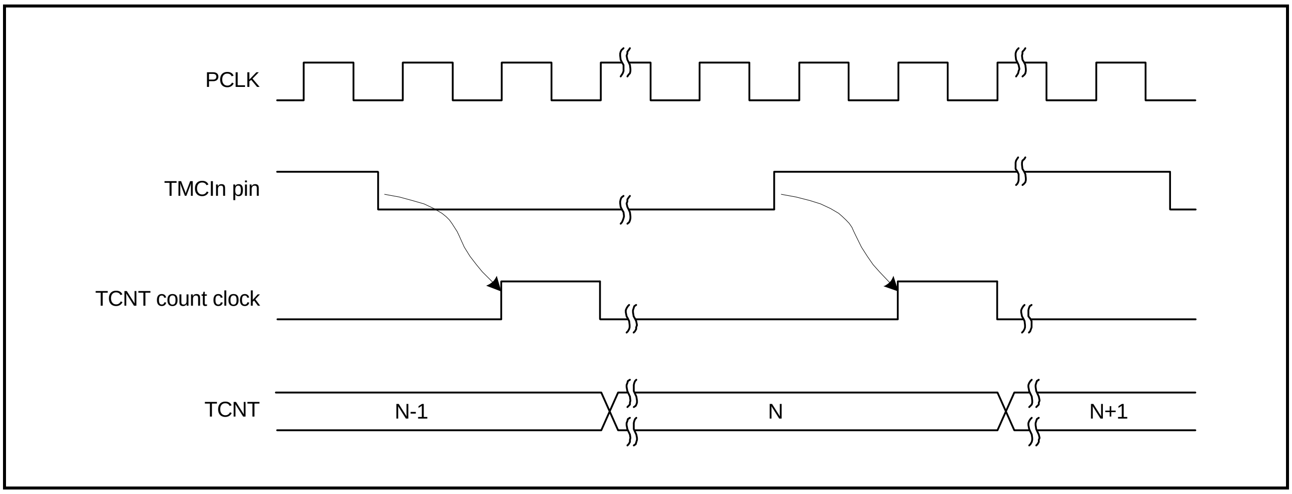


Figure 22.6 Count Timing for External Clock (at Both Edges)

22.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 22.7 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 22.6.

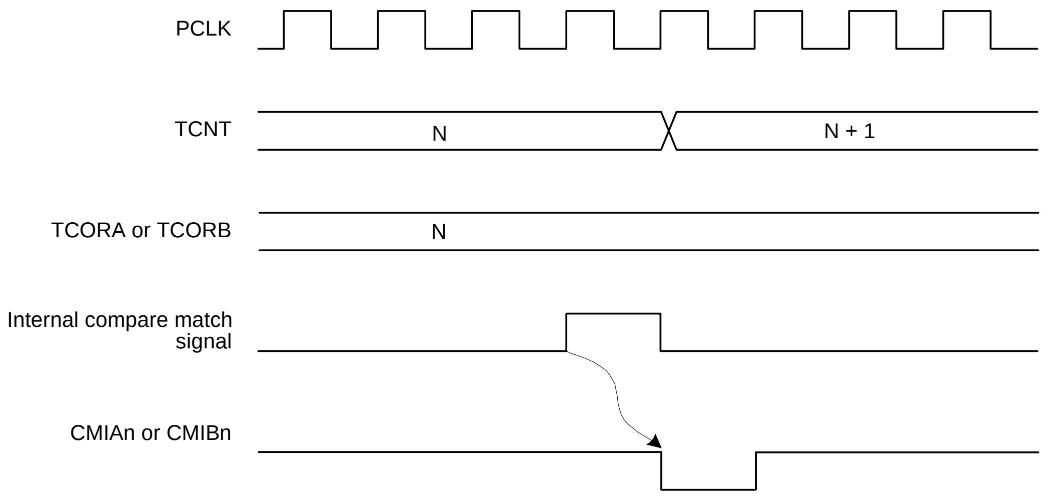


Figure 22.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

22.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).

Figure 22.8 shows the timing when the timer output is toggled by the compare match A signal.

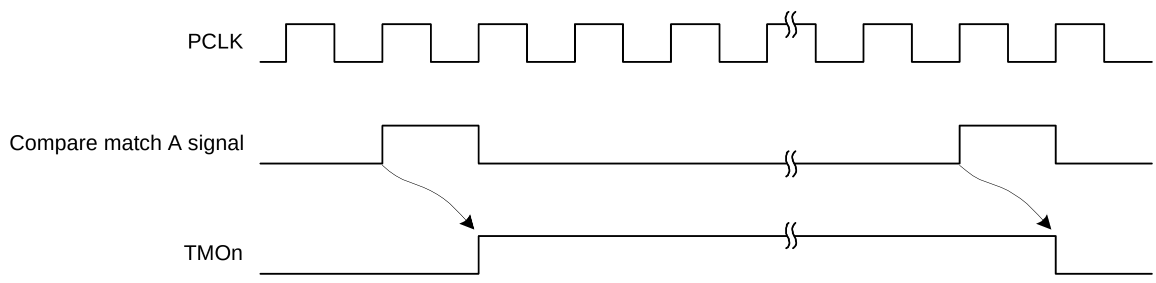


Figure 22.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

22.4.4 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 22.9 shows the timing of this operation.

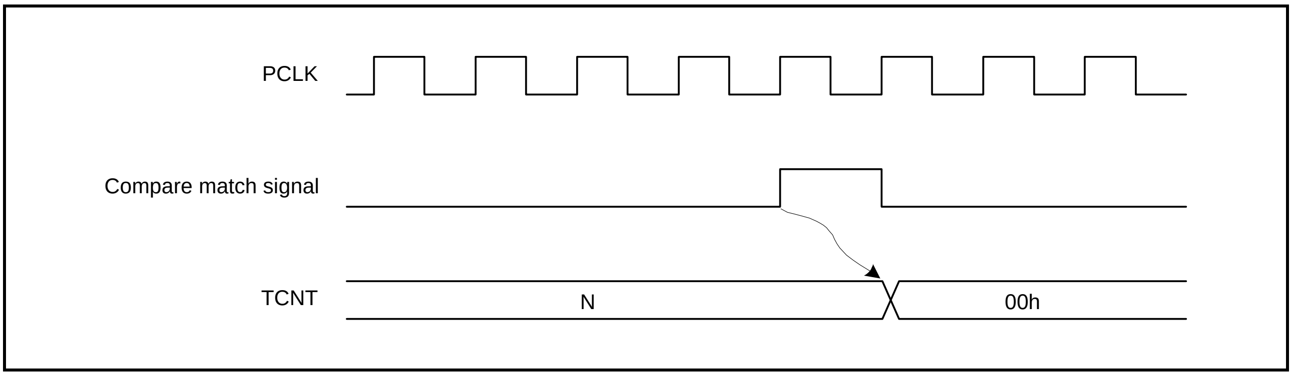


Figure 22.9 Timing of Counter Clear by Compare Match

22.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT.
Figure 22.10 and Figure 22.11 show the timing of this operation.

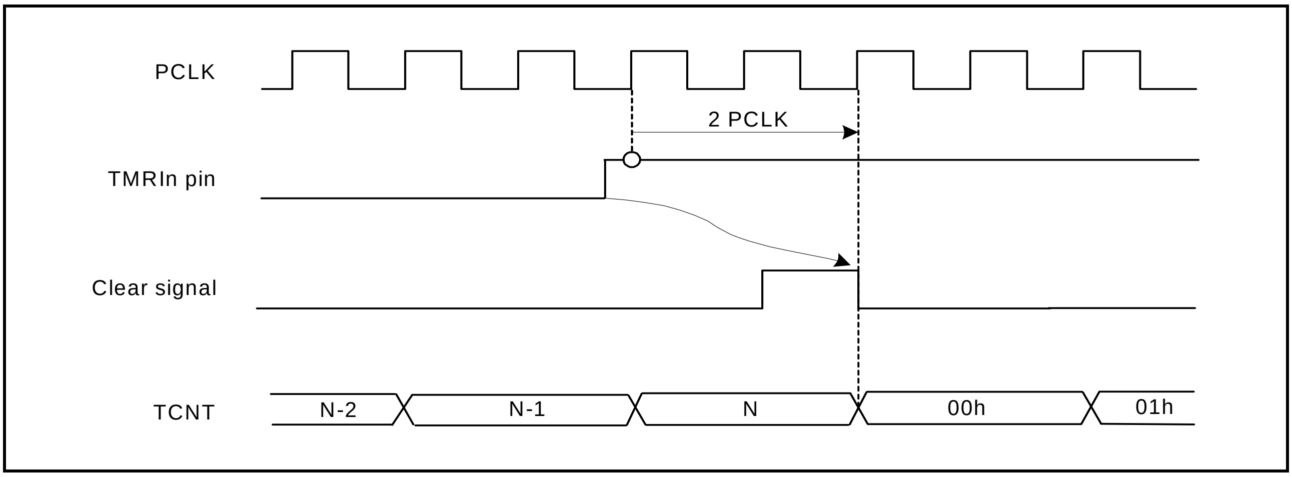


Figure 22.10 Clear Timing by External Counter Reset Signal (Rising Edge)

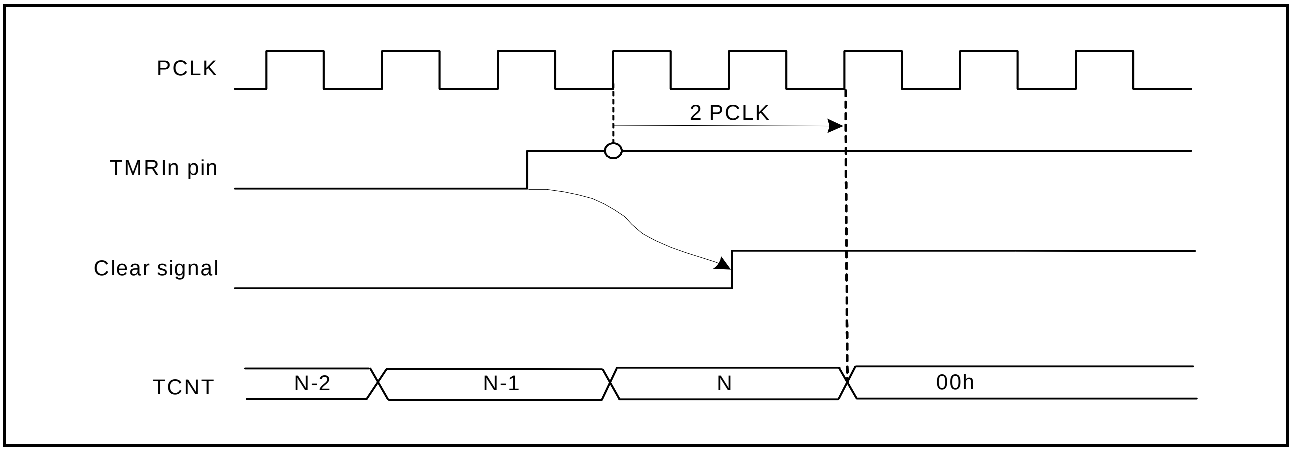


Figure 22.11 Clear Timing by External Counter Reset Signal (High Level)

22.4.6 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 22.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 22.6.

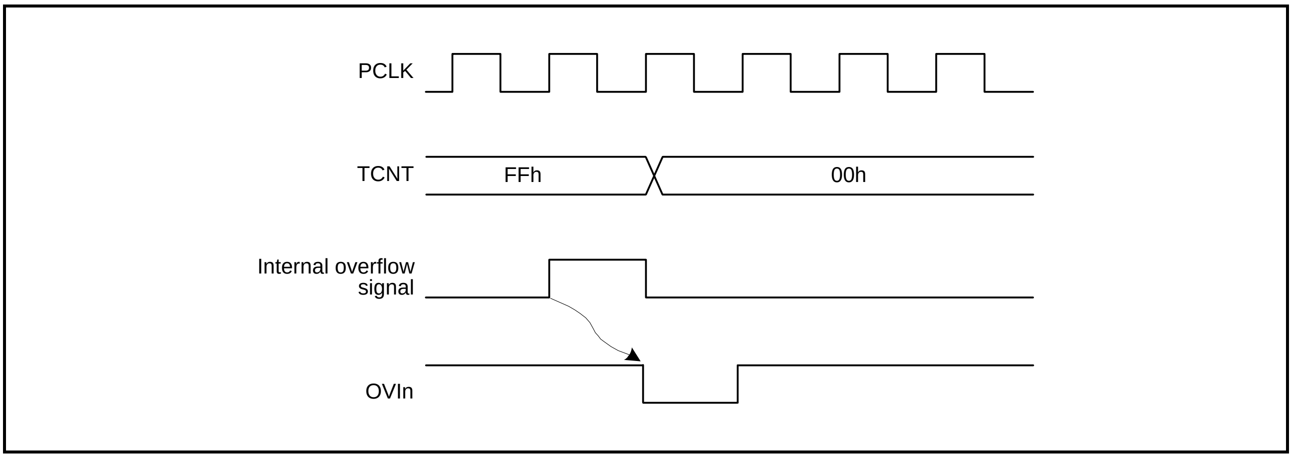


Figure 22.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

22.5 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.

22.5.1 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.

22.5.2 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.

22.6 Interrupt Sources

22.6.1 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 22.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 22.6 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

22.6.2 Startup of the A/D Converter

The compare match A of TMR0 and TMR2 allows the A/D converter*1 to be started.
An A/D conversion start request is issued to the A/D converter in response to a generation of compare match A when the TMRn.TCSR.ADTE bit is 1 (i.e., when an A/D conversion request in response to compare match A is enabled). In this case, the conversion trigger for the 8-bit timer should be selected in the A/D converter to start A/D conversion.

Note 1. For the corresponding unit of the A/D converter, see section 29, 12-Bit A/D Converter (S12ADE).

Table 22.7 Startup of A/D Converter

Module Symbol	Unit	Target	A/D Startup Request	A/D Conversion Start Request
S12AD	0	Between TMR0.TCORA and TMR0.TCNT	Compare match	TMTRG0AN_0
	1	Between TMR2.TCORA and TMR2.TCNT		TMTRG0AN_1

22.7 Usage Notes

22.7.1 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.

22.7.2 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).

$$f = PCLK / (N + 1)$$

22.7.3 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 22.13.

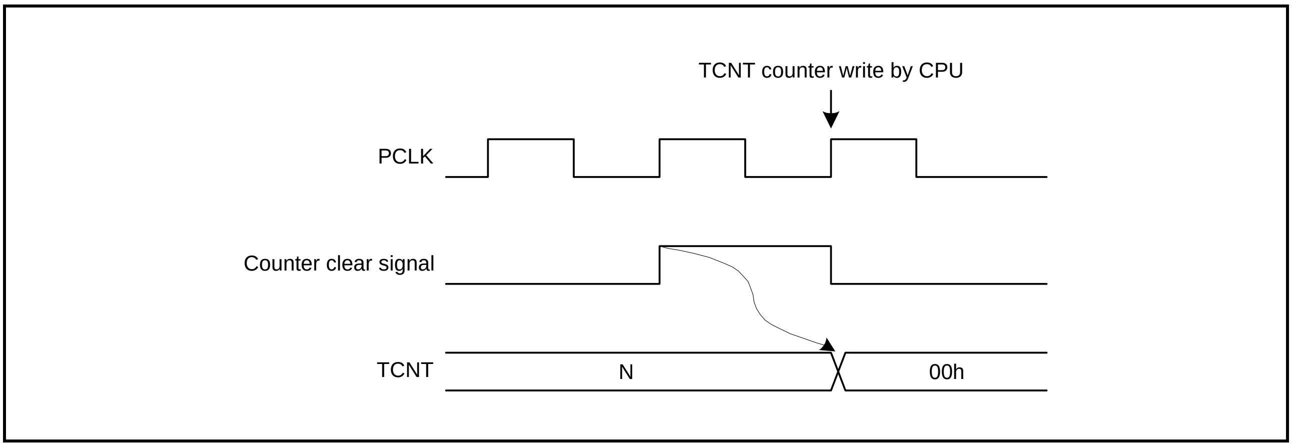


Figure 22.13 Conflict between TCNT Write and Counter Clear

22.7.4 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 22.14.

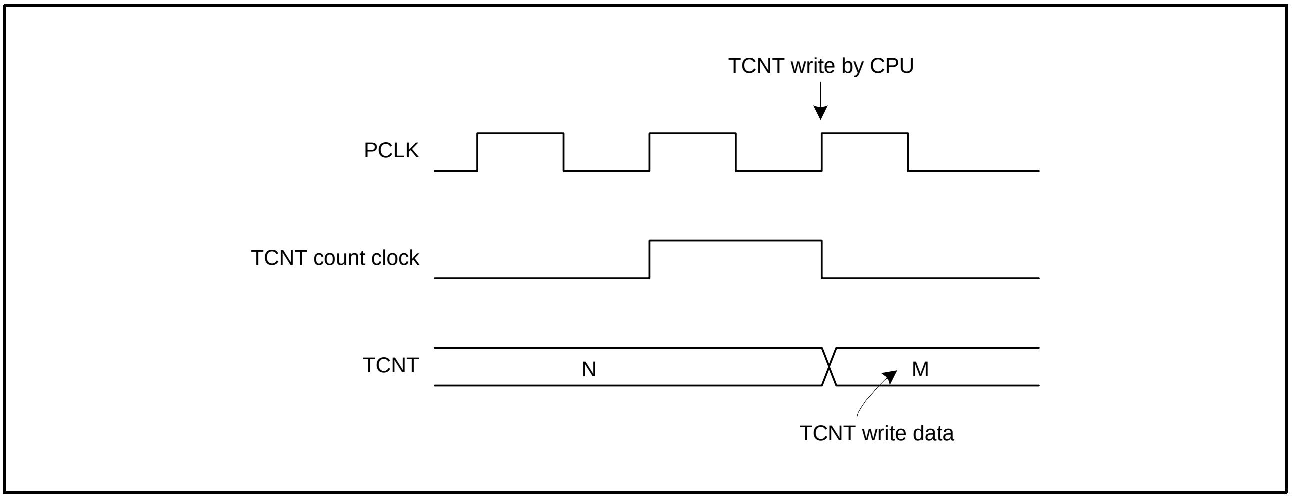


Figure 22.14 Conflict between TCNT Write and Increment

22.7.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 22.15, the write takes priority and the compare match signal does not reach High level.

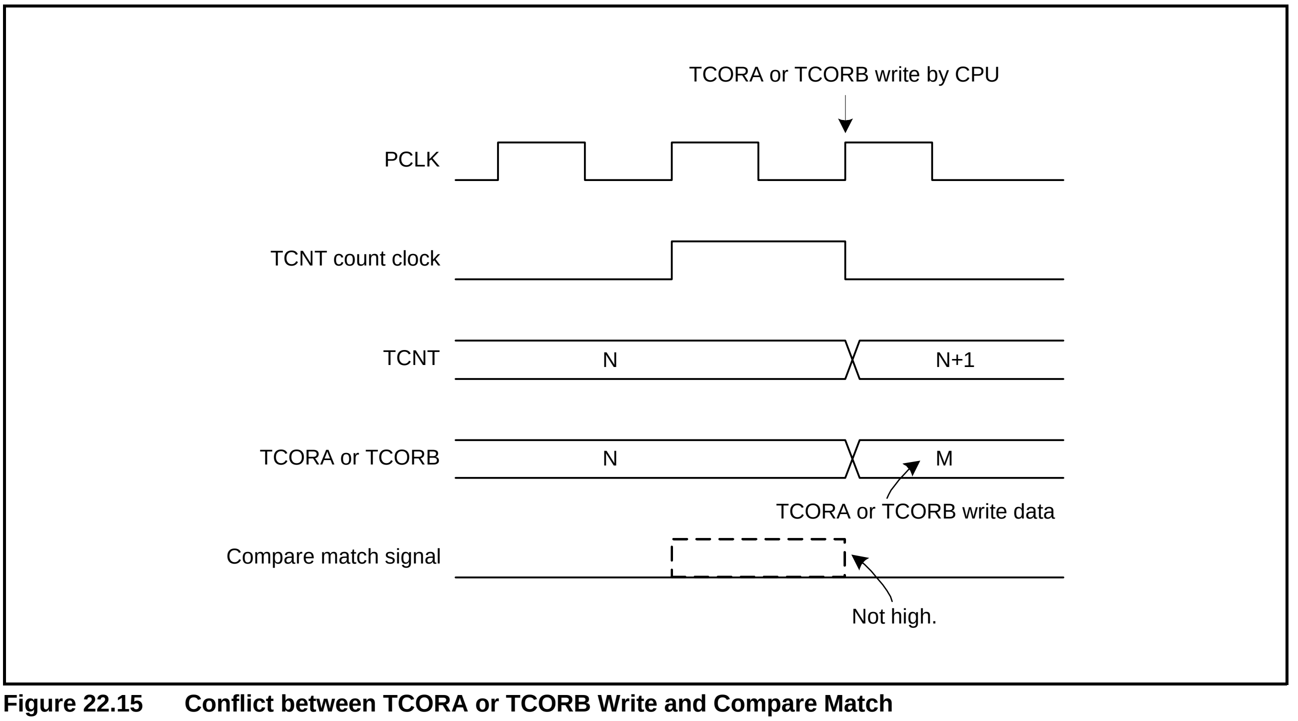


Figure 22.15 Conflict between TCORA or TCORB Write and Compare Match

22.7.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 22.8.

Table 22.8 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

22.7.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 22.9 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 22.9, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented. The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 22.9 Switching of Internal Clocks and TCNT Operation (1/2)

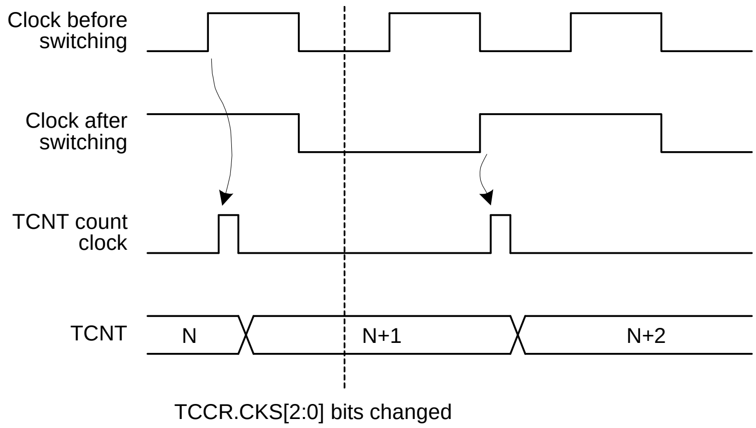
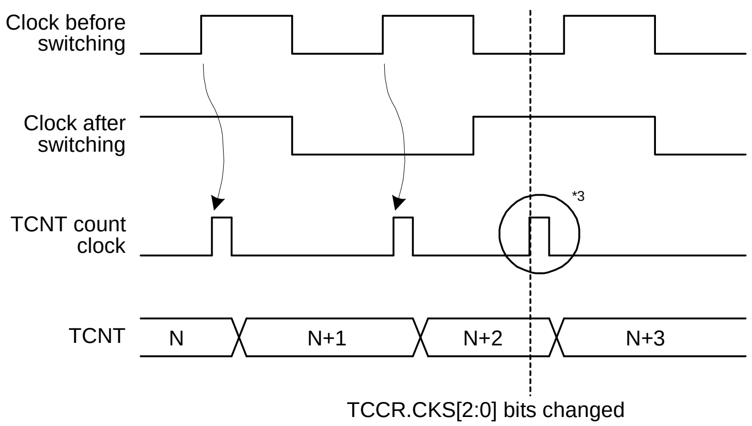
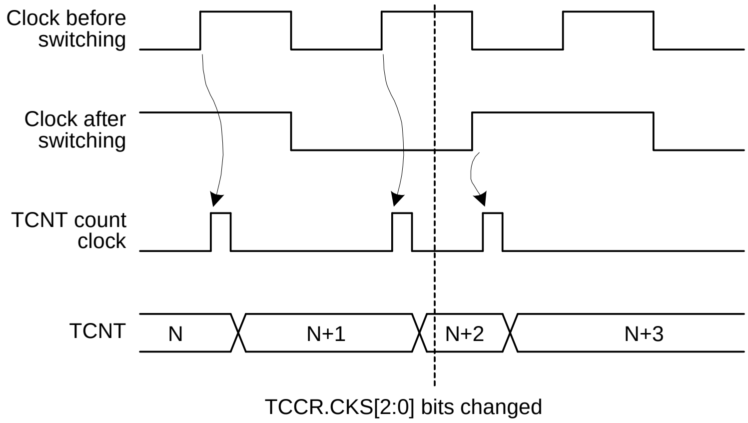
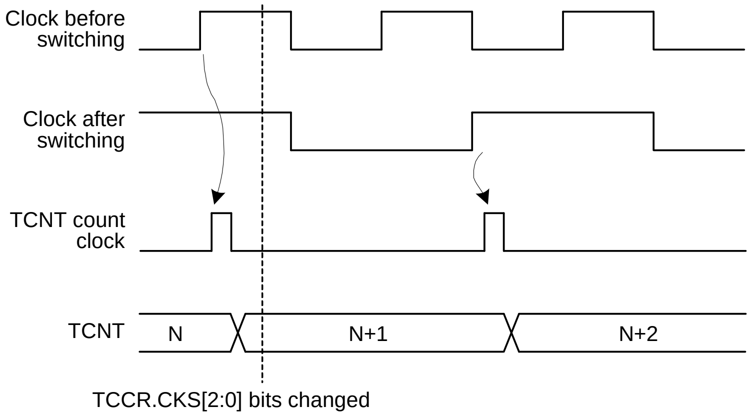
<table>
<tr><th>No.</th><th>Timing to Change the TCCR.CKS[2:0] Bits</th><th>TCNT Counter Operation</th></tr>
<tr><td>1</td><td>Switching from low to low*1</td><td>
</td></tr>
<tr><td>2</td><td>Switching from low to high*2</td><td>
</td></tr>
</table>

Table 22.9 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

22.7.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

22.7.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.

At this time, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 22.16 shows operation timing when the compare match interrupt signal is continuously output.

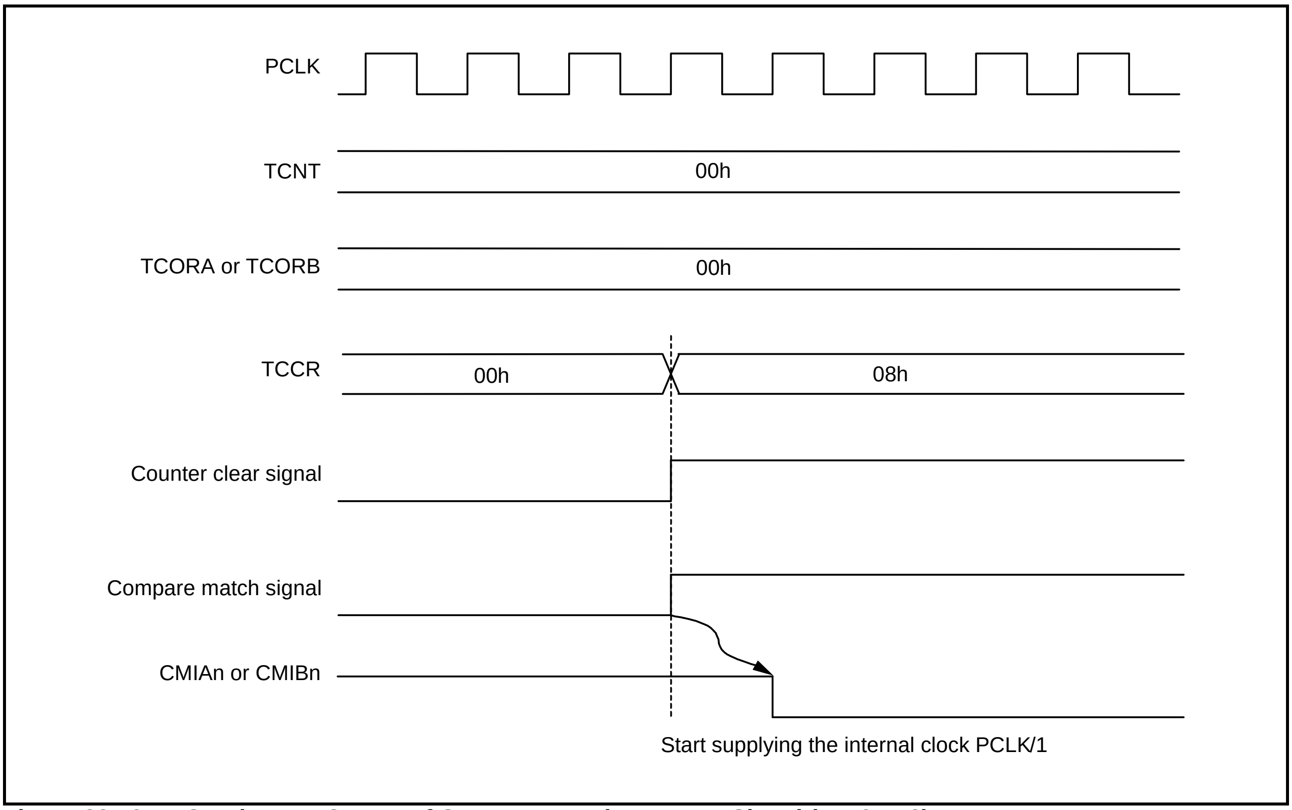


Figure 22.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

23. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

23.1 Overview

Table 23.1 lists the specifications for the CMT.
Figure 23.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 23.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Low power consumption function	Each unit can be placed in a module stop state.

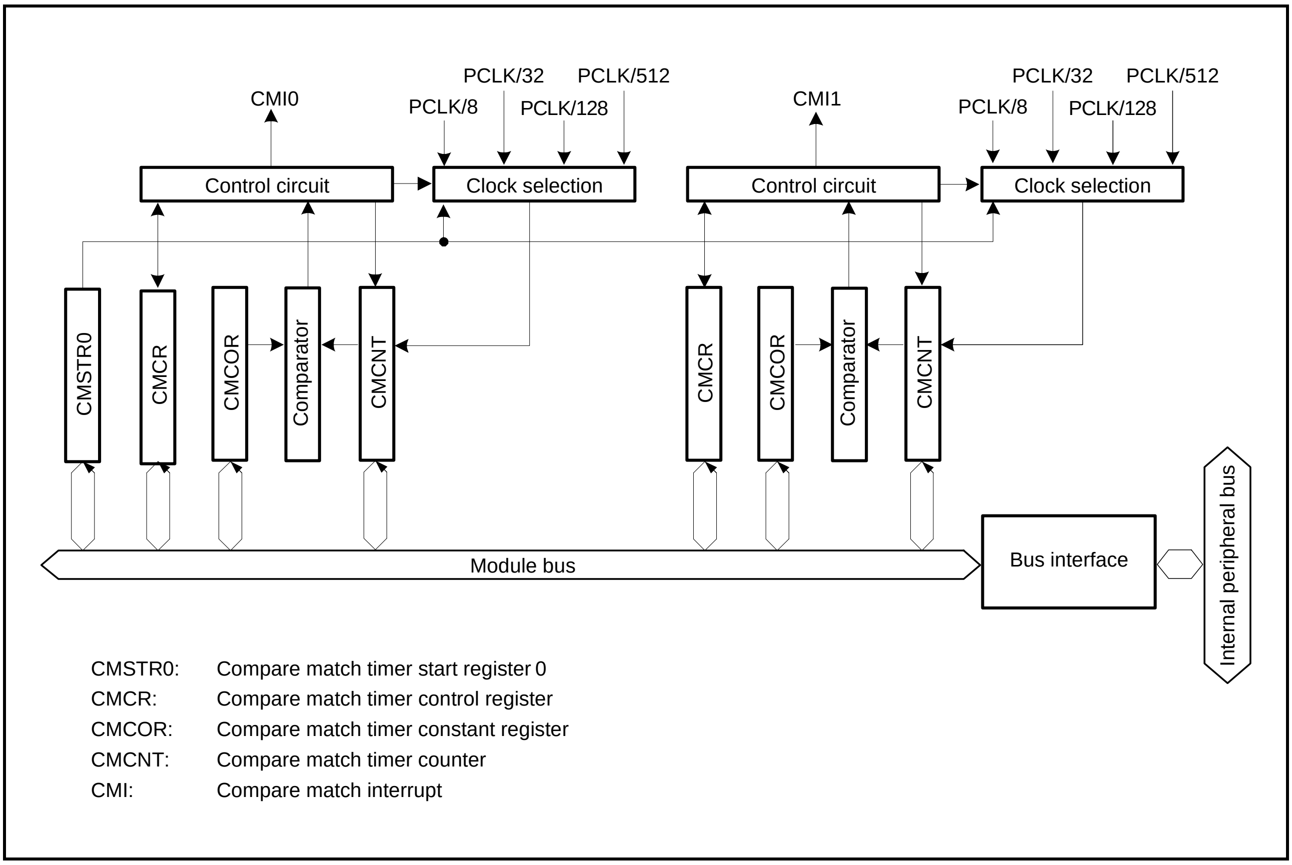


Figure 23.1 CMT (Unit 0) Block Diagram

23.2 Register Descriptions

23.2.1 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

23.2.2 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

23.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h, CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

23.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah

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

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

23.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch

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

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

23.3 Operation

23.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 23.2 shows the operation of the CMCNT counter.

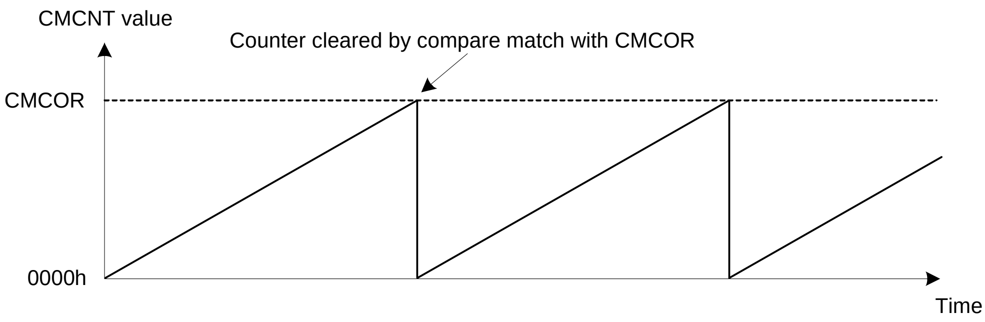


Figure 23.2 CMCNT Counter Operation

23.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 23.3 shows the timing of the CMCNT counter.

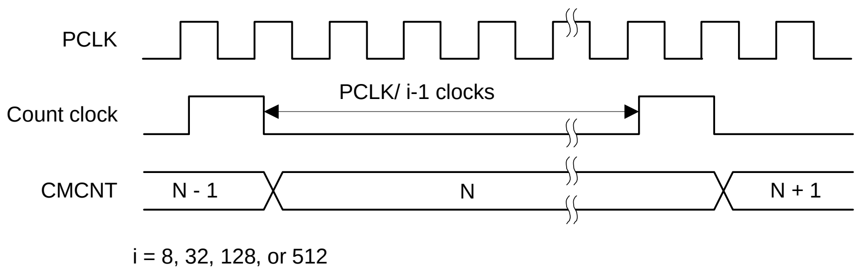


Figure 23.3 CMCNT Count Timing

23.4 Interrupts

23.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUb).

Table 23.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMI0	Compare match in CMT0	Possible
CMI1	Compare match in CMT1	Possible
CMI2	Compare match in CMT2	Possible
CMI3	Compare match in CMT3	Possible

23.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 23.4 shows the timing of a compare match interrupt.

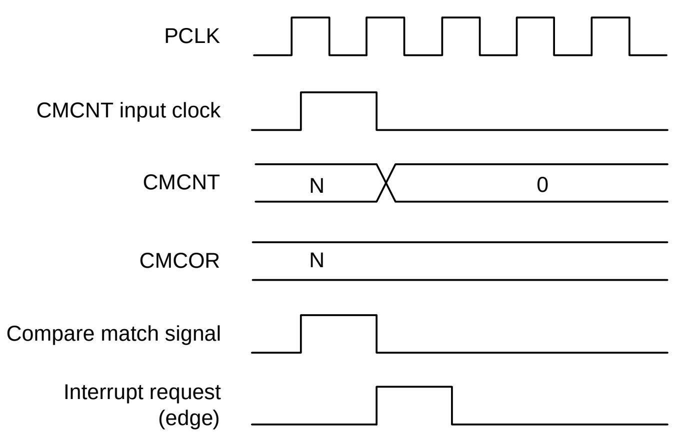


Figure 23.4 Timing of a Compare Match Interrupt

23.5 Usage Notes

23.5.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

23.5.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 23.5 shows the timing to clear the CMCNT counter.

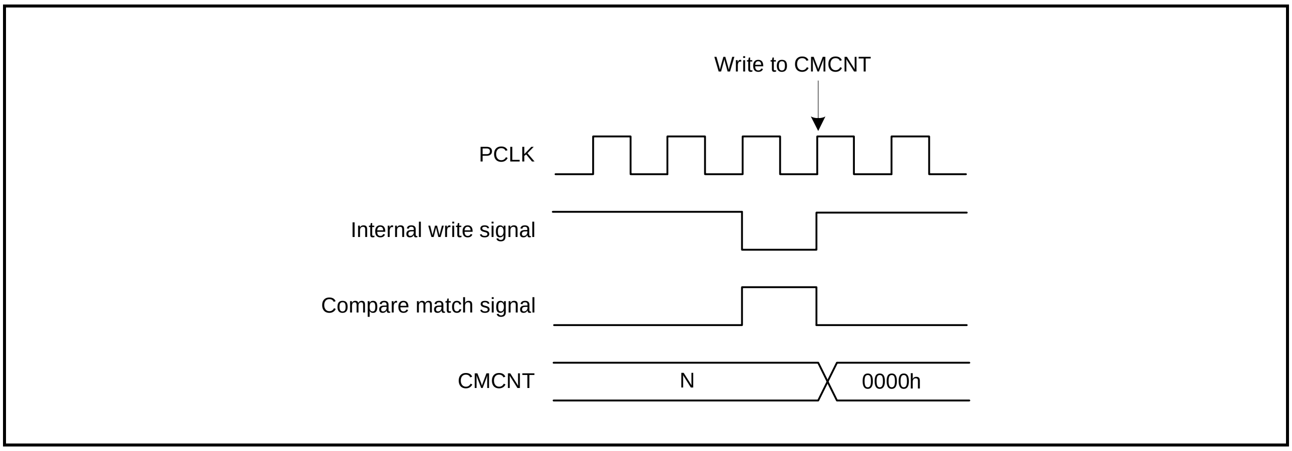


Figure 23.5 Conflict between CMCNT Counter Writing and Compare Match

23.5.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 23.6 shows the timing to write the CMCNT counter.

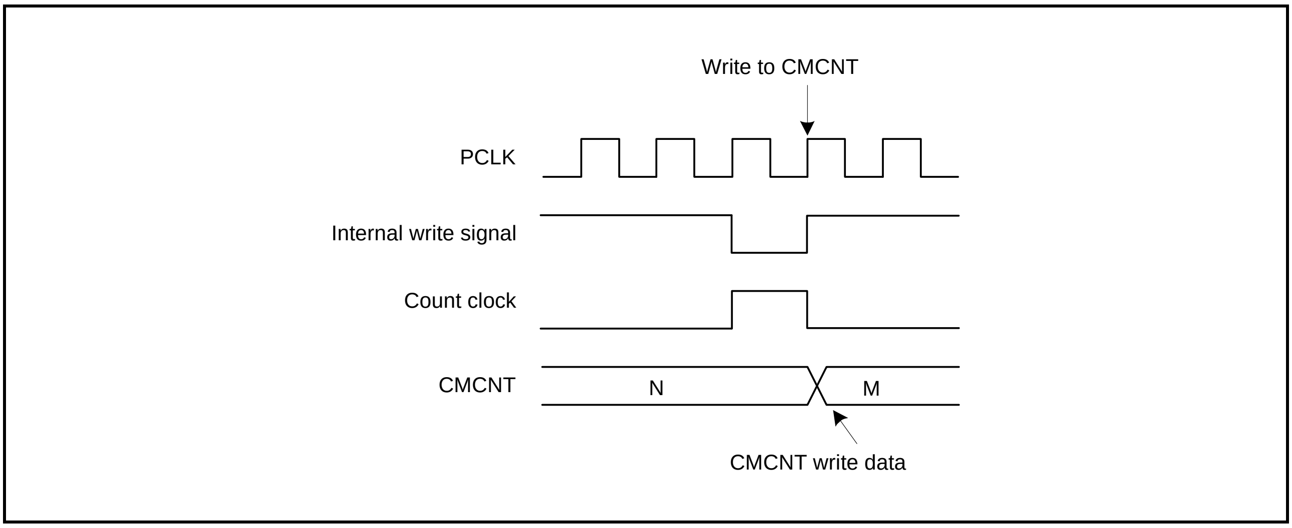


Figure 23.6 Conflict between CMCNT Counter Writing and Incrementing

24. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

24.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated low-speed clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 24.1 lists the specifications of the IWDT and Figure 24.1 shows a block diagram of the IWDT.

Table 24.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Counting automatically starts after a reset (auto-start mode) • Counting is started (register start mode) by refreshing the counter (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error occurs Counting restarts (In auto-start mode, counting automatically restarts after a reset or after a non-maskable interrupt request is output. In register start mode, counting restarts after refreshing.)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Down-counter underflows • Refreshing 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 divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) $\geq$ 4 × (the frequency of the count source after divide).

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.

Figure 24.1 is a block diagram of the IWDT.

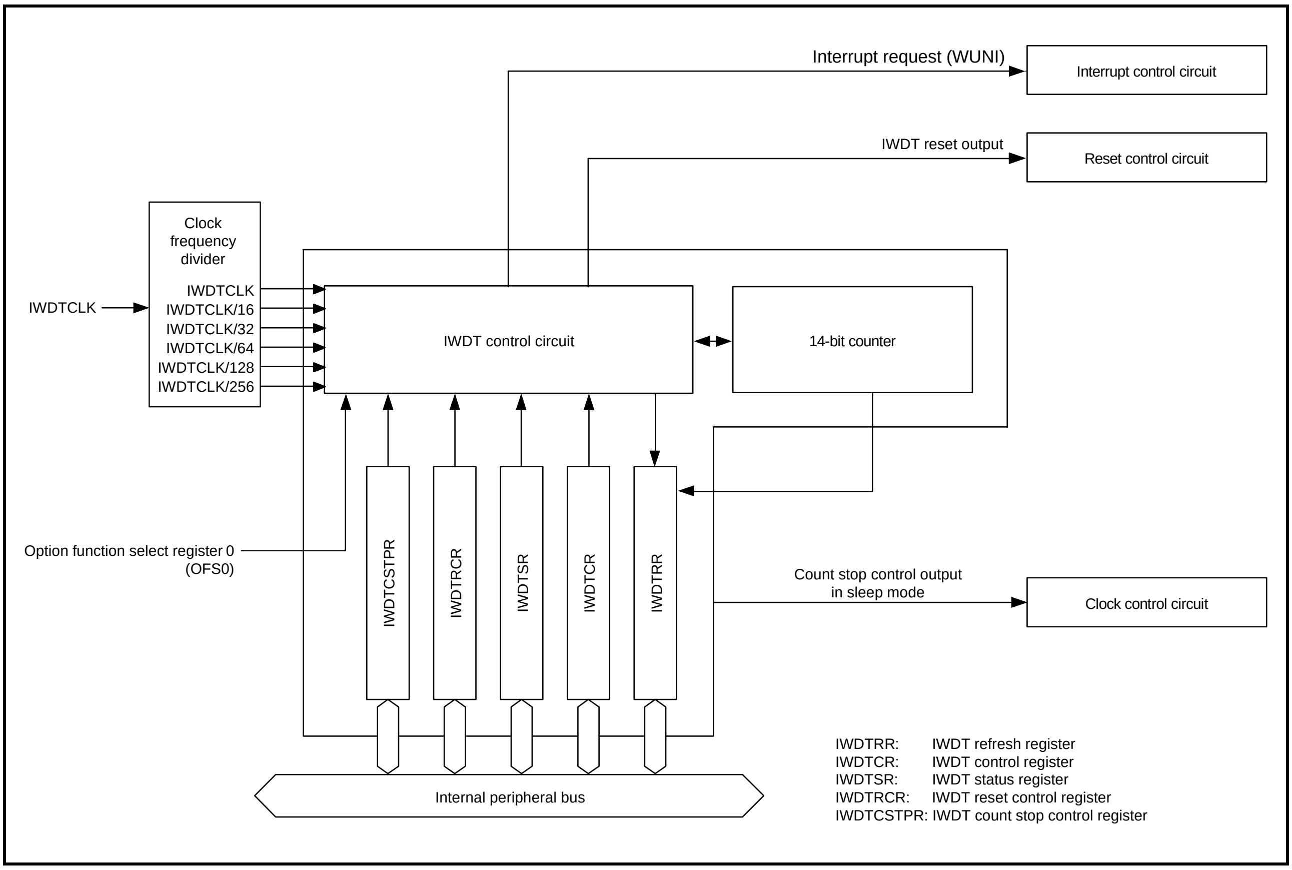


Figure 24.1 IWDT Block Diagram

24.2 Register Descriptions

24.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h

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

Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]) in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period select bits (TOPS[1:0]) in the IWDT control register (IWDTCR) in the first refresh operation after release from the reset state.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 24.3.3, Refresh Operation.

24.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 24.3.2, 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 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 24.2.

Table 24.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	No division	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

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

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 24.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 24.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

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

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 24.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting is greater than or equal to window start setting, the window end setting is set to 0%.

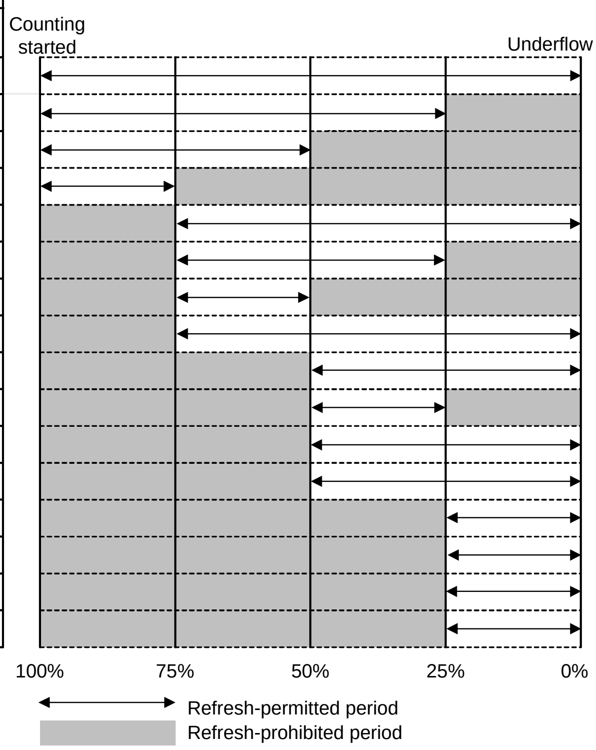


Figure 24.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

24.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

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

The IWDTSR register is initialized by the reset source of the IWDT. The IWDTSR register is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

24.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request 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 24.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in option function select register 0. For details, refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

24.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTSCTPR 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 deep sleep mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 24.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in option function select register 0 (OFS0). For details, refer to section 24.3.8, 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 deep sleep mode.

24.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

24.3 Operation

24.3.1 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.

24.3.1.1 Register Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 1, register start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled.
After the reset state is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the timeout period select bits (IWDTCR.TOPS[1:0]).
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) to select either reset output or interrupt request output.

Figure 24.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The IWDT window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

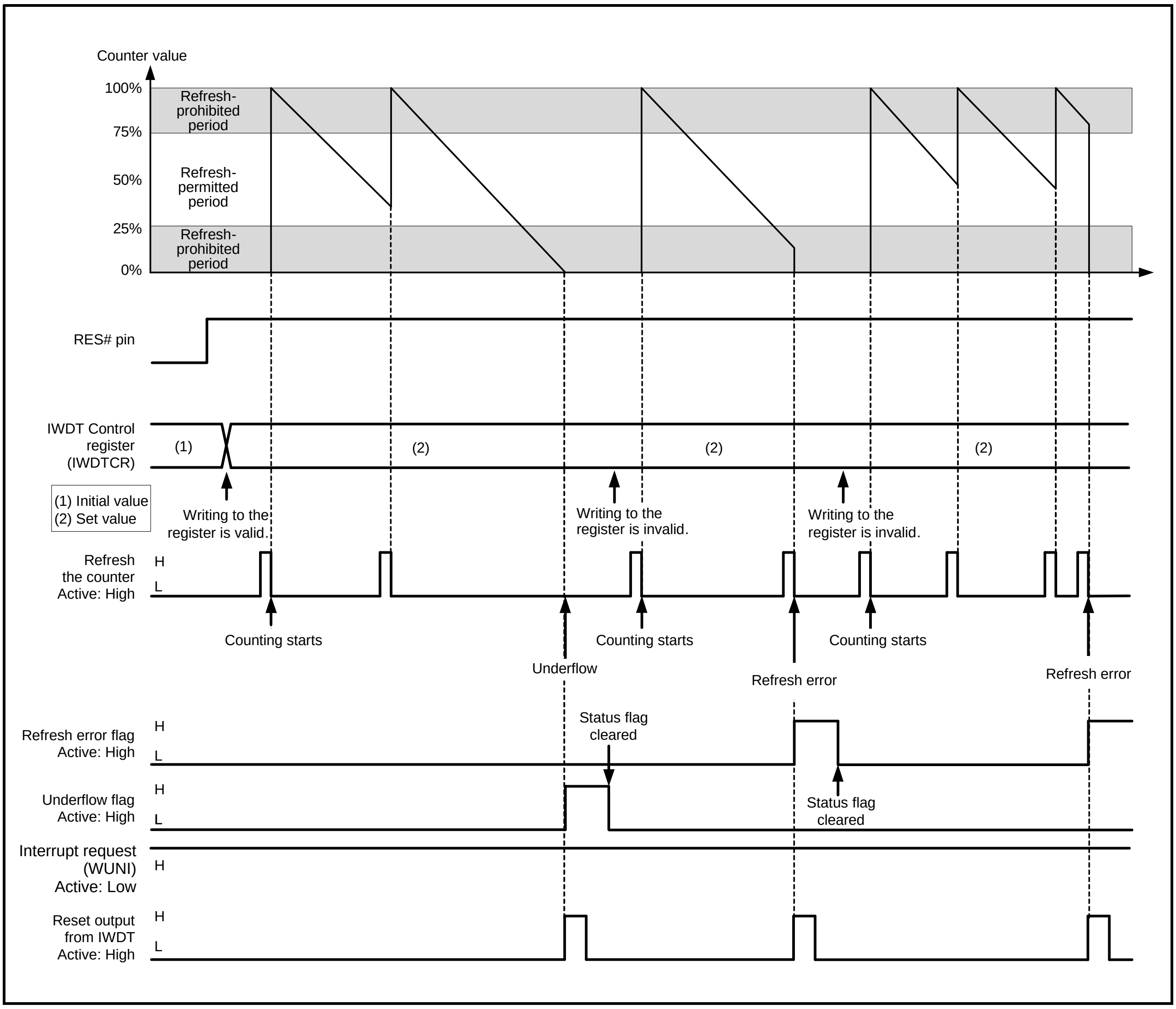


Figure 24.3 Operation Example in Register Start Mode

24.3.1.2 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 divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in option function select register 0 (OFS0). When the reset state is released, the counter automatically starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]).
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 24.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The IWDT window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

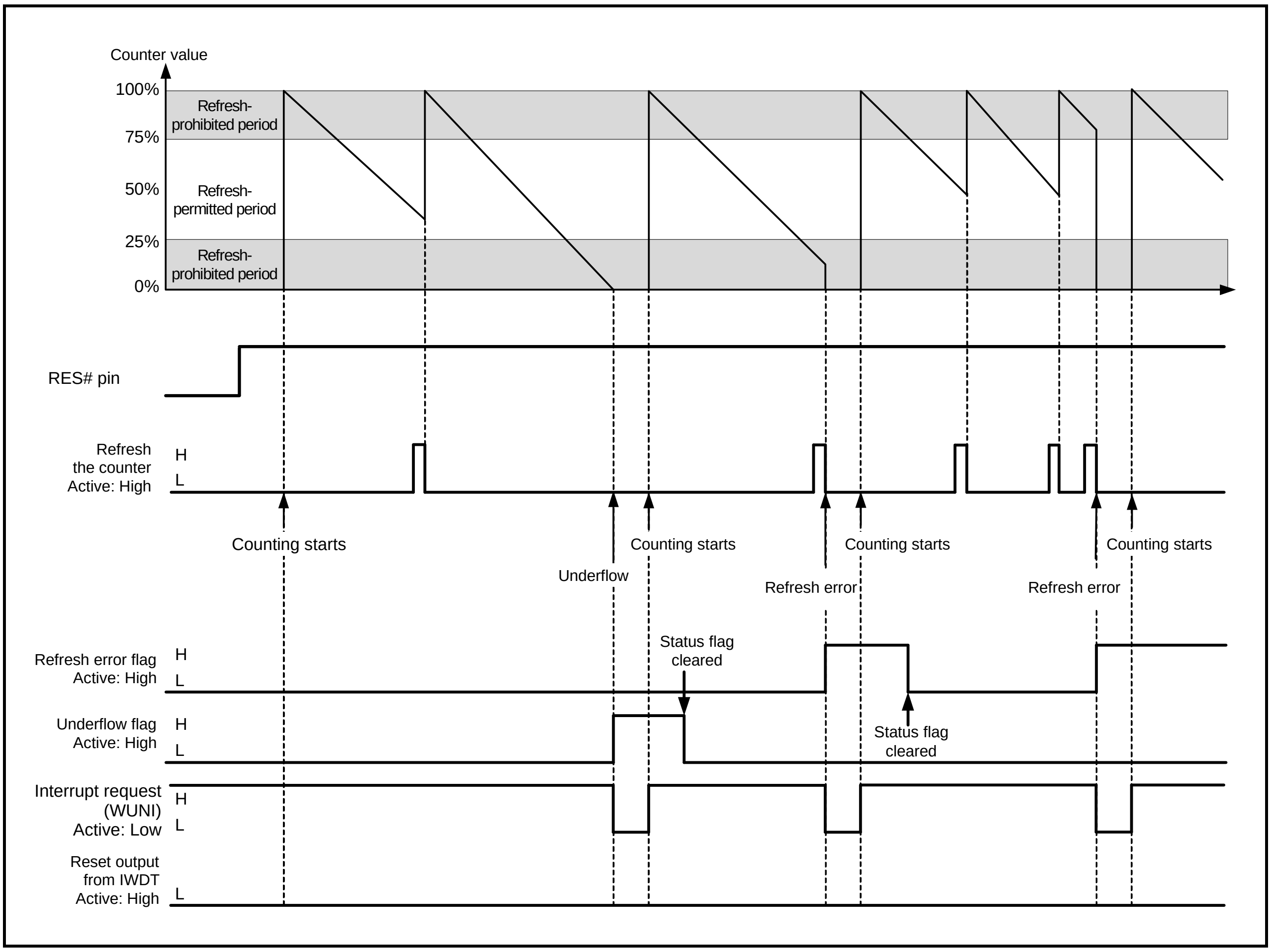


Figure 24.4 Operation Example in Auto-Start Mode

24.3.2 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 the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.
This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.
Figure 24.5 shows control waveforms produced in response to writing to the IWDTCR register.

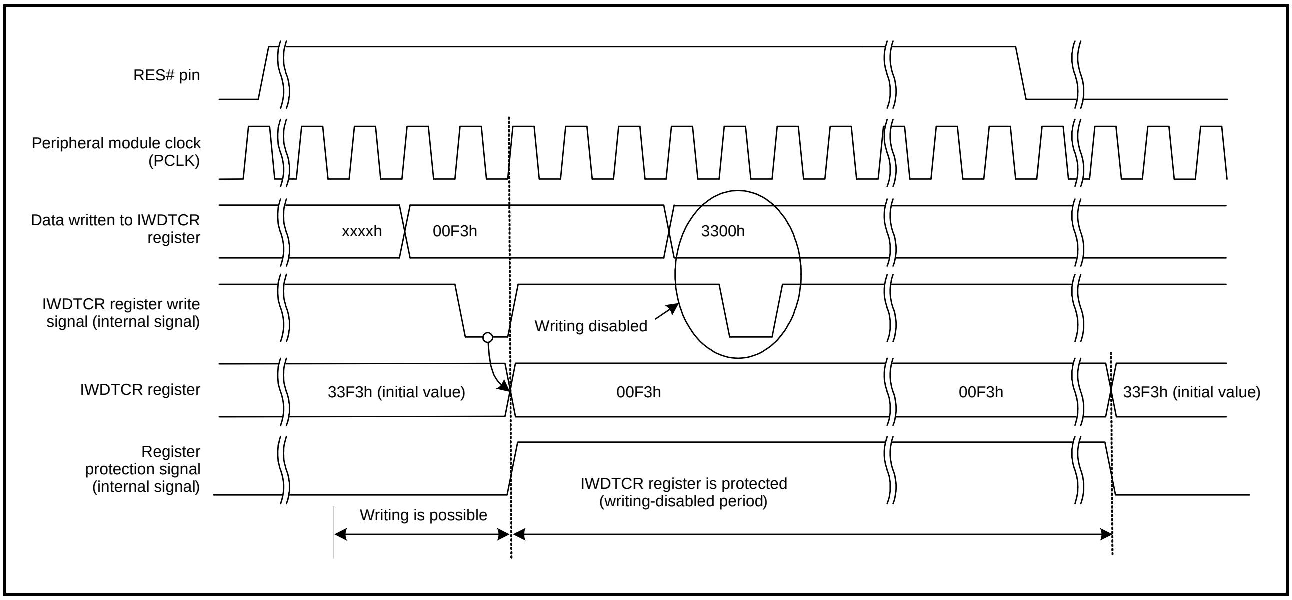


Figure 24.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

24.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDT refresh register (IWDTRR).
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the clock divide ratio selection bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 03FFh, even if 00h is written to the IWDTRR register before 03FFh is reached (0402h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 03FFh.
- When the window end position is set to 03FFh, refreshing is done if 0403h (four-count cycles before 03FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 24.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

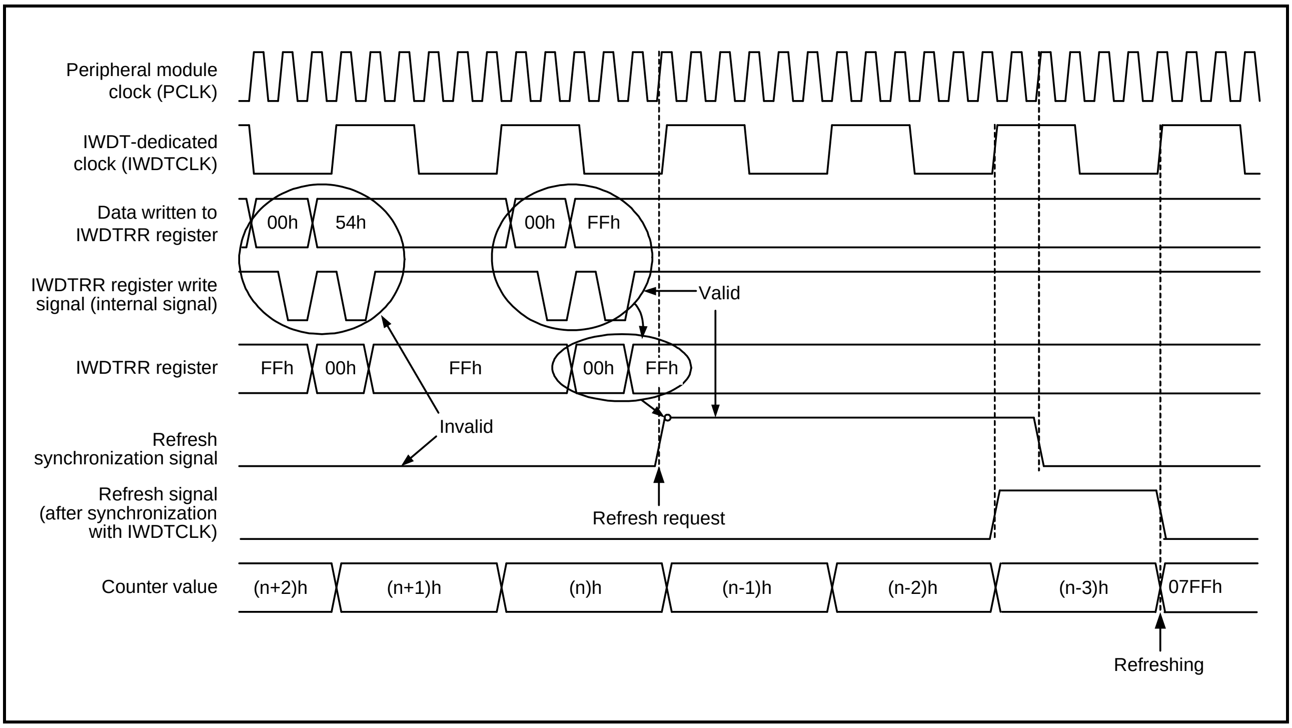


Figure 24.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

24.3.4 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.

24.3.5 Reset Output

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal.
After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

24.3.6 Interrupt Sources

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in option function select register 0 (OFS0) is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 14, Interrupt Controller (ICUb).

Table 24.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation
WUNI	Counter underflow Refresh error	Not possible

24.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT status register. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 24.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

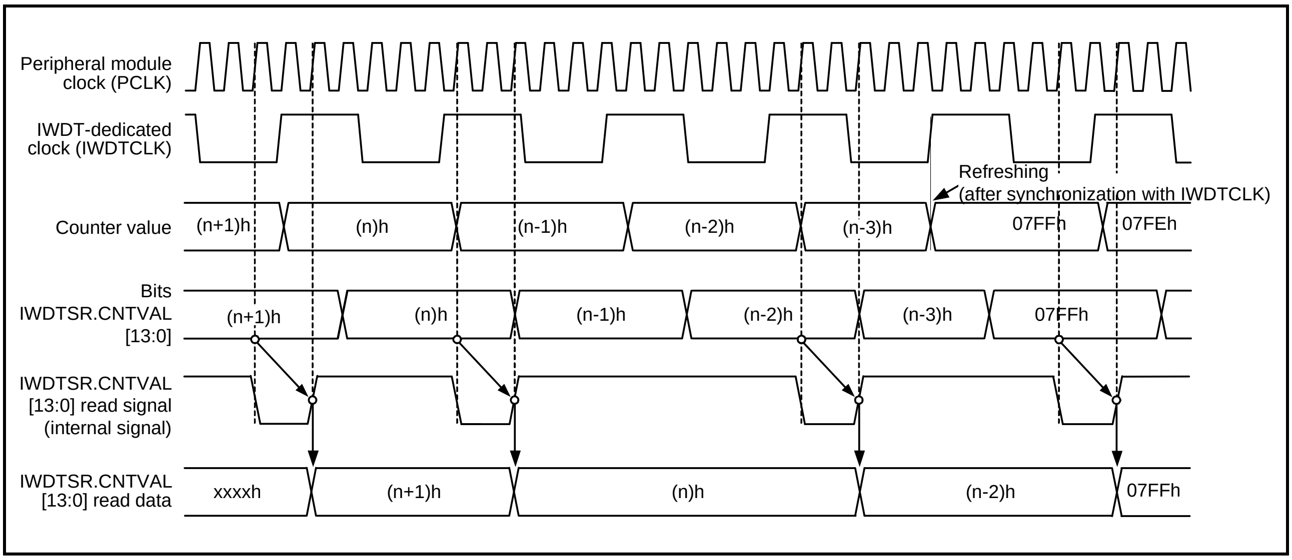


Figure 24.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

24.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 24.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on option function select register 0 (OFS0), refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 24.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

24.4 Usage Notes

24.4.1 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.

24.4.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) $\geq 4 \times$ (the frequency of the count source after divide).

25. Serial Communications Interface (SCIg)

This MCU has two independent serial communications interface (SCI) channels. The SCI consists of the SCIg module (SCI1 and SCI5).
The SCIg module (SCI1 and SCI5) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I^2C-bus interfaces when configured for single-master systems.
In this section, “PCLK” is used to refer to PCLKB.

25.1 Overview

Table 25.1 lists the specifications of the SCIg module and Table 25.2 lists the specifications of the individual SCI channels.
Figure 25.1 and Figure 25.2 show the block diagrams of the SCIg module.

Table 25.1 SCIg Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		See Table 25.3 to Table 25.5.
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, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.

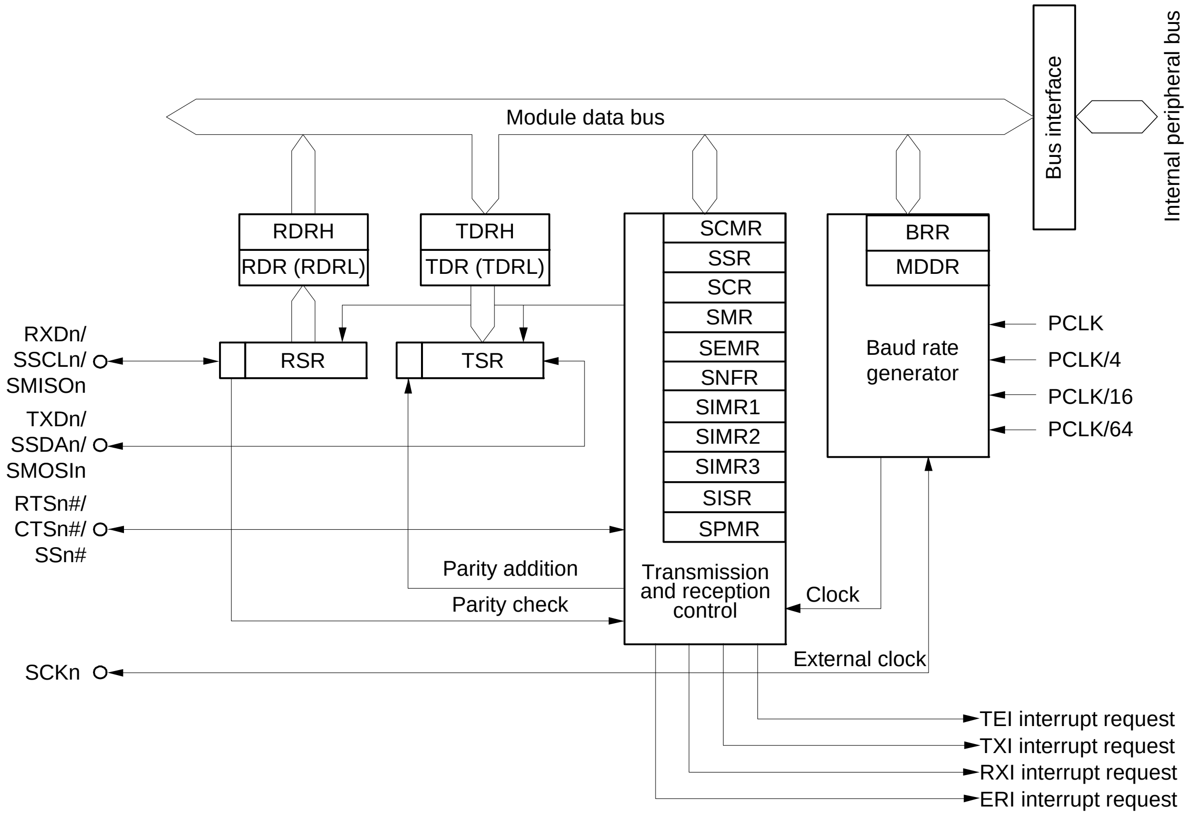
Table 25.1 SCIg Specifications (2/2)

Item		Description
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (see section 25.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I2C mode, only MSB first is available.

Table 25.2 Functions of SCI Channels

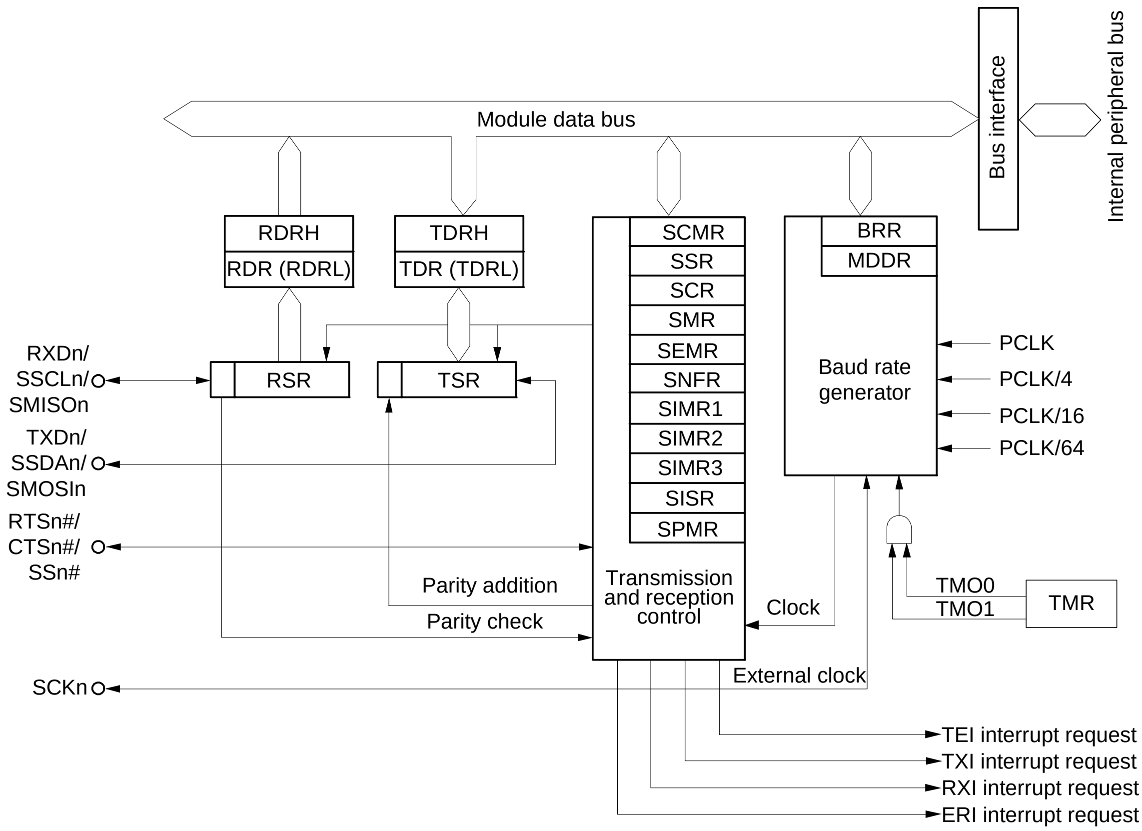
Item	SCI1	SCI5
Asynchronous mode	Available	Available
Clock synchronous mode	Available	Available
Smart card interface mode	Available	Available
Simple I2C mode	Available	Available
Simple SPI mode	Available	Available
TMR clock input	Not available	Available



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
MDDR: Modulation duty register
SEMR: Serial extended mode register

SNFR: Noise filter setting register
SIMR1: I^{2}C mode register 1
SIMR2: I^{2}C mode register 2
SIMR3: I^{2}C mode register 3
SISR: I^{2}C status register
SPMR: SPI mode register
RDRH: Receive data register H
RDRL: Receive data register L
TDRH: Transmit data register H
TDRL: Transmit data register L

Figure 25.1 Block Diagram of SCIg (SCI1)



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
MDDR: Modulation duty register
SEMR: Serial extended mode register

SNFR: Noise filter setting register
SIMR1: I2C mode register 1
SIMR2: I2C mode register 2
SIMR3: I2C mode register 3
SISR: I2C status register
SPMR: SPI mode register
RDRH: Receive data register H
RDRL: Receive data register L
TDRH: Transmit data register H
TDRL: Transmit data register L

Figure 25.2 Block Diagram of SCIg (SCI5)

Table 25.3 to Table 25.5 list the pin configuration of the SCIs for the individual modes.

Table 25.3 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output

Table 25.4 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output

Table 25.5 SCI Pin Configuration in 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

25.2 Register Descriptions

25.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.
When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

25.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h

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

RDR is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from RSR to RDR. Then the RSR register can receive the next data.
Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.
Read RDR only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs.
RDR cannot be written to by the CPU.

25.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h

	RDRHL															
	RDRH								RDRL							
	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	0	0	0	0	0	0	0	0

RDRH and RDRL are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

RDRL is the shadow register of RDR; i.e. access to RDRL is equivalent to access to RDR.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read RDRH and RDRL should be performed only once in the order from RDRH to RDRL when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from RDRL.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in RDRH are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

25.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h

	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 the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

25.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh

	TDRHL															
	TDRH								TDRL							
	b7	b6	b5	b4	b3	b2	b1	b0	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

TDRH and TDRL are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.

TDRL is the shadow register of TDR; i.e. access to TDRL is equivalent to access to TDR.

When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to TSR; i.e., transmitting is started.

The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in RDRH are fixed to 1. These bits are read as 1. The write value should be 1.

Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from TDRH to TDRL when a transmit data empty interrupt (TXI) request is issued.

The TDRHL register can be accessed in 16-bit units.

25.2.6 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.

25.2.7 Serial Mode Register (SMR)

Note: Some bits in SMR have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 25.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.
For the relation between the settings of these bits and the baud rate, refer to section 25.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the CHR1 bit in SCMR.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 25.6 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*3
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (refer to section 25.2.11, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (refer to section 25.2.11, Bit Rate Register (BRR)).
Note 3. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 25.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 25.6.4, Receive Data Sampling Timing and Reception Margin.

Table 25.6 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (refer to section 25.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 25.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 25.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 25.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 25.6.8, Clock Output Control.

25.2.8 Serial Control Register (SCR)

Note: Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h

	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 SCI1 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b1, b0	CKE[1:0]	Clock Enable	• For SCI5 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port according to the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The TMR clock can be used. The SCKn pin is available for use as an I/O port according to the I/O port settings when the TMR clock is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
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

Bit	Symbol	Bit Name	Description	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^{2}C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 25.4, Multi-Processor Communications Function.
When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.
When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags ORER and FER to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SCR 0008 A022h, SMCI5.SCR 0008 A0A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port according to the I/O port settings.) 0 1: Clock output 1 x: (Setting prohibited) • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 25.11, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 25.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

25.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is read from RDR

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is written to TDR

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SSR 0008 A024h, SMCI5.SSR 0008 A0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit = 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is read from RDR

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is written to TDR

25.2.10 Smart Card Mode Register (SCMR)

Address(es): SMCI1.SCMR 0008 A026h, SMCI5.SCMR 0008 A0A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I²C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert	0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction	This bit can be used in the following modes. • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode Set this bit to 1 if operation is to be in simple I²C mode. 0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1	(Only valid in asynchronous mode)*2 Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 25.7 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 3. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I²C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

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 the SMR register.

CHR1 bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the CHR bit in SMR.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 25.7 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (refer to section 25.2.11, Bit Rate Register (BRR)).

25.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h

	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 25.8 shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.

The initial value of BRR is FFh.

BRR can be read from by the CPU, but it can be written to only when the TE and RE bits in the SCR register are 0.

Table 25.8 Relationship between N Setting in BRR and Bit Rate B

Mode	SEMR Settings		BRR Setting	Error
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor transfer	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
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 I^2C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)

N: BRR setting for on-chip baud rate generator ($0 \leq N \leq 255$)

PCLK: Operating frequency (MHz)

n and S: Determined by the settings of the SMR and SCMR registers as listed in the table below.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C standard.

Table 25.9 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 25.10 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

Table 25.11 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 25.12 lists examples of N settings in BRR in normal asynchronous mode. Table 25.13 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 25.16. Examples of BRR (N) settings in smart card interface mode are listed in Table 25.18. Examples of BRR (N) settings in simple I^2C mode are listed in Table 25.20. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 25.6.4, Receive Data Sampling Timing and Reception Margin. Table 25.14 and Table 25.17 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 25.12. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 25.12 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			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Note: This is an example when the ABCS and BGDM bits in SEMR are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in SEMR are set to 1, the bit rate increases four times.

Table 25.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

PCLK (MHz)	SEMR Settings BGDM Bit	SEMR Settings ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	SEMR Settings BGDM Bit	SEMR Settings ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	18	0	0	0	0	562500
		1	0	0	500000			1	0	0	1125000
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2250000
9.8304	0	0	0	0	307200	19.6608	0	0	0	0	614400
		1	0	0	614400			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2457600
10	0	0	0	0	312500	20	0	0	0	0	625000
		1	0	0	625000			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	2500000
12	0	0	0	0	375000	25	0	0	0	0	781250
		1	0	0	750000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	3125000
12.288	0	0	0	0	384000	30	0	0	0	0	937500
		1	0	0	768000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	3750000
14	0	0	0	0	437500	33	0	0	0	0	1031250
		1	0	0	875000			1	0	0	2062500
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	4125000
16	0	0	0	0	500000	40	0	0	0	0	1250000
		1	0	0	1000000			1	0	0	2500000
	1	0	0	0			1	0	0	0	
		1	0	0	2000000			1	0	0	5000000
17.2032	0	0	0	0	537600						
		1	0	0	1075200						
	1	0	0	0							
		1	0	0	2150400						

Table 25.14 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
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000

Table 25.15 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	TMR Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000
33	16.5	1031250	2062500
40	20	1250000	2500000

Table 25.16 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)															
	8		10		16		20		25		30		33		40	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	155
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19
1 M	0	1	—	—	0	3	0	4	—	—	—	—	—	—	0	9
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	3
5 M							0	0*1							0	1
7.5 M																

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is impossible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is $\frac{8}{9}$ times the bit rate.

Table 25.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000
33	5.5000	5.5000
40	6.6667	6.6667

Table 25.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	30
	10.7136	0	1	25
	13.00	0	1	8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	15.99
	20.00	0	2	6.66
	25.00	0	3	12.49
	30.00	0	3	5.01
	33.00	0	4	7.59
	40.00	0	5	–6.66

Table 25.19 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0

Table 25.20 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71

Table 25.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60

25.2.12 Modulation Duty Register (MDDR)

Address(es): SCI1.MDDR 0008 A032h, SCI5.MDDR 0008 A0B2h

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

MDDR corrects the bit rate adjusted by the BRR register.

When the BRME bit in SEMR is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 25.22.

The initial value of MDDR is FFh. Bit 7 in this register is fixed to 1.

The CPU can read the MDDR register, but this register is only writable when the TE and RE bits in SCR are 0.

Table 25.22 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode	SEMR Settings		BRR Setting	Error
	BGDM Bit	ABCS Bit		
Asynchronous multi-processor communication	0	0	$N = \dfrac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\dfrac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	0	1	$N = \dfrac{PCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\dfrac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	1	0		
	1	1	$N = \dfrac{PCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\dfrac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
Clock synchronous mode, simple SPI mode*1			$N = \dfrac{PCLK \times 10^6}{8 \times 2^{2n-1} \times (256/M) \times B} - 1$	
Smart card interface mode			$N = \dfrac{PCLK \times 10^6}{S \times 2^{2n+1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\dfrac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (256/M) \times (N+1)} - 1\} \times 100$
Simple I^2C*2			$N = \dfrac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

B: Bit rate (bps)

M: MDDR setting ($128 \leq MDDR \leq 256$)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

PCLK: Operating frequency (MHz)

n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 25.10 and Table 25.11, section 25.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C standard.

25.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	—	BRME	—	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from TMR (valid for SCI5 only) Available compare match output varies per SCI channel.	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W
b3	—	Reserved	This bit is 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^{2}C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the CKE[1] bit in SCR is 0 in asynchronous mode). 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.
The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal TMR.
Set the ACS0 bit to 0 in other than asynchronous mode.
For SCI5, the TMOn output (n = 0, 1) of TMR unit 0 can be set as the serial transfer base clock. Refer to Table 25.23 for details.
The ACS0 bit for SCI1 is reserved. The write value to this bit for SCI1 should be 0.

Table 25.23 Correspondence between SCI Channels and Compare Match Outputs

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1

Figure 25.3 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

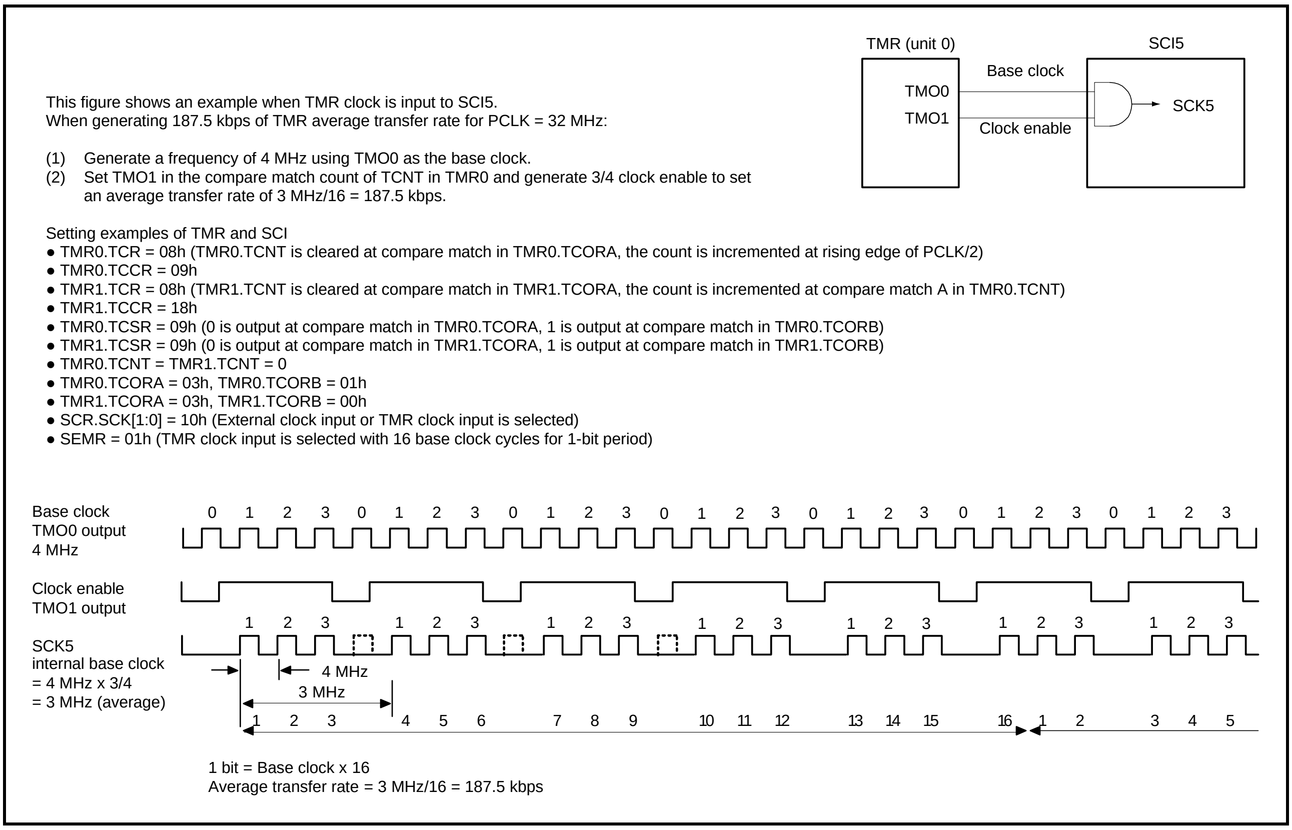


Figure 25.3 Example of Average Transfer Rate Setting When TMR Clock is Input

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

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^{2}C mode.

In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 0 in modes other than asynchronous mode.

25.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I²C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I^2C mode, set the bits to a value in the range from 001b to 100b.

25.2.15 I^{2}C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h

	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 I^{2}C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I^{2}C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I^{2}C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I^{2}C Mode Select)

In conjunction with the SMIF bit in the SCMR register, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I^{2}C mode. In simple I^{2}C mode, set the bits to a value in the range from 00001b to 11111b.

25.2.16 I2C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I2C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I2C mode.

IICINTM Bit (I2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I2C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

25.2.17 I²C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTP REQ	IICRST AREQ	IICSTA REQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Only generate a start condition after checking the bus state and confirming that it is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that it is busy.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I²C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.

Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

25.2.18 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

SISR is used to monitor state in relation to simple I²C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.
The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

25.2.19 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the TXDn pin and reception is through the RXDn pin (master mode). 1: Reception is through the TXDn pin and transmission is through the RXDn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).

Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin.

Set this bit to 0 in modes other than simple SPI mode.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.
In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. See Figure 25.55 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. See Figure 25.55 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

25.3 Operation in Asynchronous Mode

Figure 25.4 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

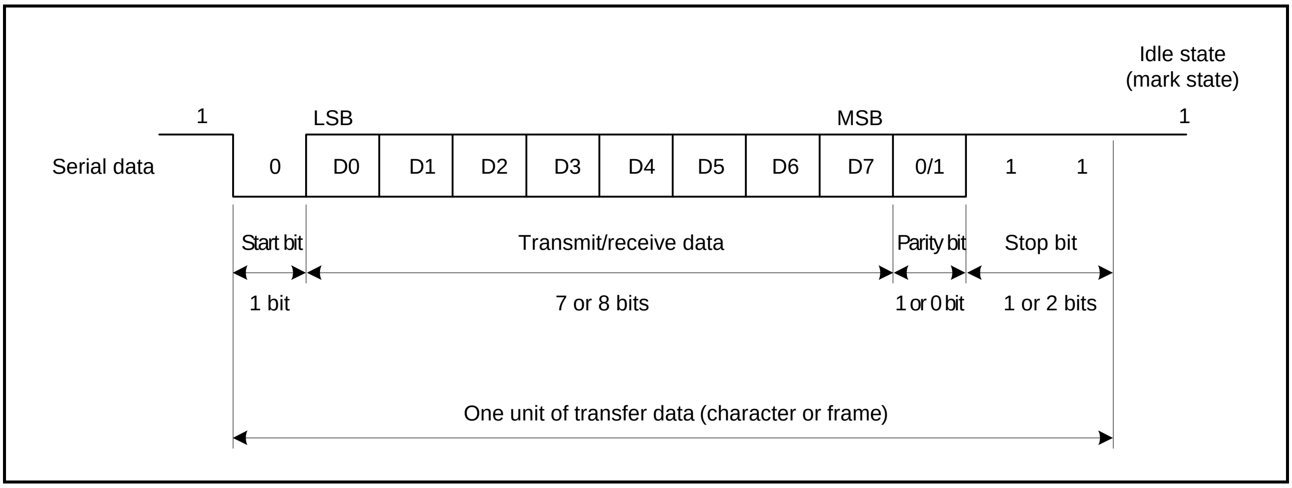


Figure 25.4 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

25.3.1 Serial Data Transfer Format

Table 25.24 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 25.4, Multi-Processor Communications Function.

Table 25.24 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S, 9-bit data, STOP
0	0	0	0	1	S, 9-bit data, STOP, STOP
0	0	1	0	0	S, 9-bit data, P, STOP
0	0	1	0	1	S, 9-bit data, P, STOP, STOP
1	0	0	0	0	S, 8-bit data, STOP
1	0	0	0	1	S, 8-bit data, STOP, STOP
1	0	1	0	0	S, 8-bit data, P, STOP
1	0	1	0	1	S, 8-bit data, P, STOP, STOP
1	1	0	0	0	S, 7-bit data, STOP
1	1	0	0	1	S, 7-bit data, STOP, STOP
1	1	1	0	0	S, 7-bit data, P, STOP
1	1	1	0	1	S, 7-bit data, P, STOP, STOP
0	0	—	1	0	S, 9-bit data, MPB, STOP
0	0	—	1	1	S, 9-bit data, MPB, STOP, STOP
1	0	—	1	0	S, 8-bit data, MPB, STOP
1	0	—	1	1	S, 8-bit data, MPB, STOP, STOP
1	1	—	1	0	S, 7-bit data, MPB, STOP
1	1	—	1	1	S, 7-bit data, MPB, STOP, STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

25.3.2 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 25.5. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\ F - \frac{\left| D - 0.5 \right|}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{ Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCS bit in the SEMR register 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.

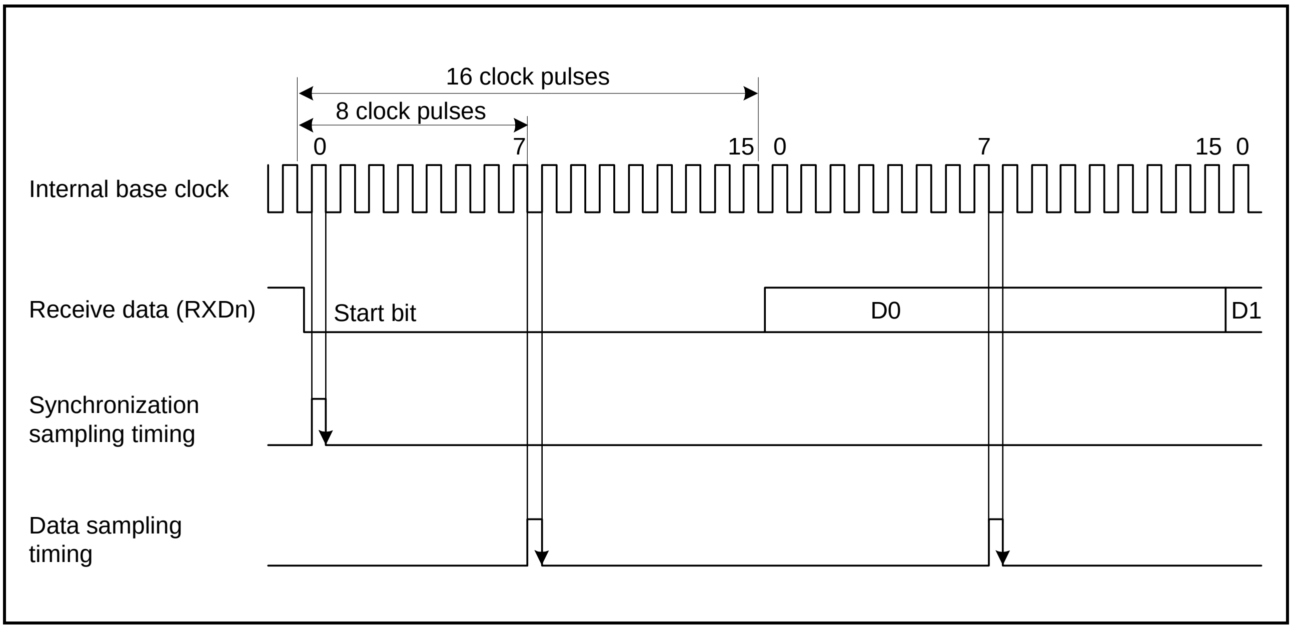


Figure 25.5 Receive Data Sampling Timing in Asynchronous Mode

25.3.3 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 the SMR register and the CKE[1:0] bits in the SCR register.

When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of TMR0 can be selected by the SCIn.SEMR.ACS0 bit (n = 5).

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 25.6.

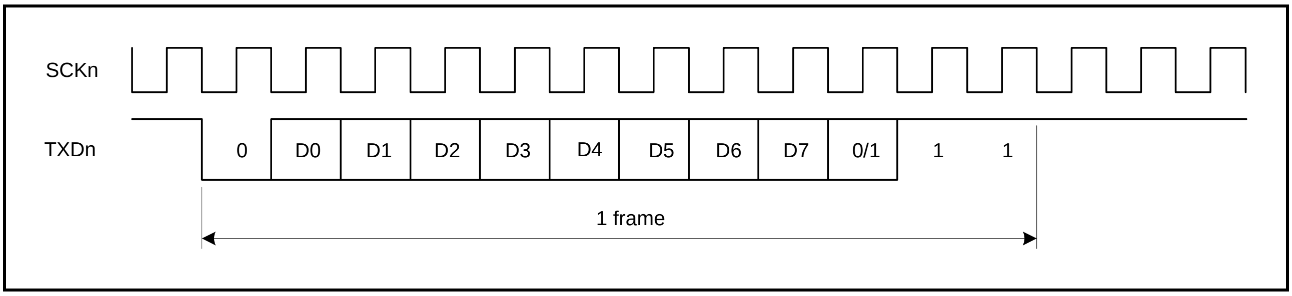


Figure 25.6 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

25.3.4 Double-Speed Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.

As shown by Formula (1) in section 25.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

25.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied

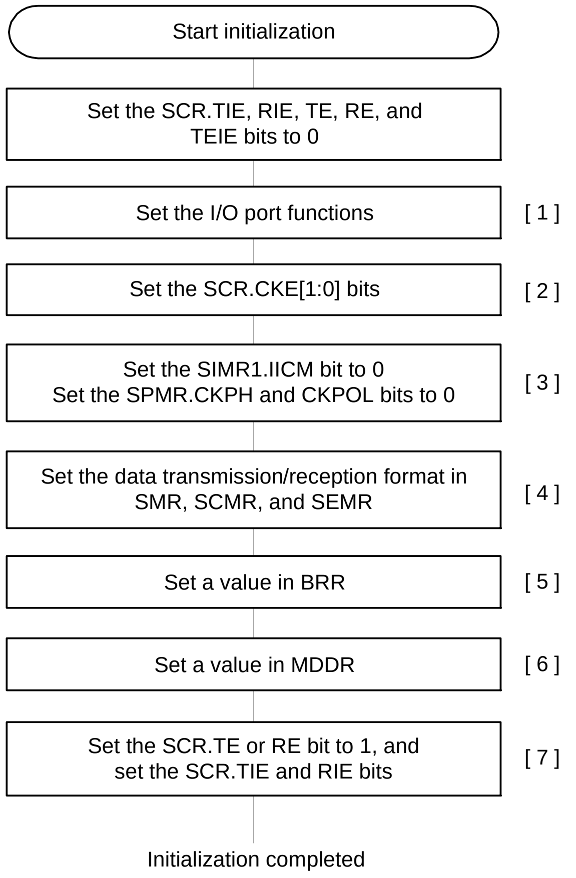
- The SCR.RE bit is 1
- Reception is not in progress
- There are no received data yet to be read
- The ORER, FER, and PER flags in the SSR are all 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

Note that either one of CTS and RTS can be selected.

25.3.6 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 25.7. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.
When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.
Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL.
Moreover, note that changing 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 transmit data empty interrupt (TXI) request.
In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a TXI interrupt request.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in SCR.
When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits to 0.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value corresponding to the bit rate to BRR.
This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 25.7 Sample SCI Initialization Flowchart (Asynchronous Mode)

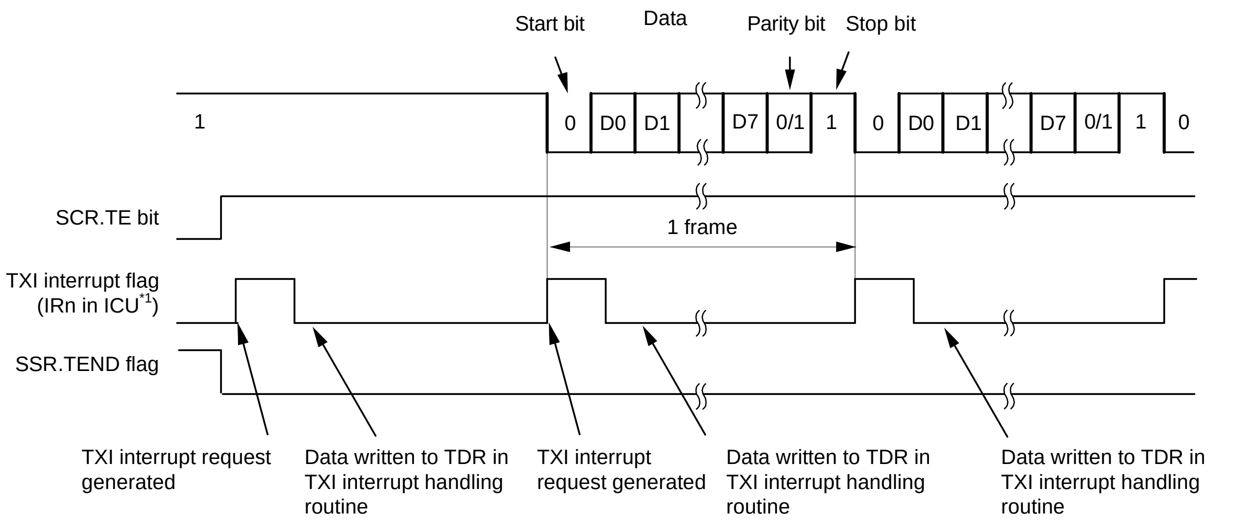
25.3.7 Serial Data Transmission (Asynchronous Mode)

Figure 25.8 to Figure 25.10 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

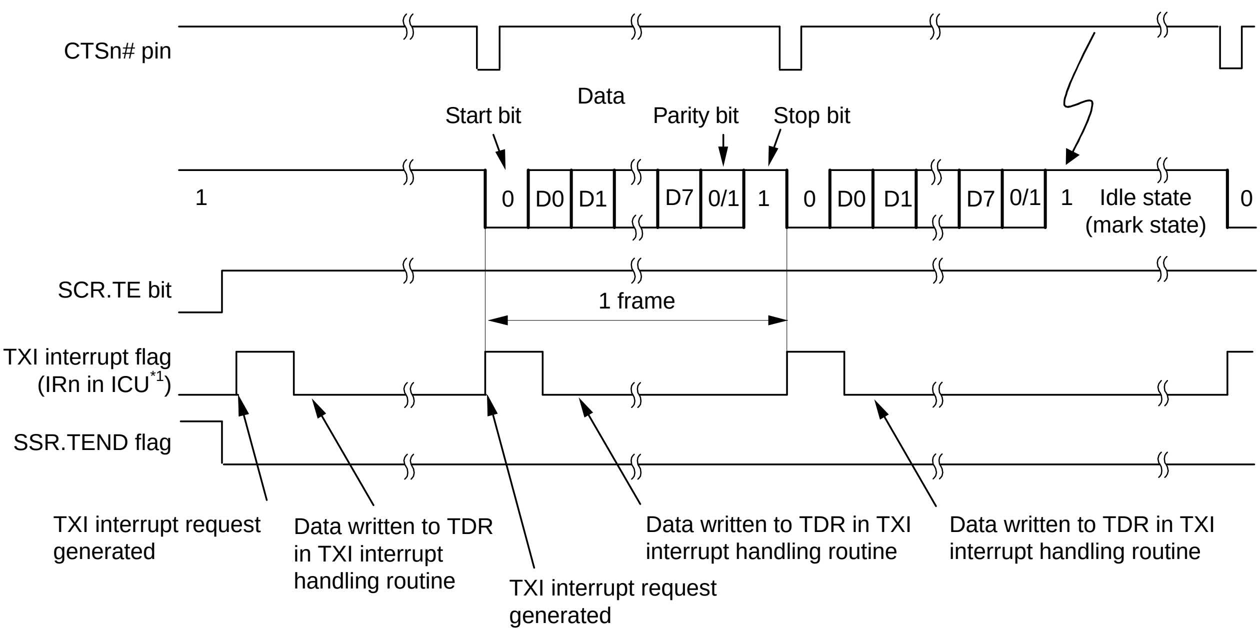
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 25.11 shows a sample flowchart for serial transmission in asynchronous mode.



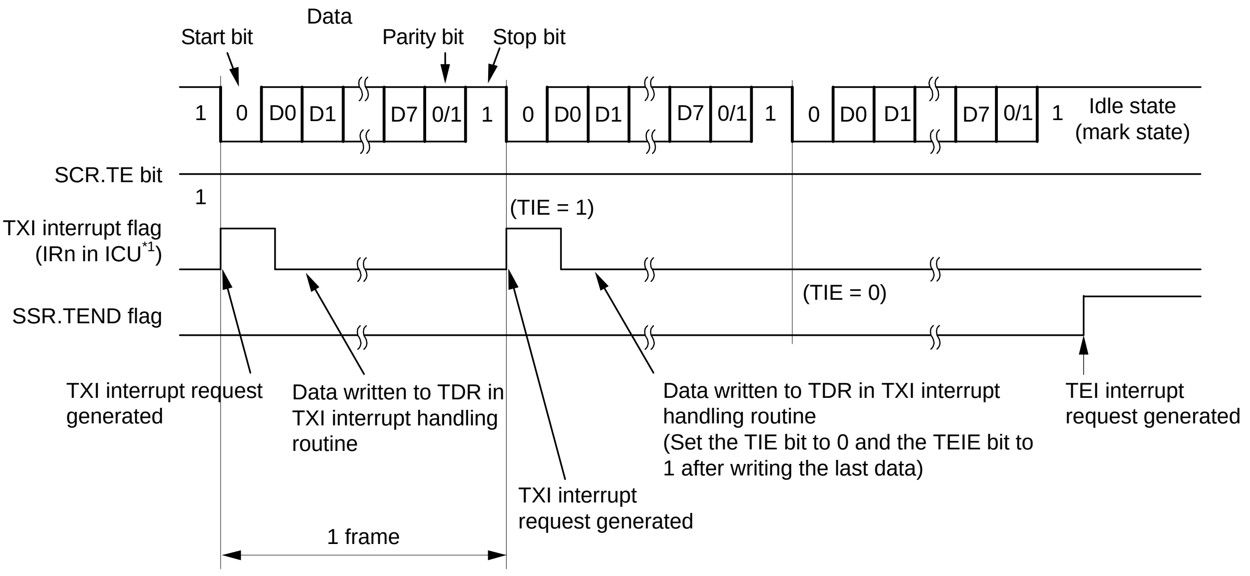
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.8 Example of Operation for Serial Transmission in Asynchronous Mode (1) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



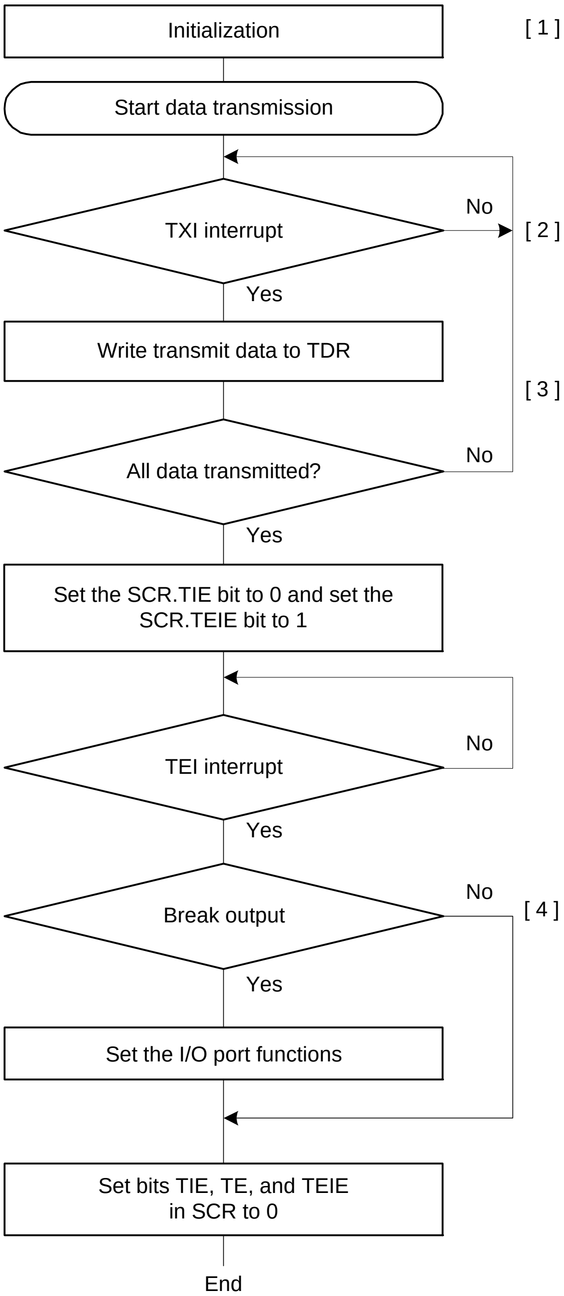
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.9 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.10 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 25.11 Example of Serial Transmission Flowchart in Asynchronous Mode

25.3.8 Serial Data Reception (Asynchronous Mode)

Figure 25.12 and Figure 25.13 show an example of the operation for serial data reception in asynchronous mode.
In serial data reception, the SCI operates as described below.

1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.

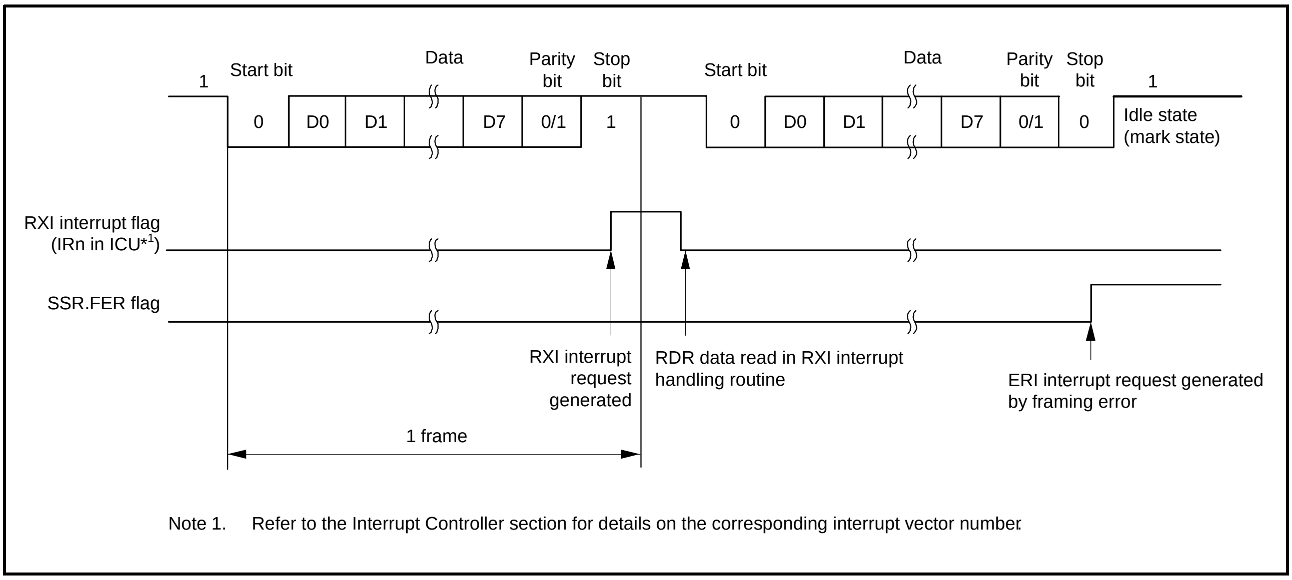


Figure 25.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

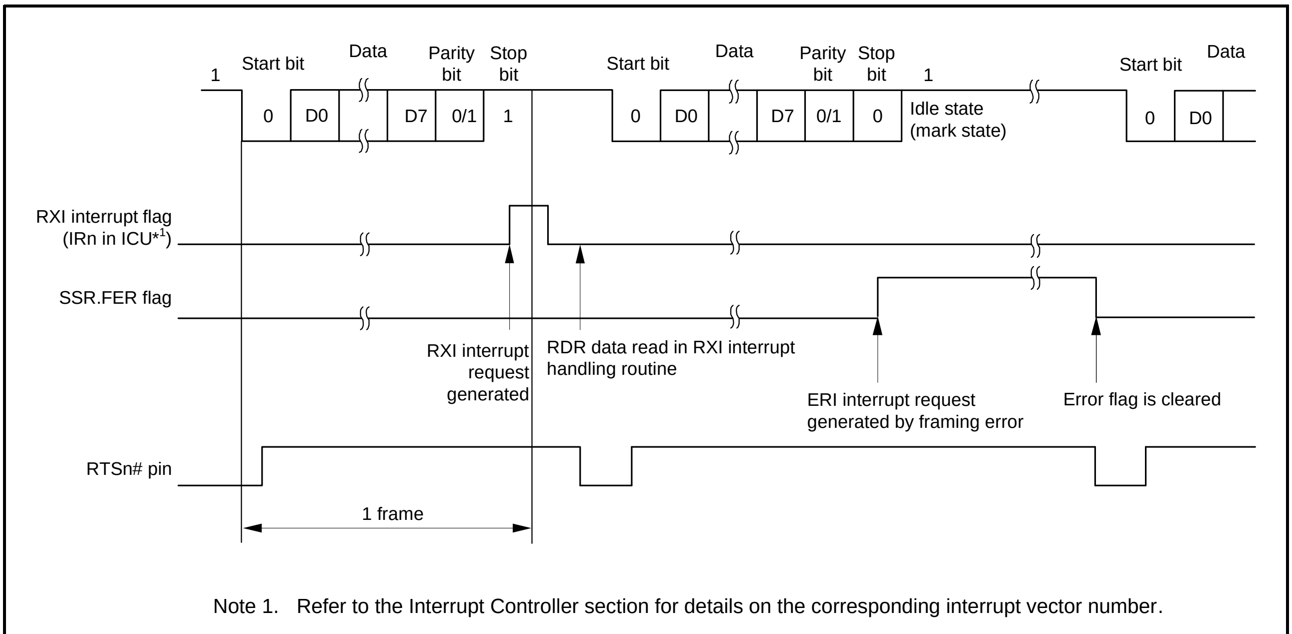


Figure 25.13 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 25.25 lists the states of the flags in the SSR status register and receive data handling when a receive error is detected.

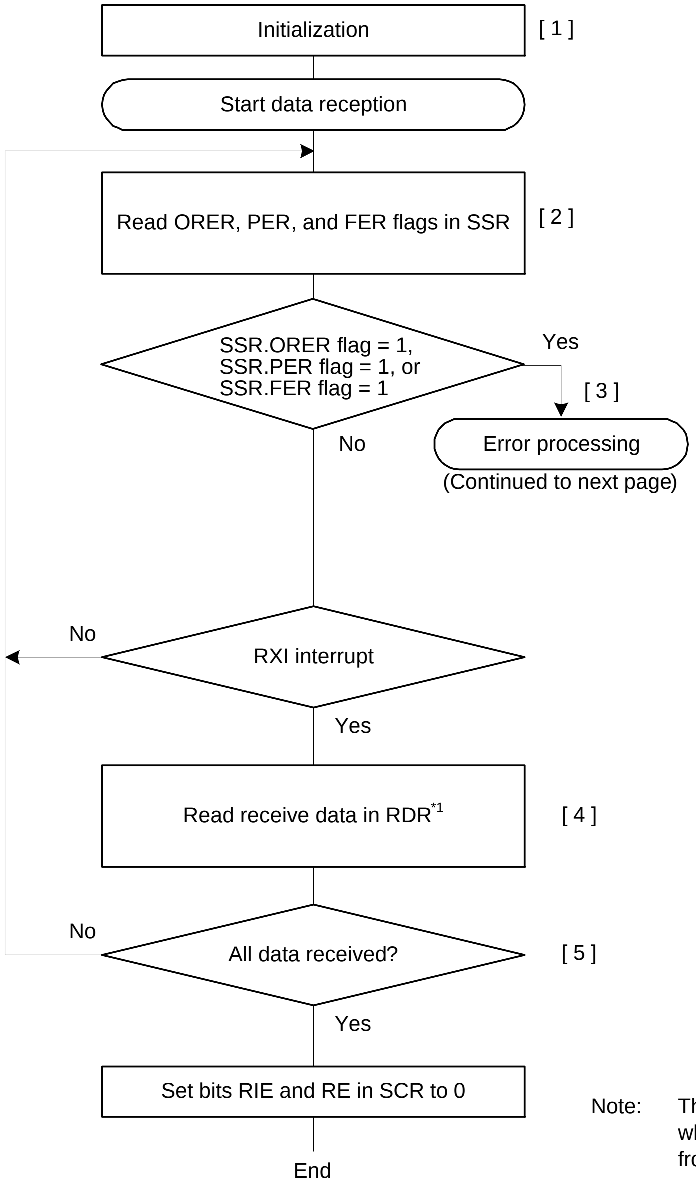
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in RDR (or the RDRL).

Figure 25.14 and Figure 25.15 show samples of flowcharts for serial data reception.

Table 25.25 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	FER	PER	Receive Data	Receive Error Type
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR*1	Framing error
0	0	1	Transferred to RDR*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in RDR but in the RDRH and RDRL registers when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in RDR once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt processing routine. The RDR data can also be read by activating the DTC.

Note: The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 25.14 Example Flowchart of Serial Reception in Asynchronous Mode (1)

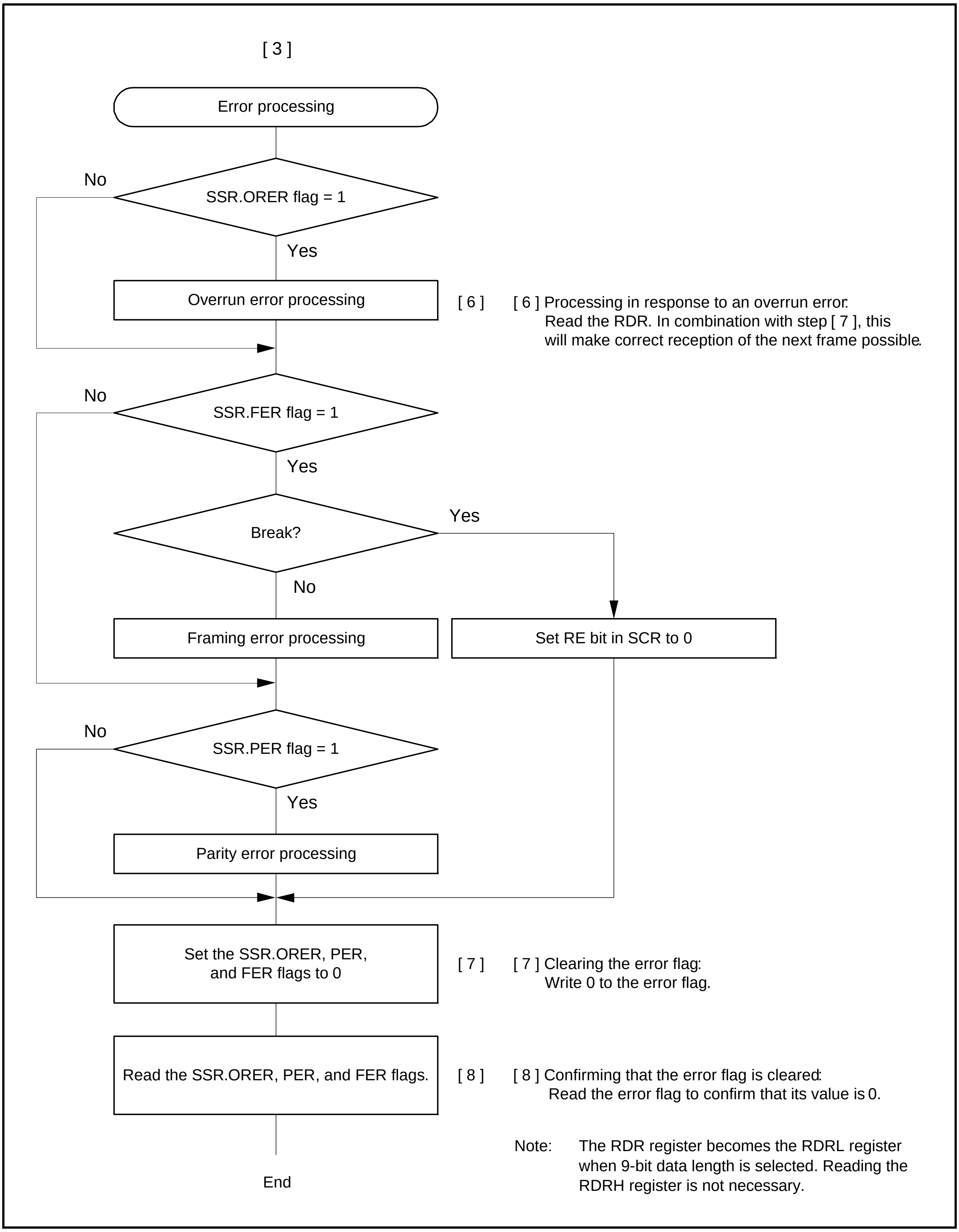


Figure 25.15 Example Flowchart of Serial Reception in Asynchronous Mode (2)

25.4 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 25.16 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags ORER and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

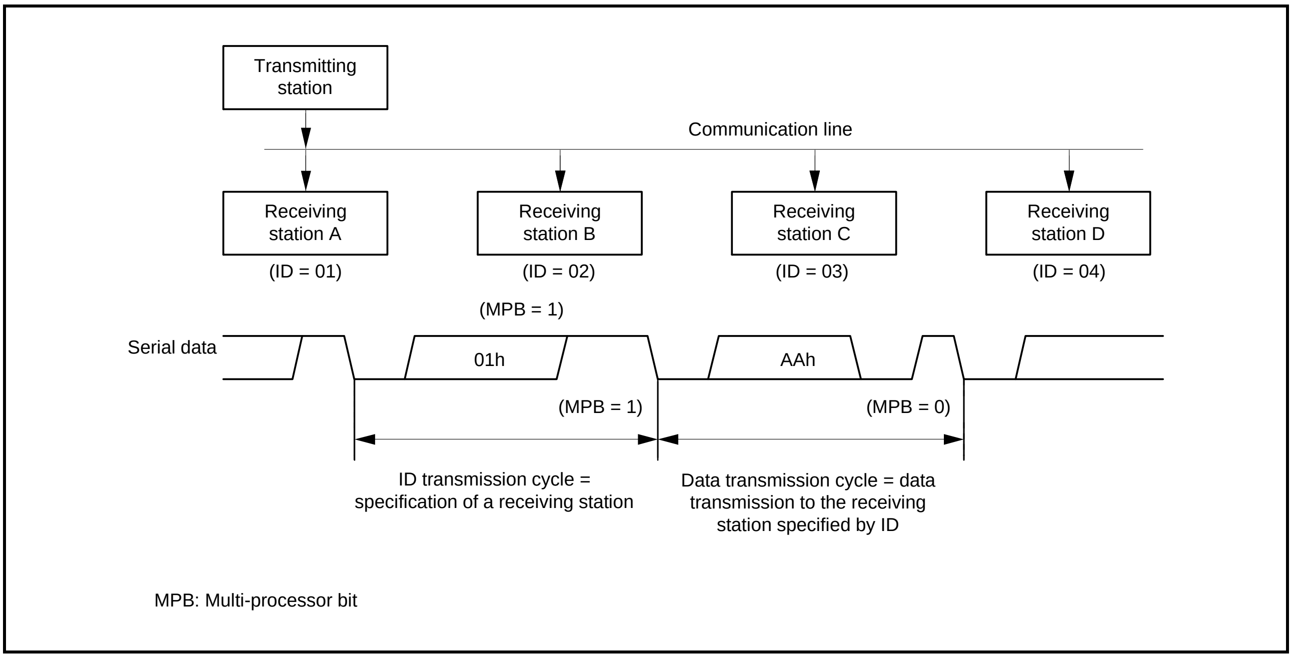
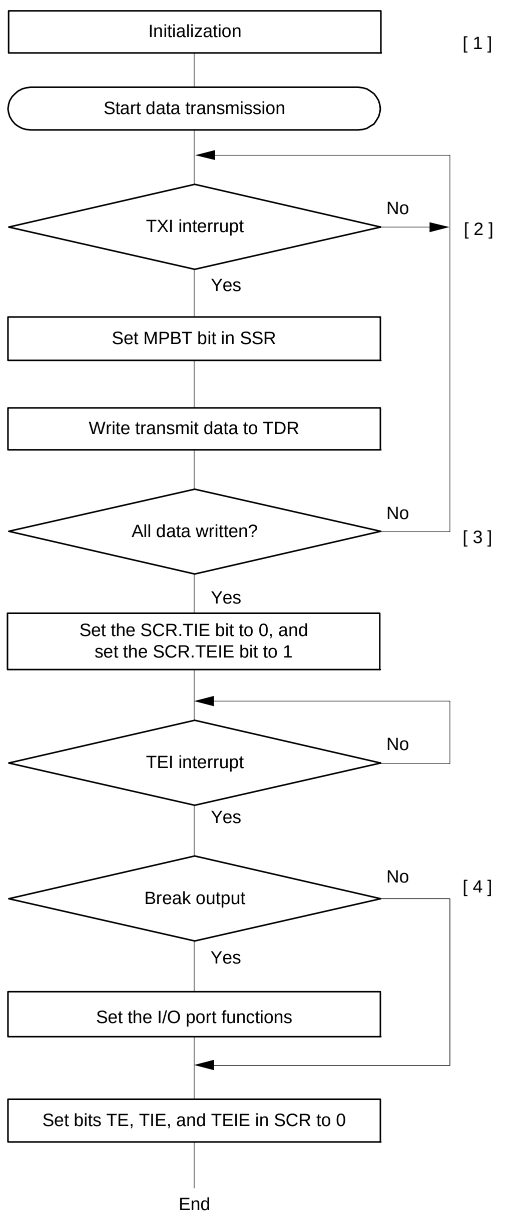


Figure 25.16 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

25.4.1 Multi-Processor Serial Data Transmission

Figure 25.17 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

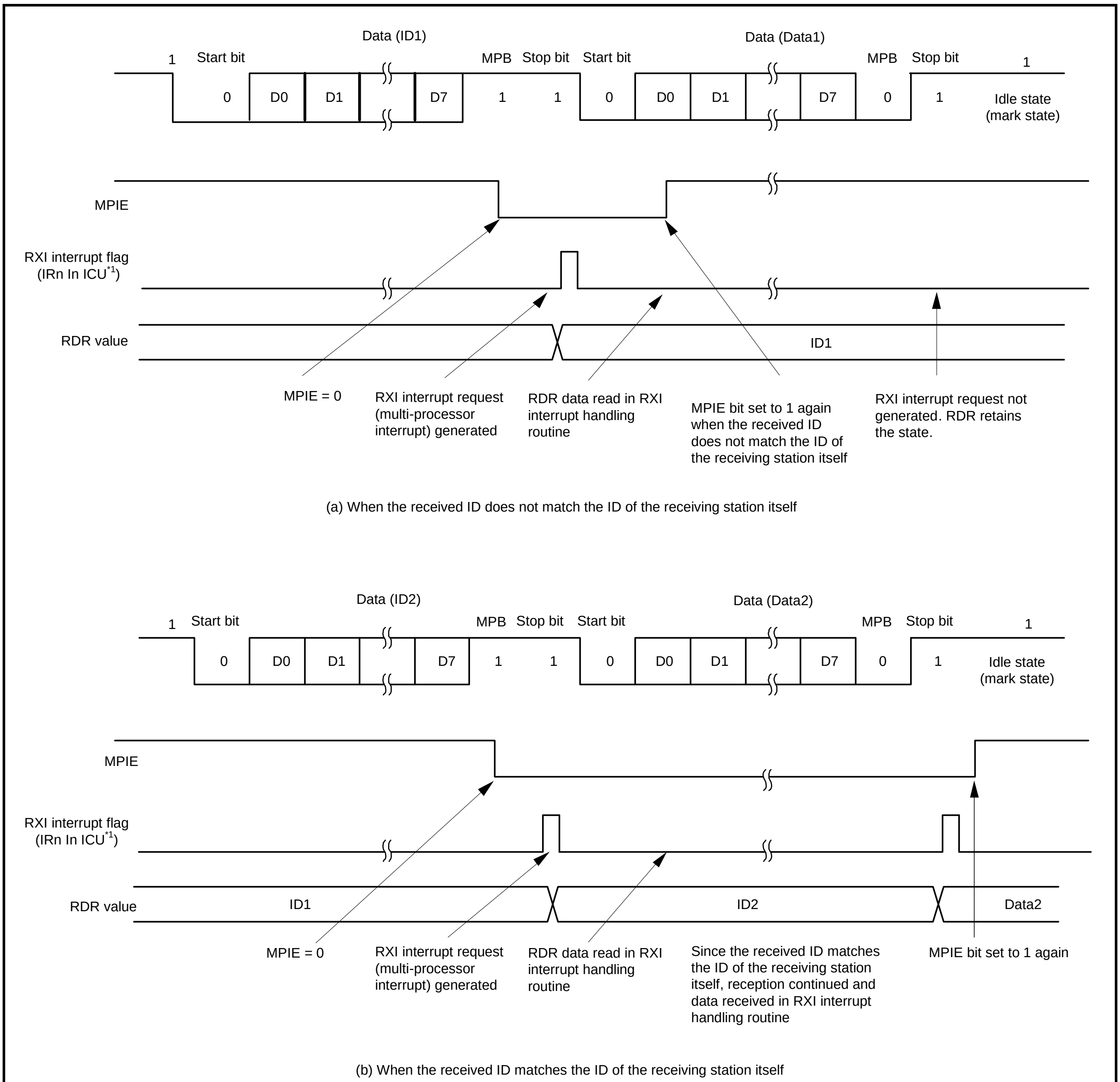
[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 25.17 Example of Multi-Processor Serial Transmission Flowchart

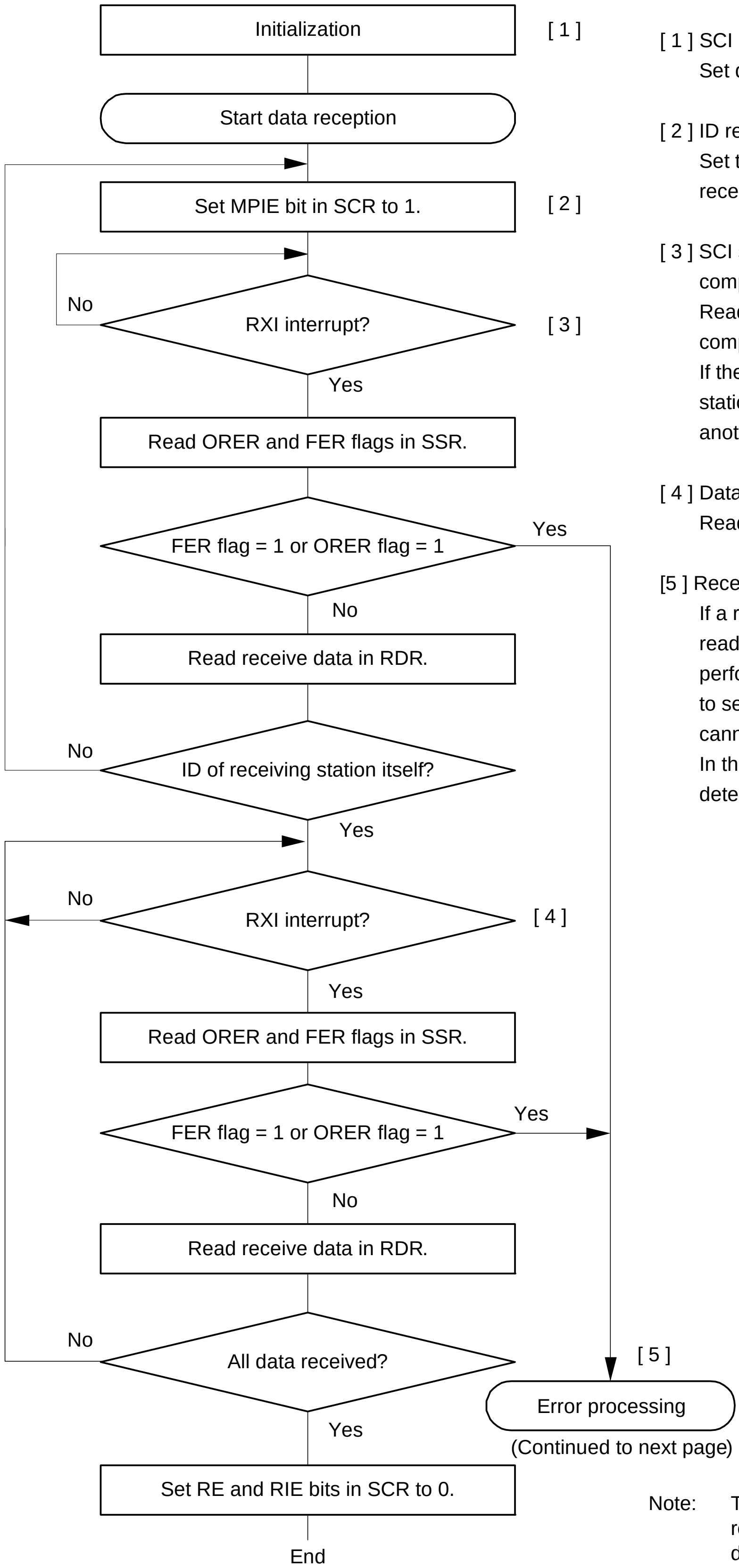
25.4.2 Multi-Processor Serial Data Reception

Figure 25.19 and Figure 25.20 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 25.18 is the example of operation for reception.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.18 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



(Continued to next page)

[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first RXI interrupt, and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in RDR once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER 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 RXDn pin.

Note: The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 25.19 Example of Multi-Processor Serial Reception Flowchart (1)

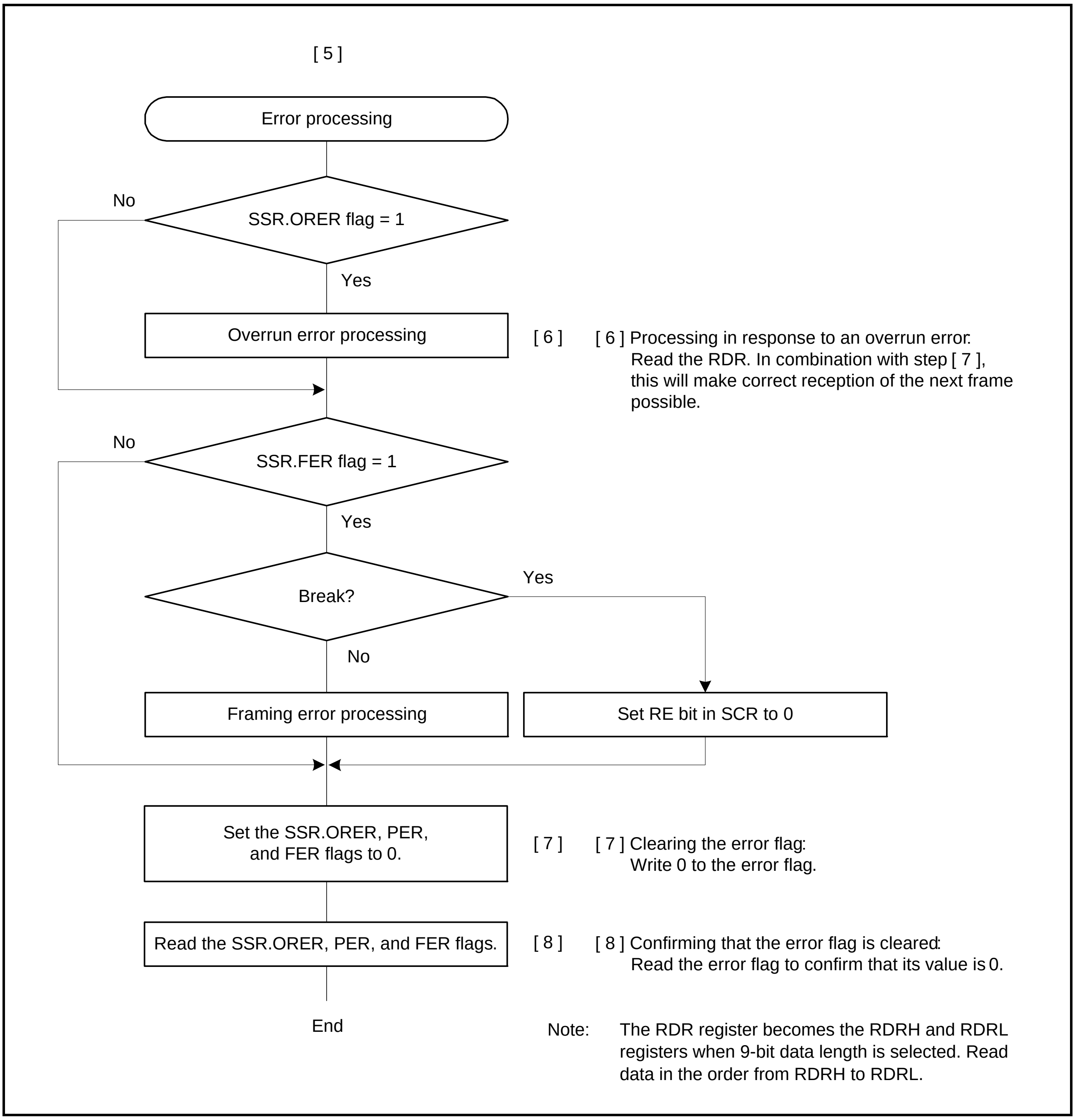
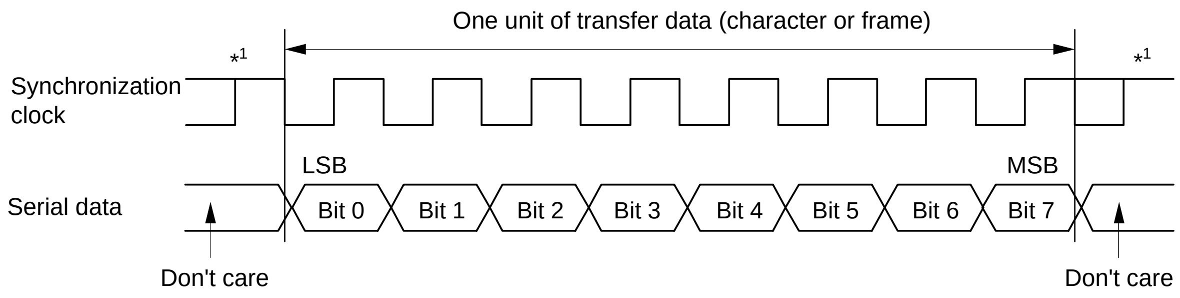


Figure 25.20 Example of Multi-Processor Serial Reception Flowchart (2)

25.5 Operation in Clock Synchronous Mode

Figure 25.21 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



Note 1. Holds a high level except during continuous transfer.

Figure 25.21 Data Format in Clock Synchronous Serial Communications (LSB First)

25.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

25.5.2 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]
When the following conditions are all satisfied

- The SCR.RE or SCR.TE bit is 1
- Transmission or reception of data is not 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)
- The SSR.ORER flag is 0

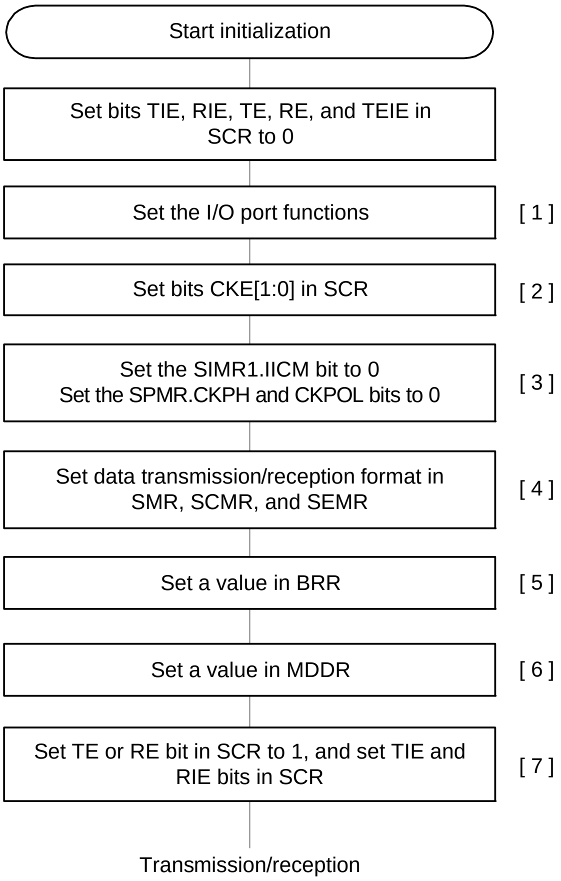
[Condition for high-level output]
The conditions for low-level output have not been satisfied.

25.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 25.22. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.

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.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in SCR.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits to 0.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set the data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value corresponding to the bit rate to BRR.
This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is cleared to 0 or an external clock is used.

[7] Set the TE bit or RE bit in SCR to 1. Also set the TIE and RIE bits in SCR.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR should both be set to 0 or 1 simultaneously.

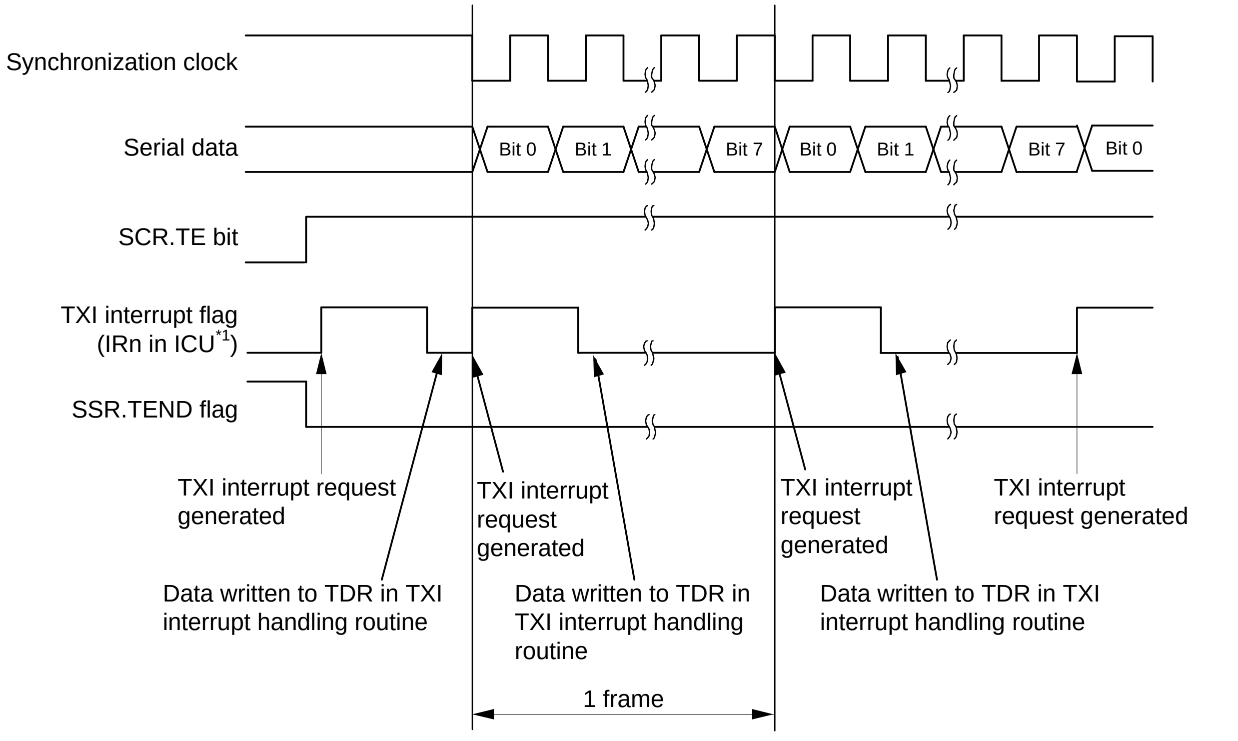
Figure 25.22 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

25.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 25.22, Figure 25.23, and Figure 25.24 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

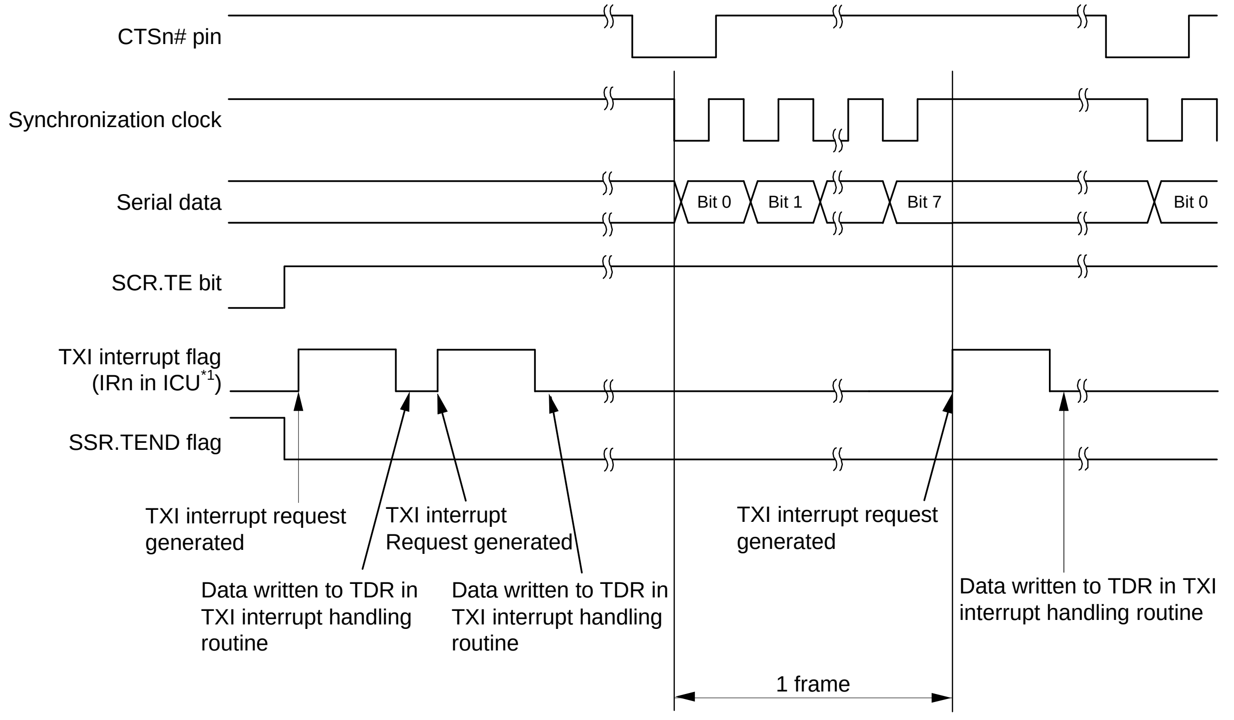
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in the SCR register is set to 1 after the TIE bit in the SCR register is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in the SPMR register is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in the SCR register is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 25.26 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the RE bit in the SCR register to 0 does not clear the receive error flags.



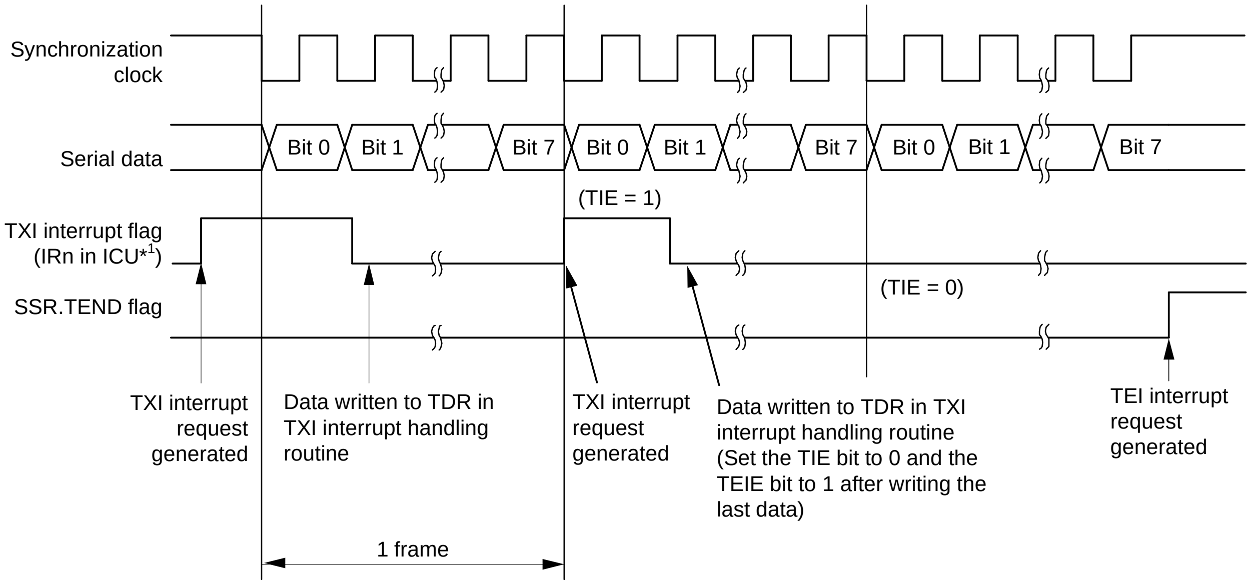
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.23 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



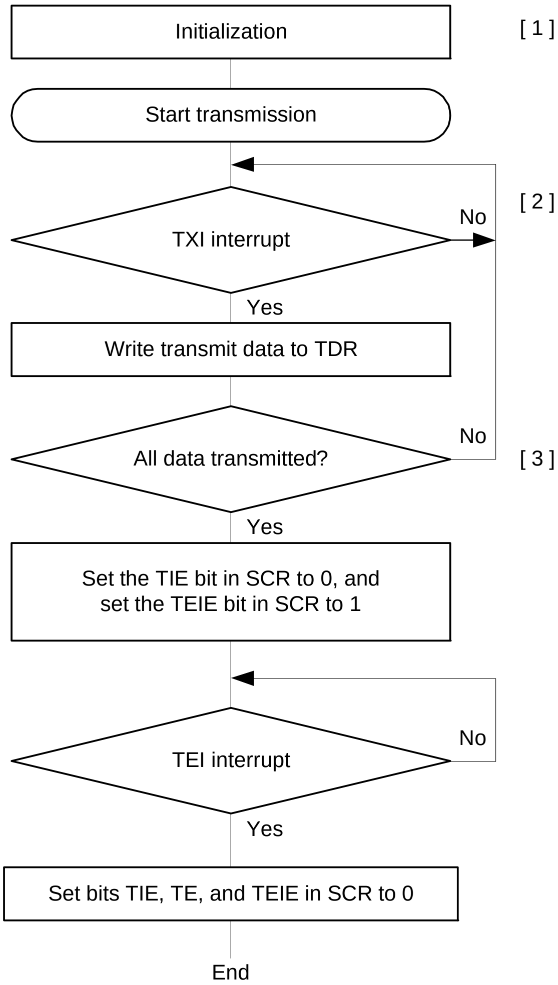
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.24 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 25.25 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to TDR by activating the DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

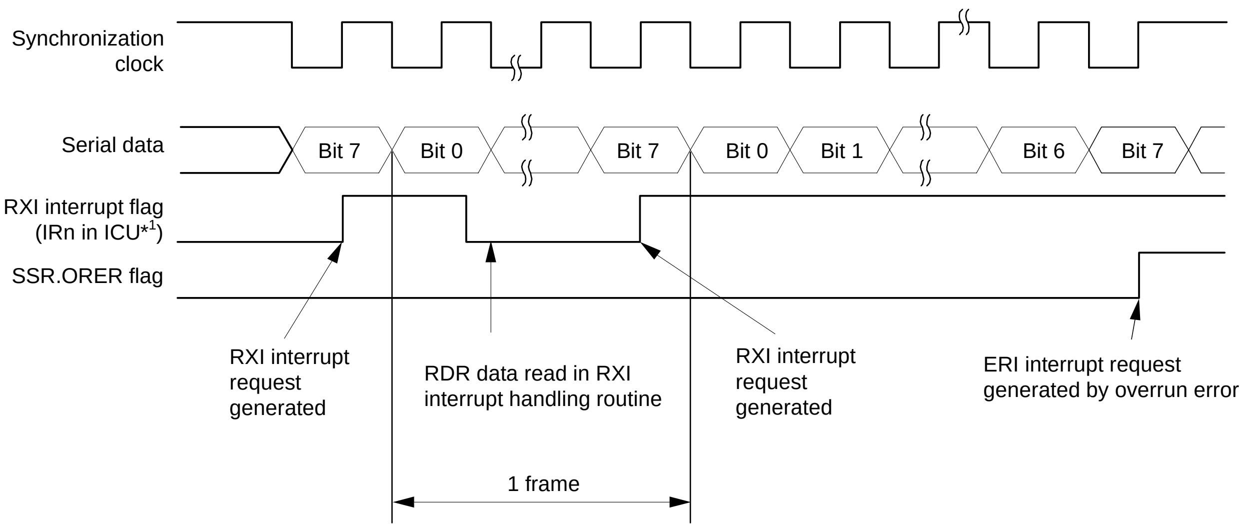
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 25.26 Example Flowchart of Serial Transmission in Clock Synchronous Mode

25.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 25.27 and Figure 25.28 show an example of SCI operation for serial reception in clock synchronous mode.
In serial data reception, the SCI operates as described below.

1. The value of the RE bit in the SCR register becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the ORER bit in the SSR register is set to 1. If the RIE bit in the SCR register is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the RIE bit in the SCR register is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to the Interrupt Controller section for the corresponding interrupt vector number.

Figure 25.27 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

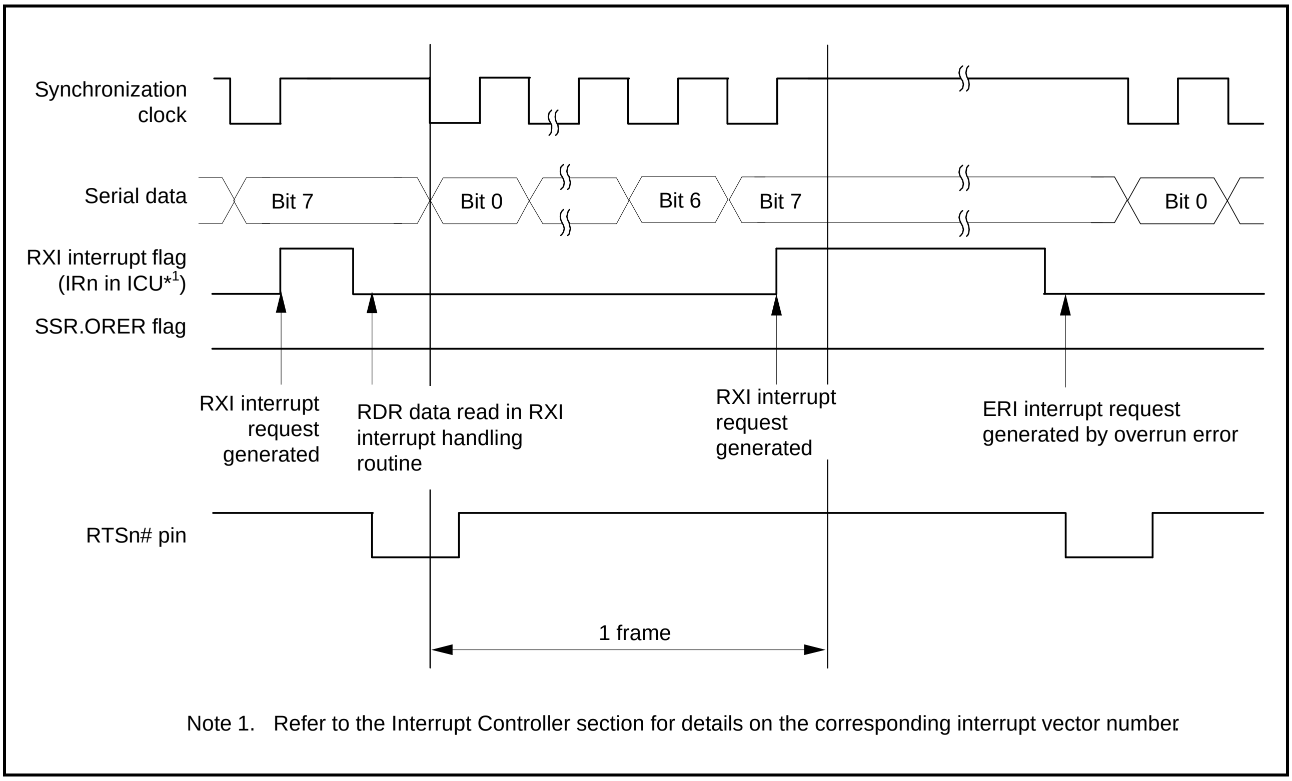
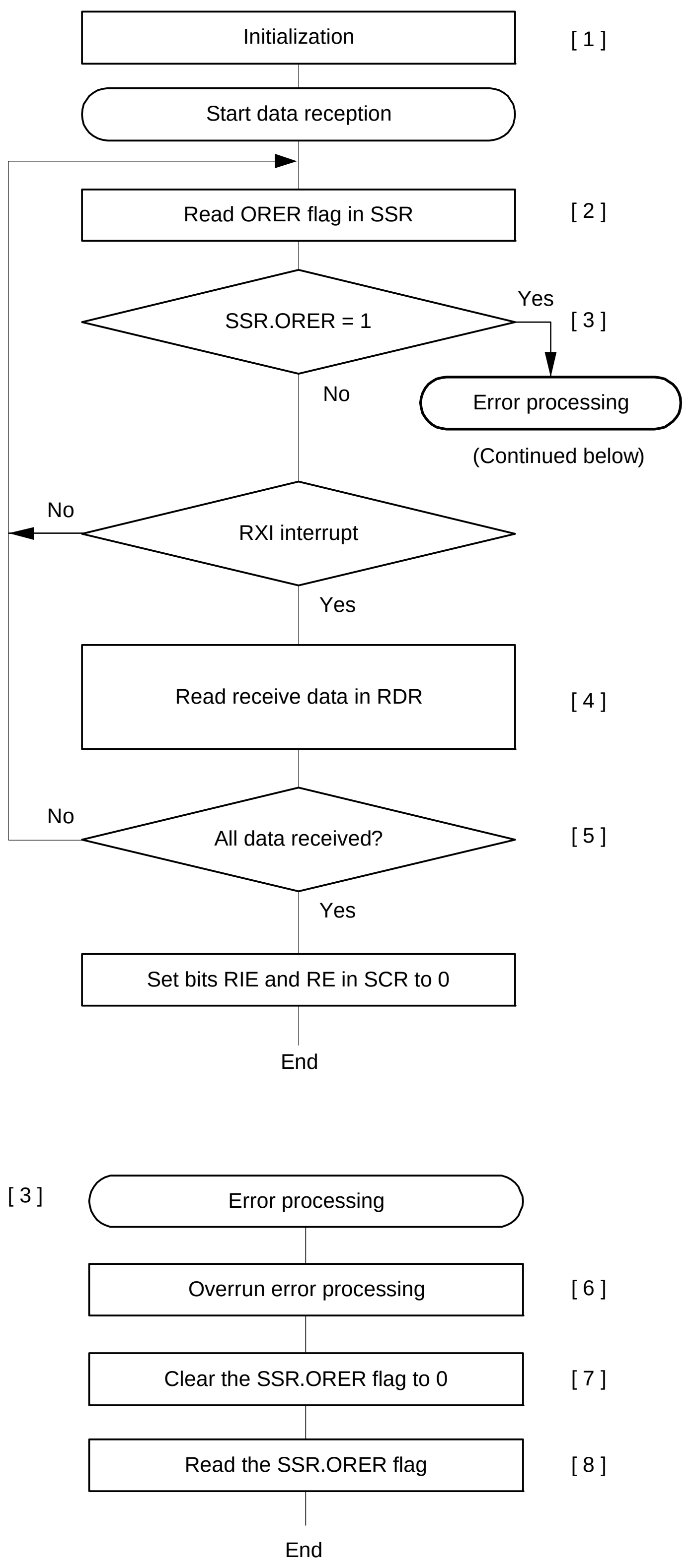


Figure 25.28 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in the SSR register to 0 before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 25.29 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 the SSR register, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 25.29 Example Flowchart of Serial Reception in Clock Synchronous Mode

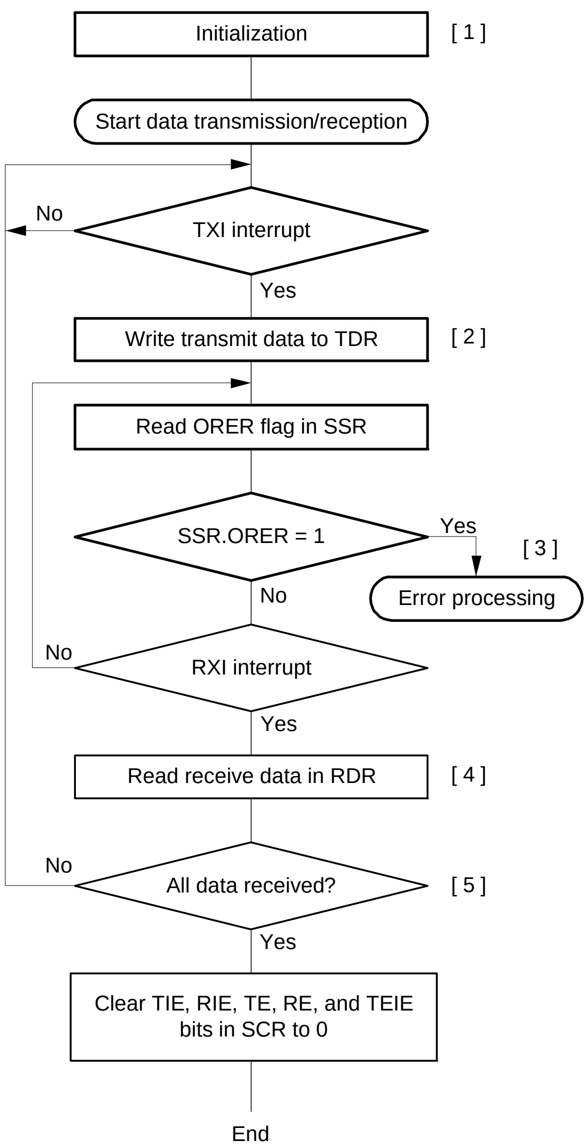
25.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 25.30 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 the SSR register is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in the SSR register, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 25.30 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

25.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

25.6.1 Sample Connection

Figure 25.31 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

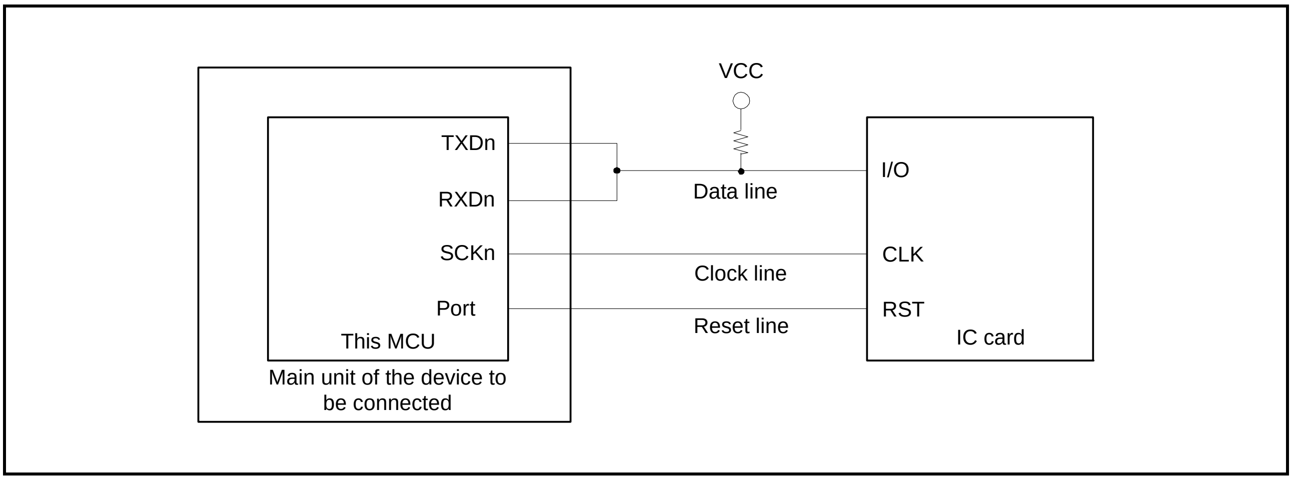


Figure 25.31 Sample Connection with a Smart Card (IC Card)

25.6.2 Data Format (Except in Block Transfer Mode)

Figure 25.32 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

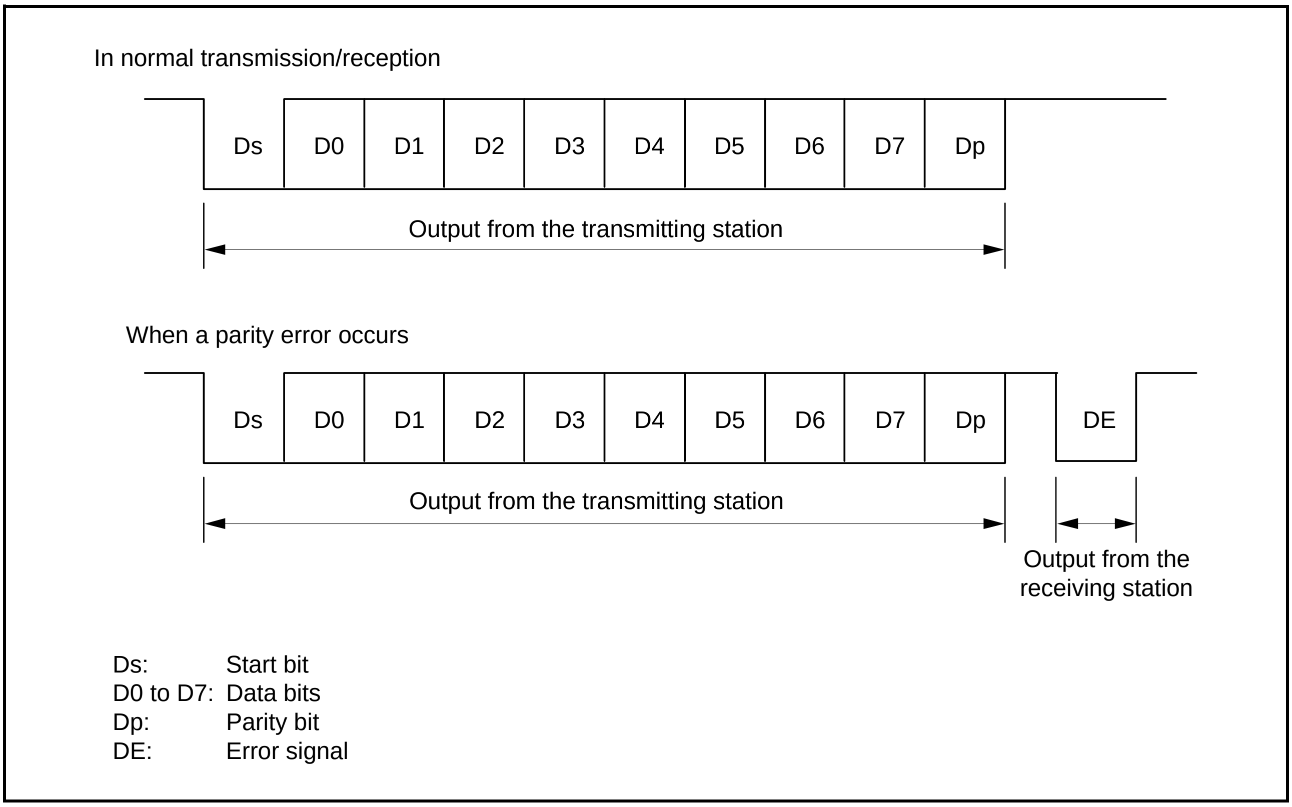


Figure 25.32 Data Formats in Smart Card Interface Mode

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 25.33. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the PM bit in the SMR register in order to use even parity, which is prescribed by the smart card standard.

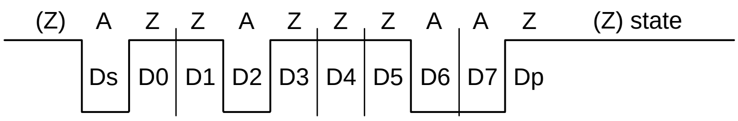


Figure 25.33 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

(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 25.34. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in the SMR register to invert the parity bit for both transmission and reception.

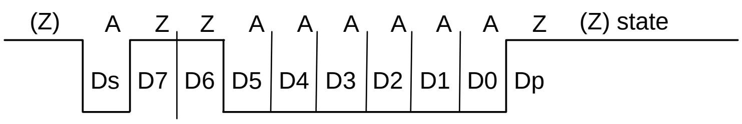


Figure 25.34 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

25.6.3 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 the SSR register is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in the SSR register is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in the SSR register indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred.

25.6.4 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 the SCMR register and the BCP[1:0] bits in the SMR register (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 25.35. 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.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

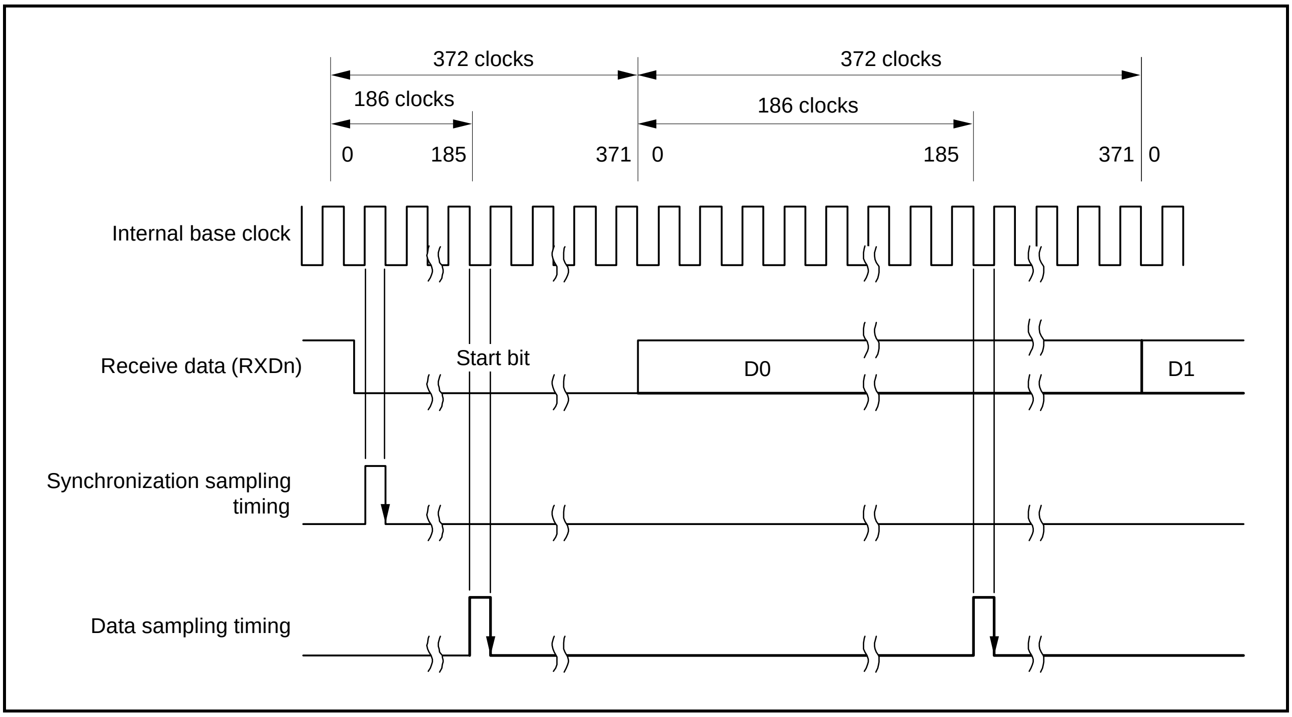
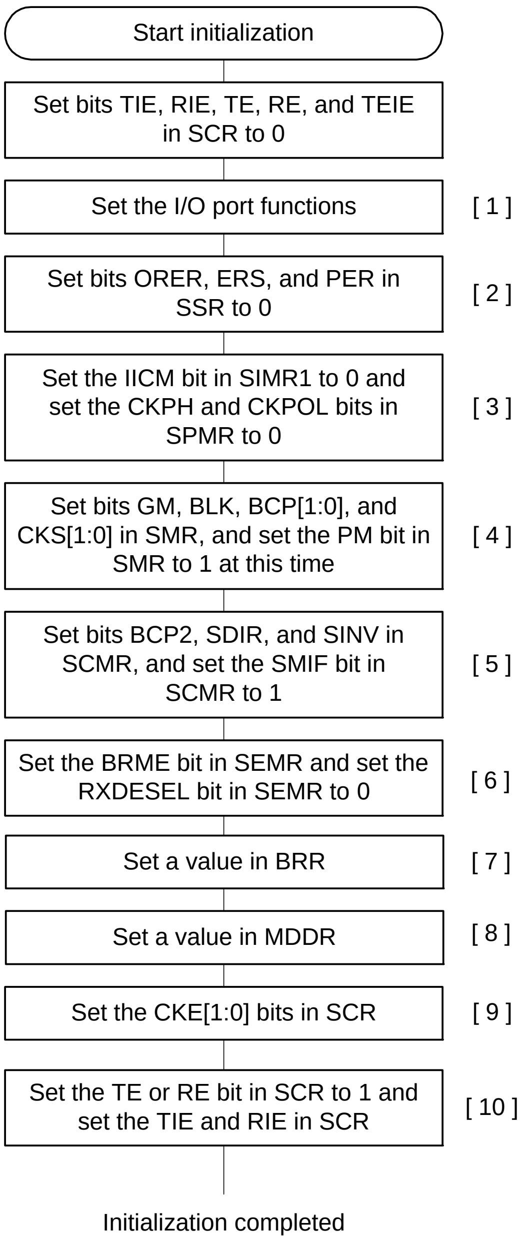


Figure 25.35 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

25.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 25.36.
Be sure to initialize the SCI before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in the SSR register.
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 the SSR register.



[1] Set the I/O ports so that the necessary pin functions can be enabled among the TXDn, RXDn, and SCKn pins.

[2] Set the ORER, ERS, and PER bits in SSR to 0.
After reading SSR, write 0 in the target bits.

[3] Set the IICM bit in SIMR1 to 0 and set the CKPH and CKPOL bits in SPMR to 0.
Skip this step when the initial values are not changed.

[4] Set the transmission/reception format in SMR.

[5] Set the transmission/reception format in SCMR.
The TXDn and RXDn pins enter the high-impedance state.

[6] Set the BRME bit in SEMR and set the RXDESEL bit in SEMR to 0.

[7] Write the value corresponding to the bit rate in BRR.

[8] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the bit rate modulation function is not used.

[9] Set the CKE[1:0] bits in SCR.
When the CKE[0] bit is set to 1, the clock is output from the SCKn pin.

[10] Set the TE or RE bit in SCR to 1 and set the TIE, and RIE bits in SCR.
Do not simultaneously set the TE and RE bits to 1 if self-diagnosis is not used.

Figure 25.36 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

25.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 25.37 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 the SSR register is set to 1. If the RIE bit in the SCR register is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in the SSR register is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If the TIE bit in the SCR register is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Figure 25.39 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC.
When the TEND flag in the SSR register is set to 1 in transmission, if the TIE bit in the SCR register is 1, a TXI interrupt request is generated. The DTC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC transfers the data.
If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC 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, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 17, Data Transfer Controller (DTCa).

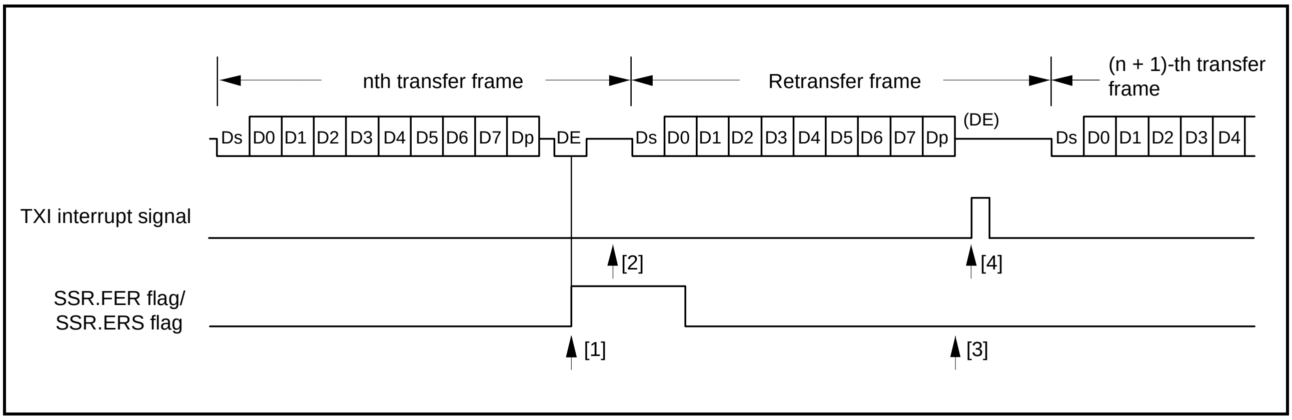


Figure 25.37 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in the SMR register. Figure 25.38 shows the TEND flag generation timing.

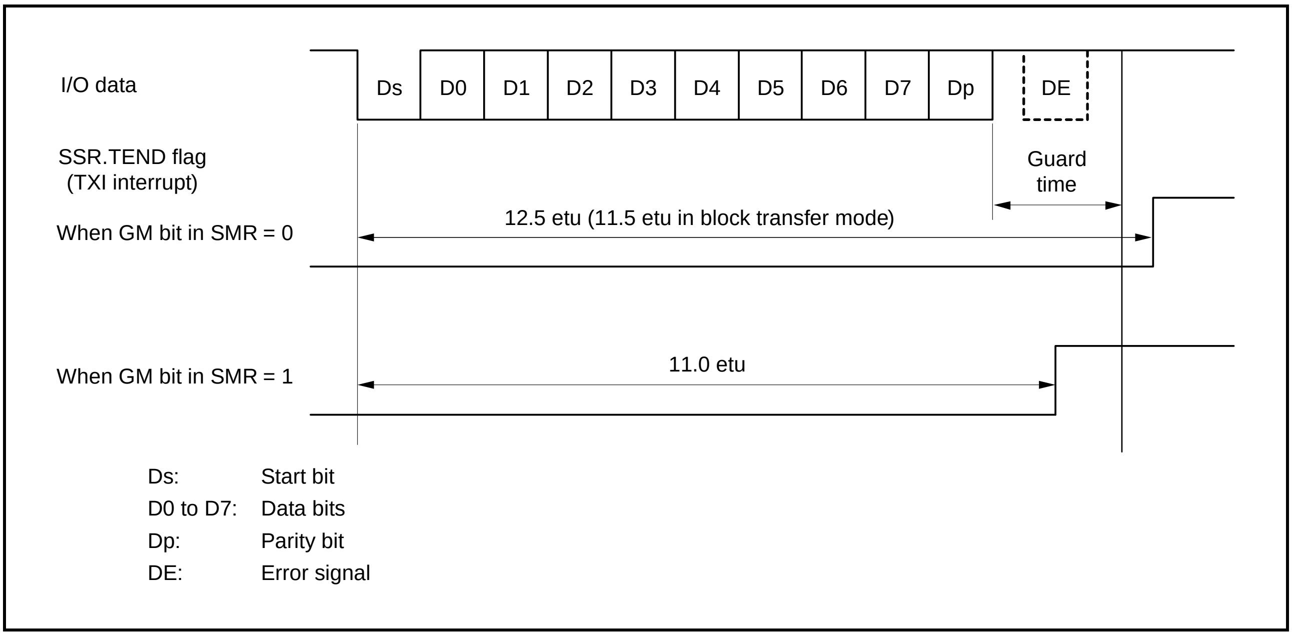


Figure 25.38 SSR.TEND Flag Generation Timing during Transmission

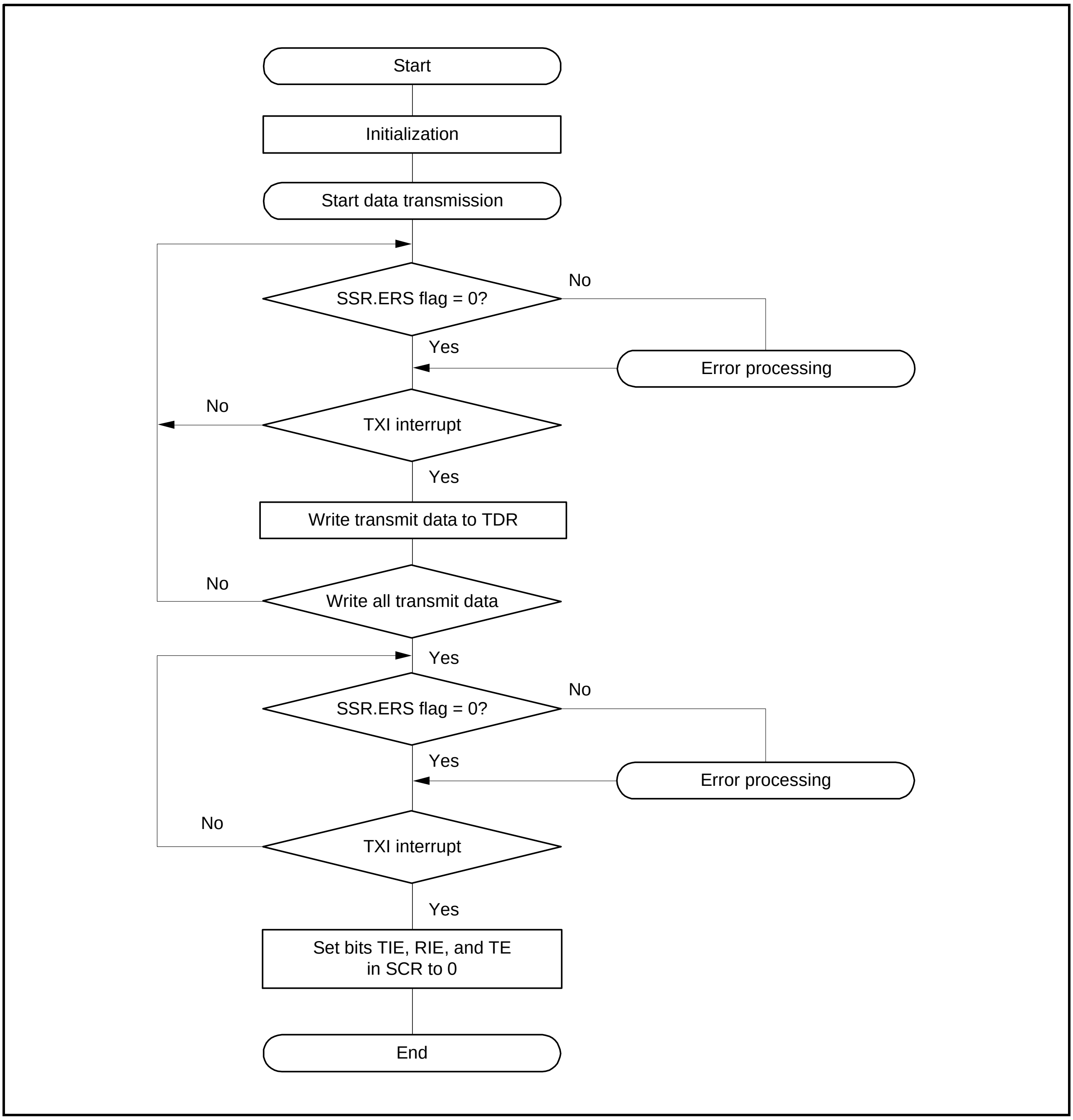


Figure 25.39 Sample Smart Card Interface Transmission Flowchart

25.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 25.40 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in the SSR register is set to 1. When the RIE bit in the SCR register 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 the SSR register is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in the SCR register is 1, an RXI interrupt request is generated.

Figure 25.41 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC.

In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in the SSR register is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC 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.

When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR.

Note 1. For operations in block transfer mode, refer to section 25.3, Operation in Asynchronous Mode.

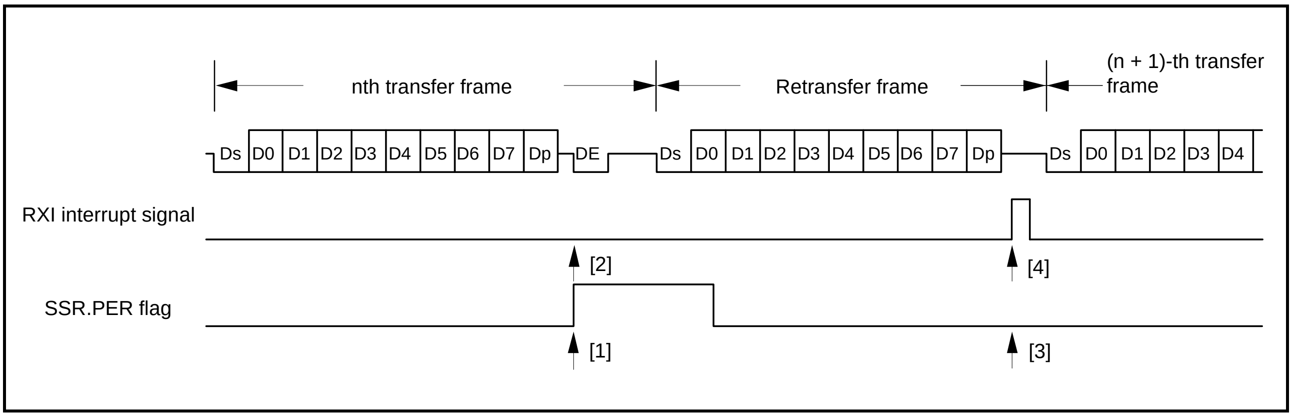


Figure 25.40 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)

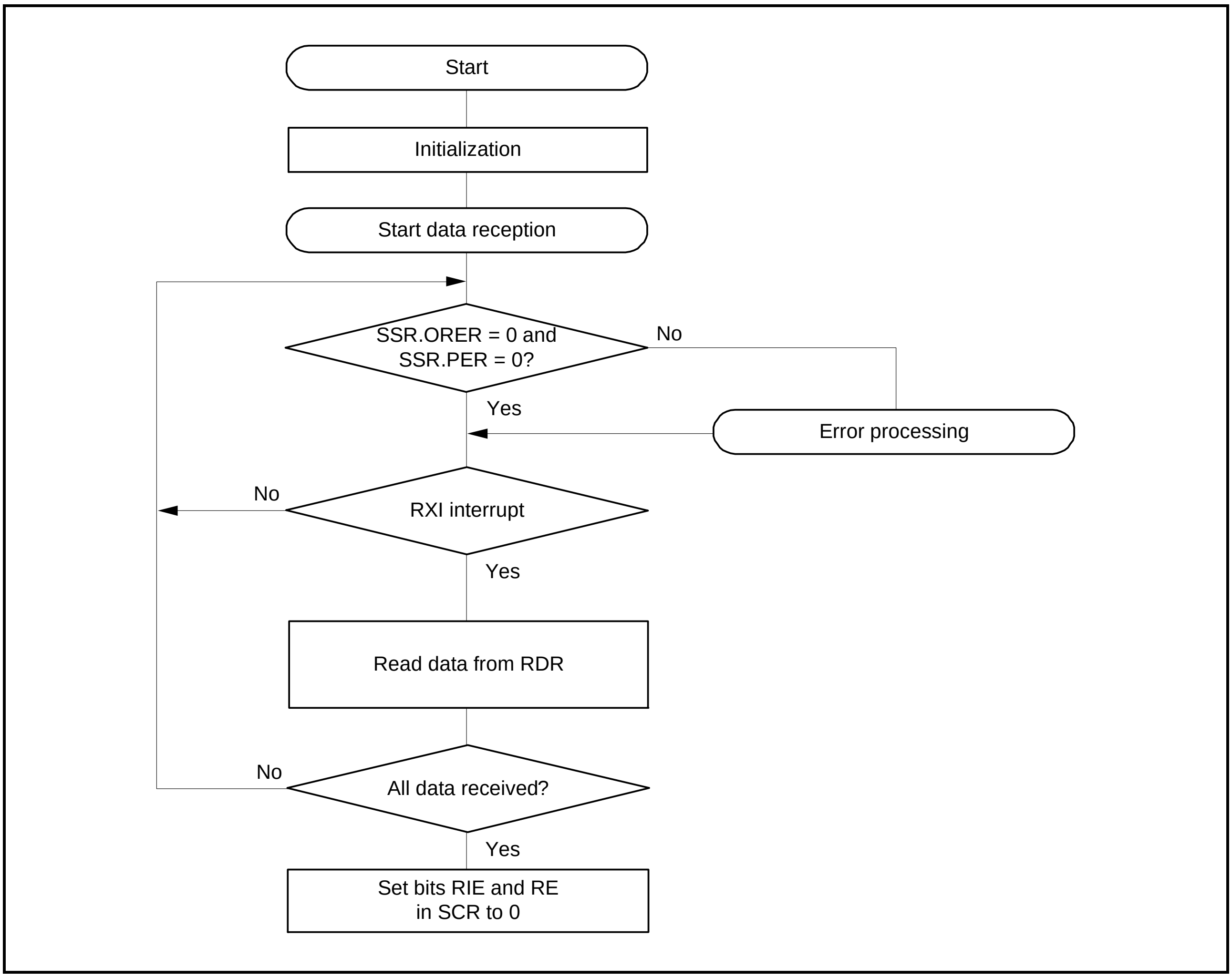


Figure 25.41 Sample Smart Card Interface Reception Flowchart

25.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in the SCR register when the GM bit in the SMR register is 1. Specifically, the minimum width of a clock pulse can be specified.
Figure 25.42 shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.

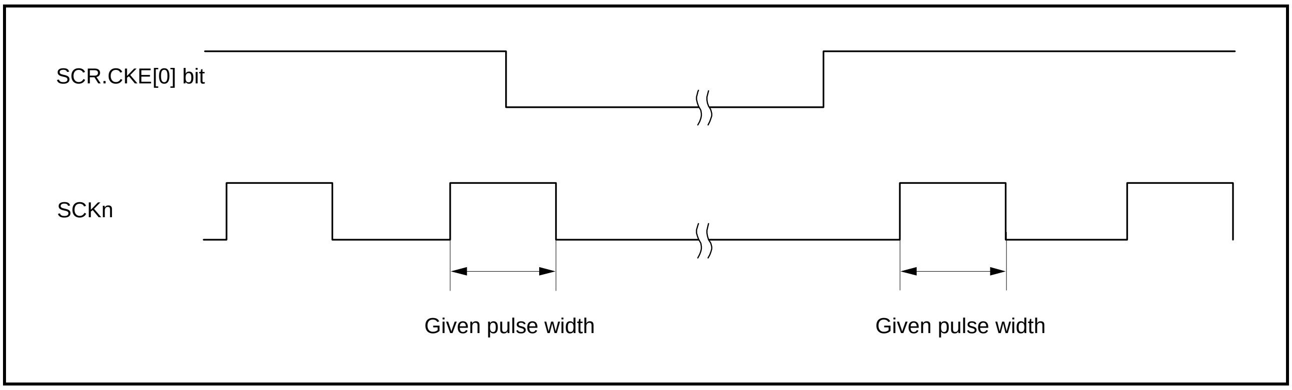


Figure 25.42 Clock Output Fixing Timing

At power-on, use the following procedure to secure the appropriate clock duty cycle.

(1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

25.7 Operation in Simple I²C Mode

Simple I²C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I²C format and timing of the I²C-bus are shown in Figure 25.43 and Figure 25.44.

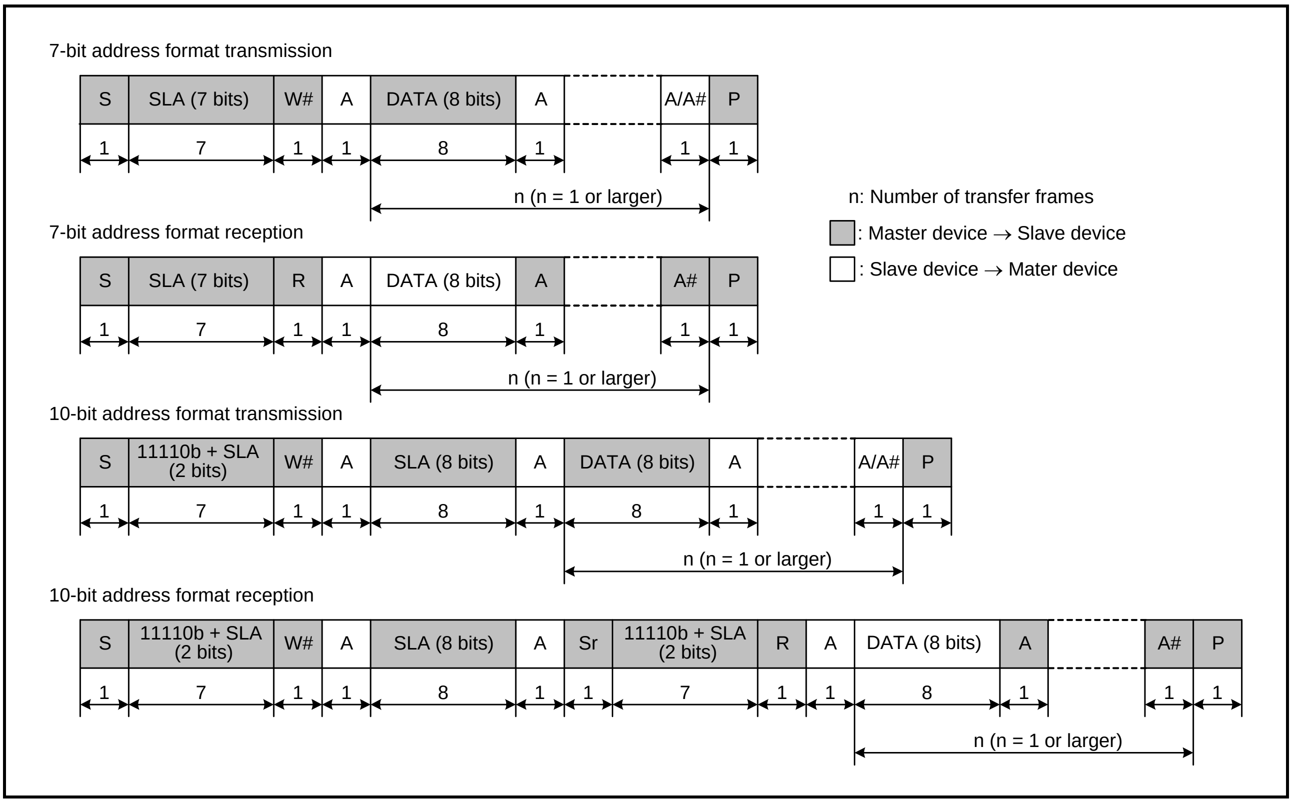


Figure 25.43 I²C-bus Format

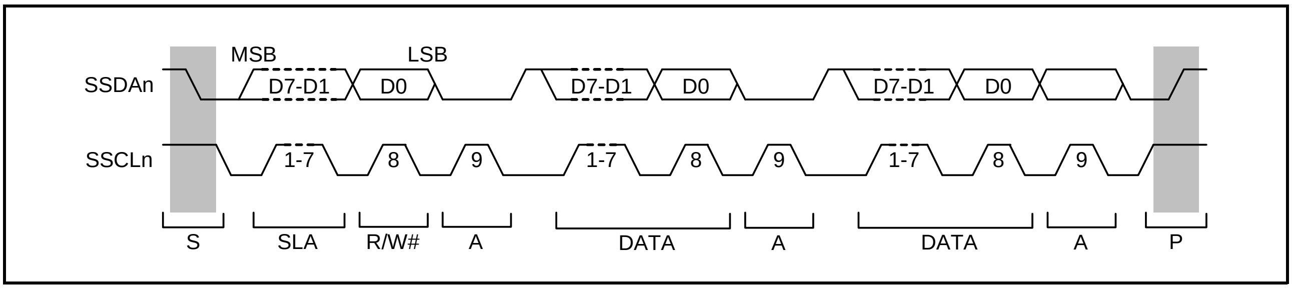


Figure 25.44 I²C-bus Timing (When SLA is 7 Bits)

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.

25.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in the SIMR3 register 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 the SIMR3 register is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in the SIMR3 register 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 the SIMR3 register is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ bit in the SIMR3 register 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 the SIMR3 register is set (to 0), and a stop-condition generated interrupt is output.

Figure 25.45 shows the timing of operations in the generation of start, restart, and stop conditions.

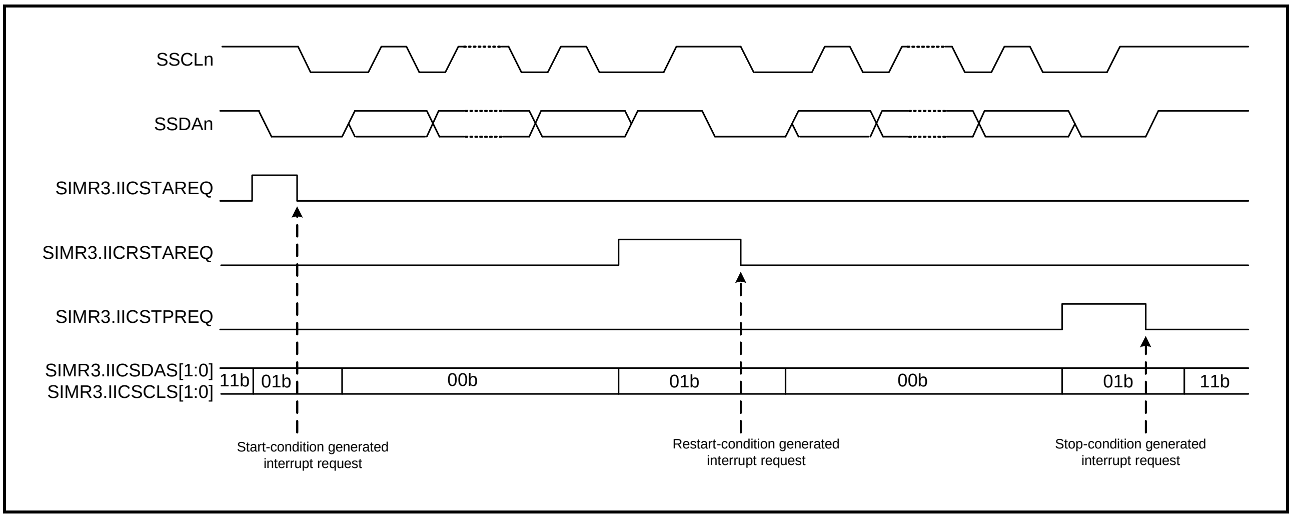


Figure 25.45 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

25.7.2 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 the SIMR2 register 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 the SIMR2 register 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 the SIMR2 register 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 the SIMR2 register 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 25.46 shows an example of operations to synchronize the clocks.

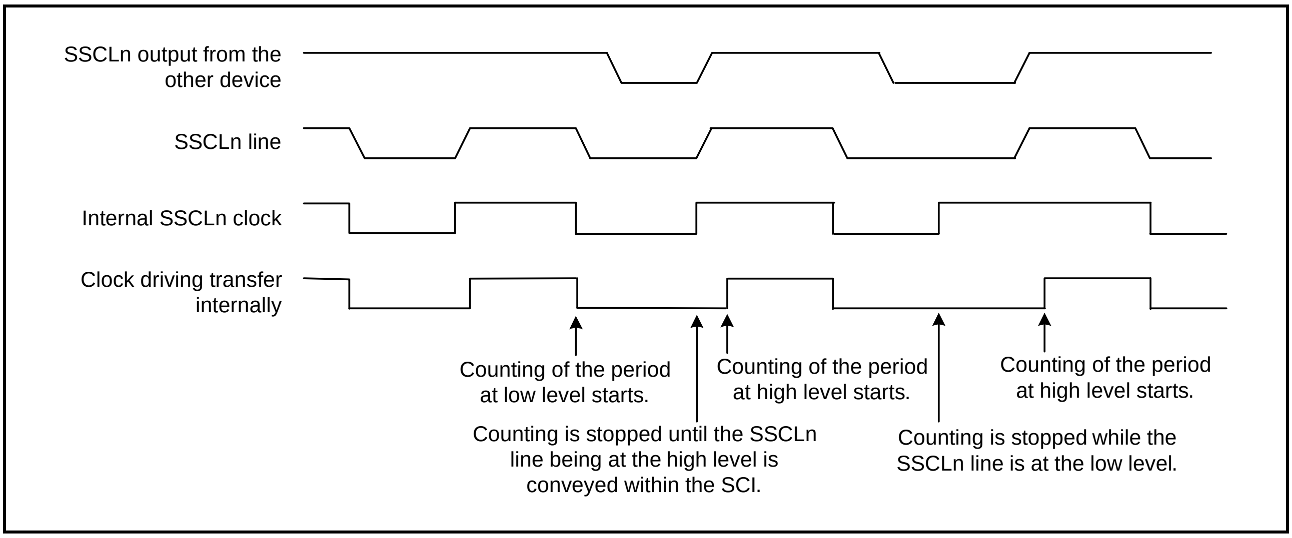


Figure 25.46 Example of Operations for Clock Synchronization

25.7.3 SSDA Output Delay

The IICDL[4:0] bits in the SIMR1 register can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the CKS[1:0] bits in the SMR register). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit. If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C in normal mode and fast mode).

Figure 25.47 shows the timing of delays in SSDA output.

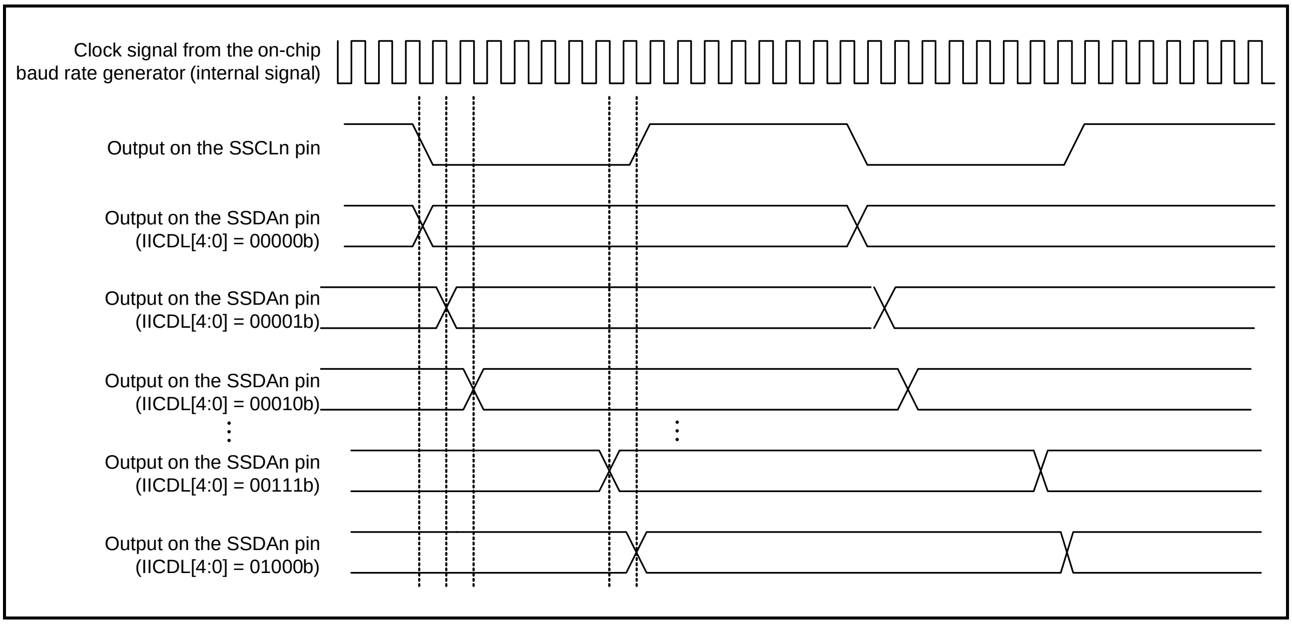
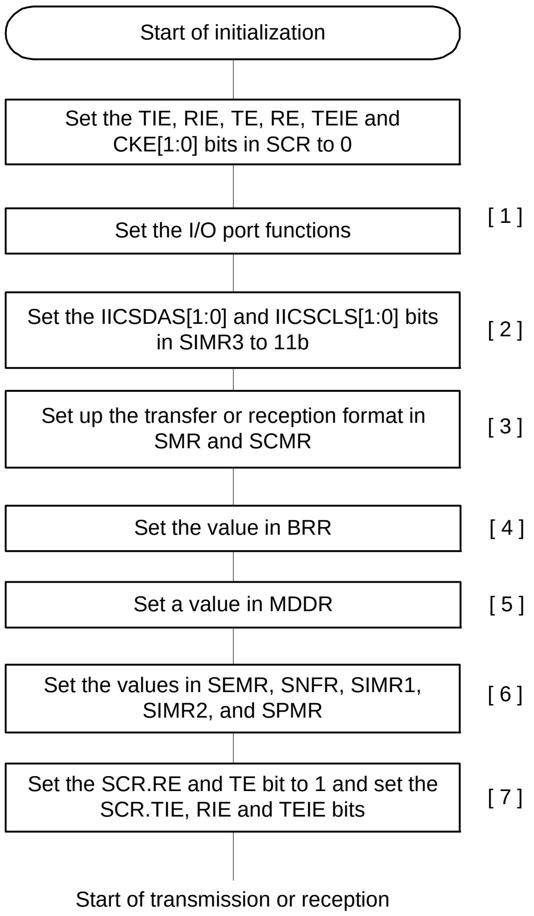


Figure 25.47 Timing of Delays in SSDA Output

25.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface following the example shown in Figure 25.48.
When changing the operating mode, transfer format, and so on, be sure to set SCR to its initial value before proceeding with the changes.
In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[2] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in SMR and SCMR.
In SMR, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to BRR.

[5] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is cleared to 0.

[6] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set the NFEN bit. Set the NFEN and BRME bits in SEMR.
In SNFR, set the NFCS[2:0] bits.
In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In SPMR, set all the bits to 0.

[7] Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0).
Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Figure 25.48 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

25.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 25.49 and Figure 25.50 show examples of operations in master transmission and Figure 25.51 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (RXI and ERI interrupt requests are disabled). See Table 25.29 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 25.51 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

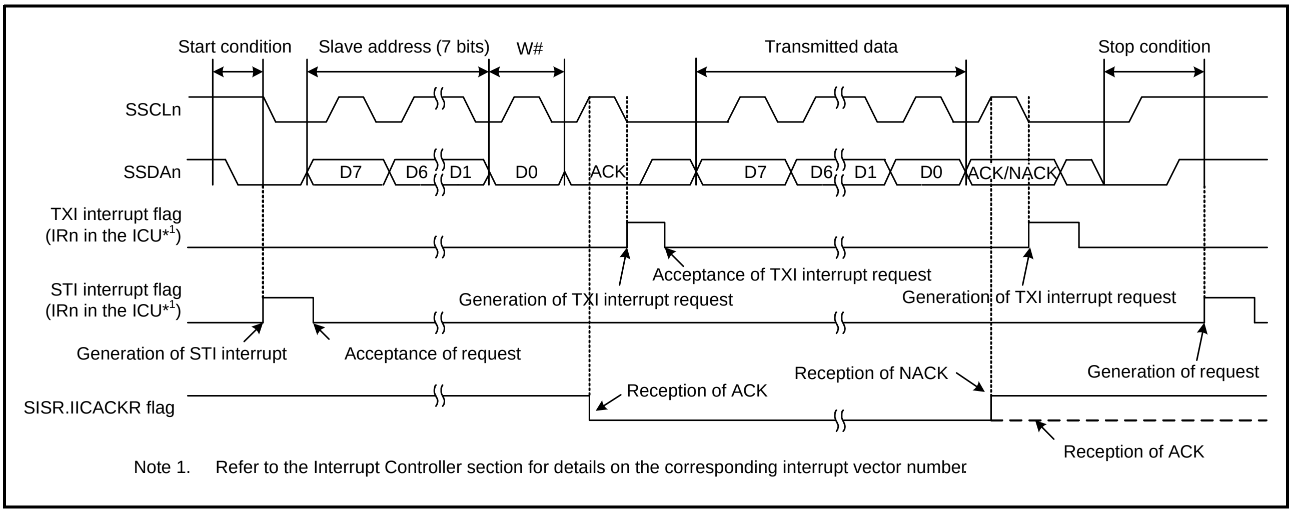


Figure 25.49 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

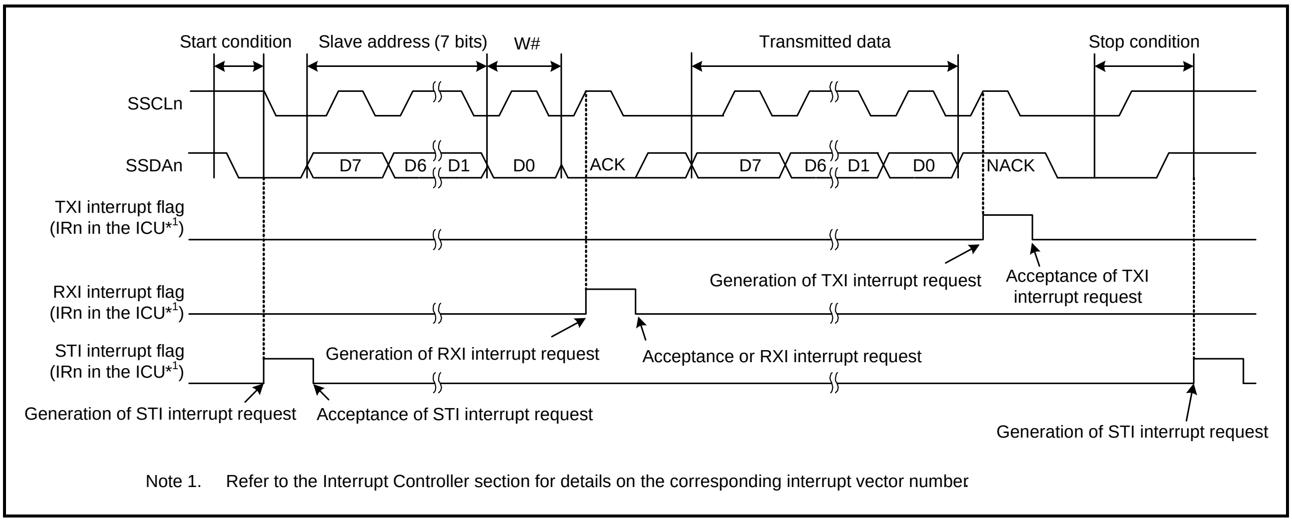
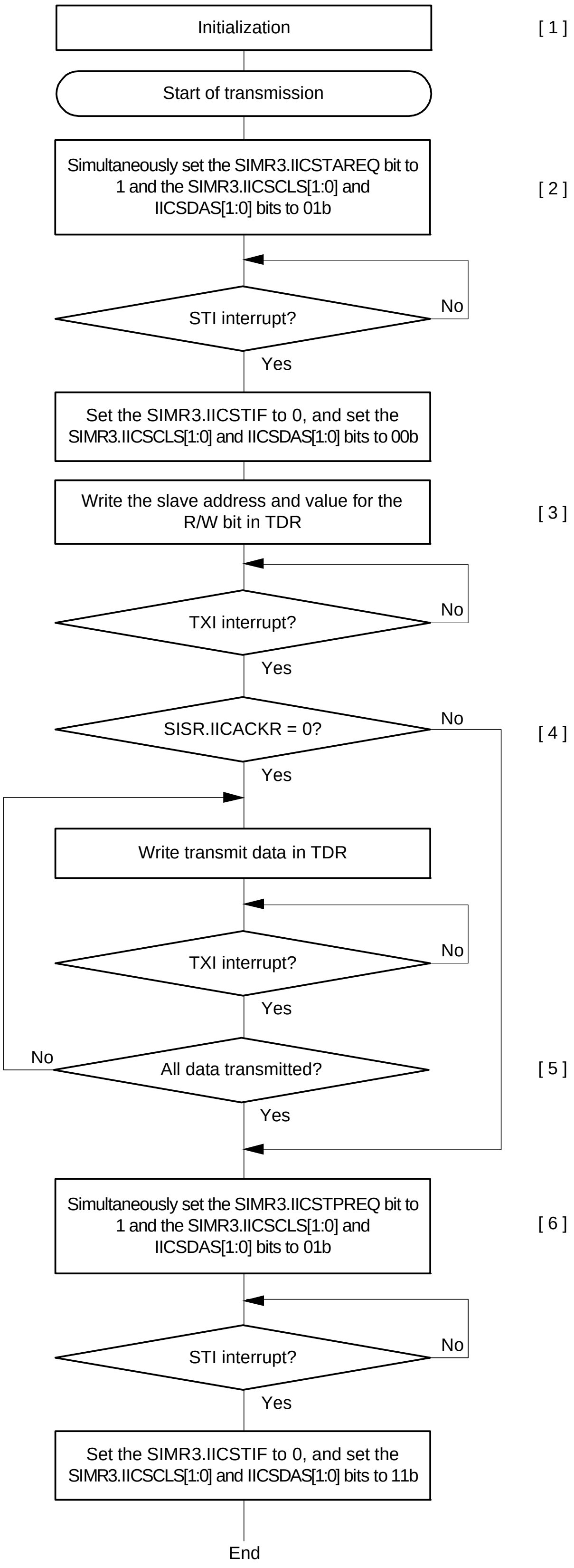


Figure 25.50 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[1] Initialization for simple I2C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a TXI request can activate the DTC to handle writing of data to TDR.

[6] Generation of a stop condition.

If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 25.51 Example of the Procedure for Master Transmission Operations in Simple I2C Mode (with Transmission Interrupts and Reception Interrupts in Use)

25.7.6 Master Reception (Simple I²C Mode)

Figure 25.52 shows an example of operations in simple I²C mode master reception and Figure 25.53 is a flowchart showing the procedure for master reception.
The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).
In simple I²C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

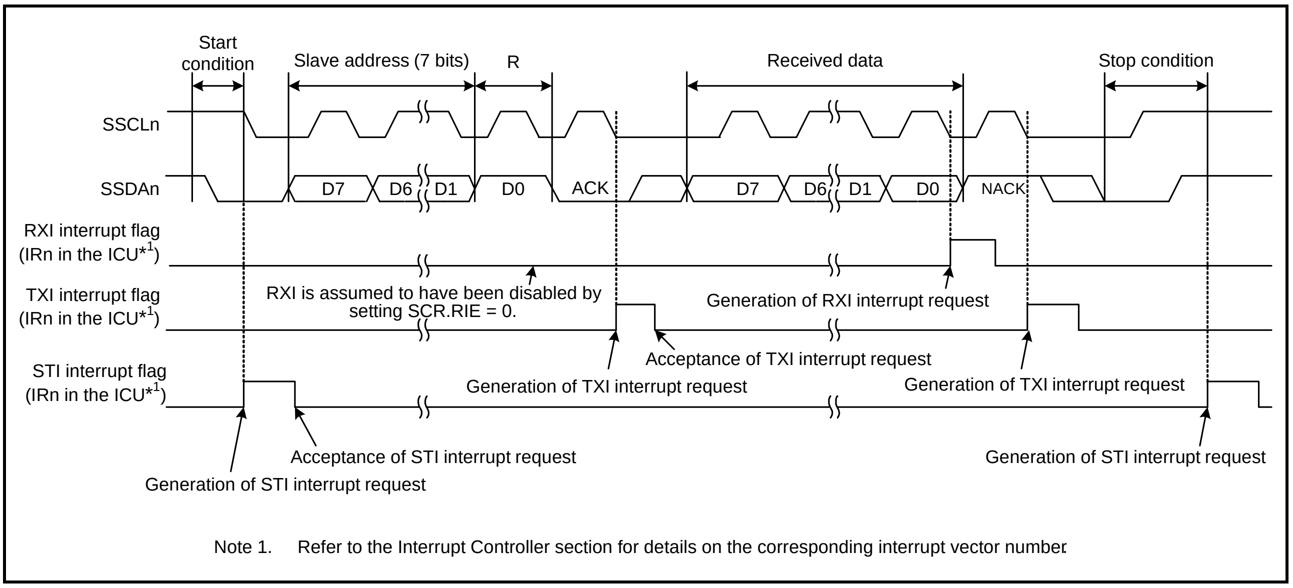
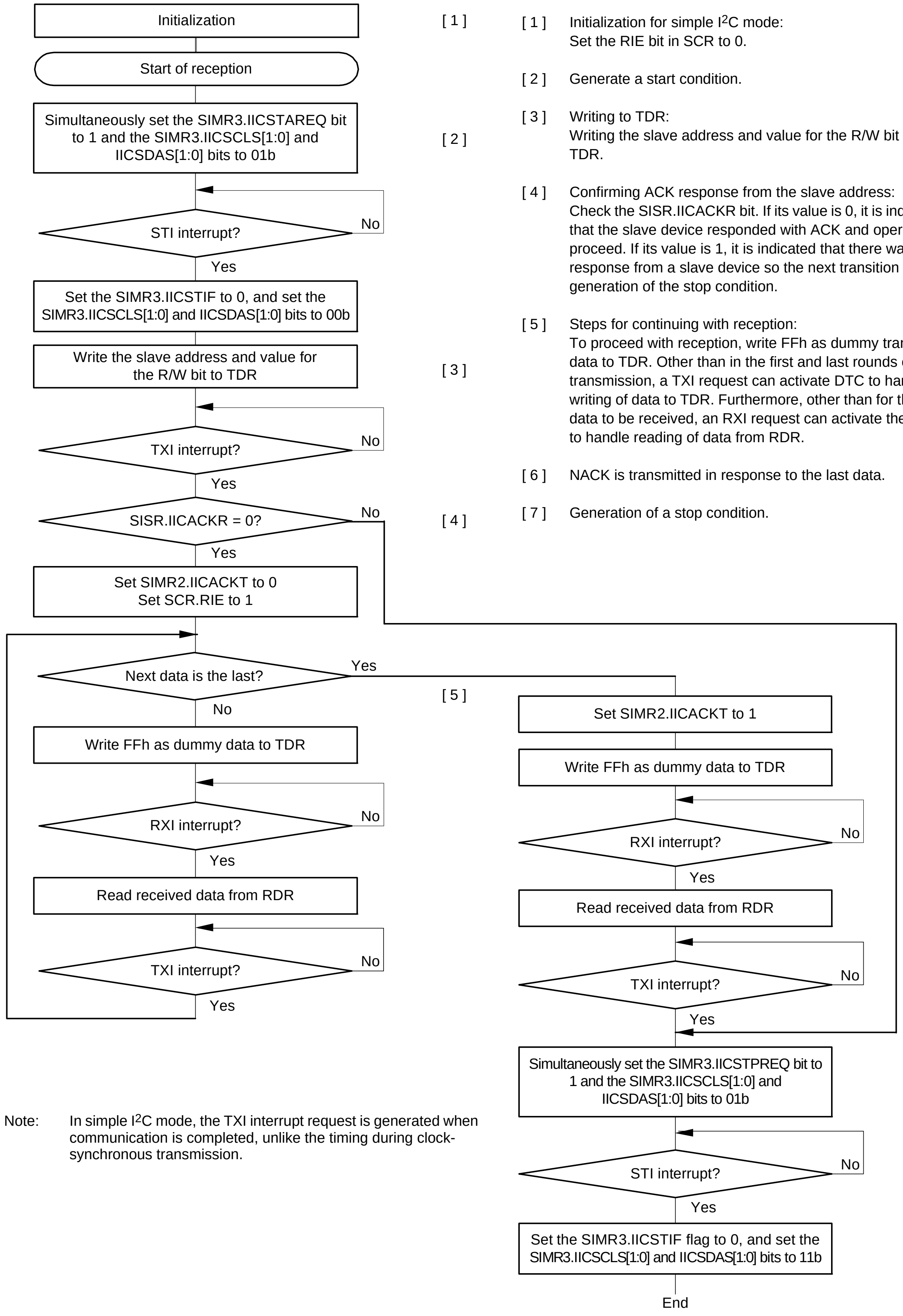


Figure 25.52 Example of Operations for Master Reception in Simple I²C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the RIE bit in SCR to 0.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with reception:
To proceed with reception, write FFh as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a TXI request can activate DTC to handle writing of data to TDR. Furthermore, other than for the last data to be received, an RXI request can activate the DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

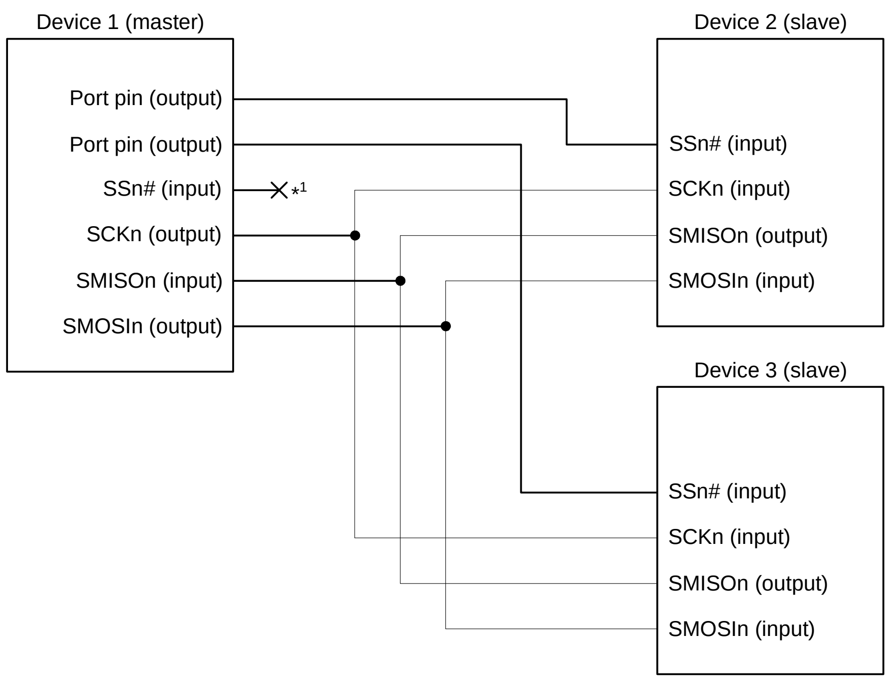
[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 25.53 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

25.8 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 25.54 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.
In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.
Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 25.54 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

25.8.1 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 25.26 lists the states of pins according to the mode and the level on the SSn# pin.

Table 25.26 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer can proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer cannot proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

25.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b and the MSS bit in the SPMR to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.
When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.
Control a general port pin to produce the SS output signal from the master.

25.8.3 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.

25.8.4 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 25.55. The relation is the same for both master and slave operation. This is the same as when the level on the SSn# pin is high. The SSn# pin can be used for another purpose. For details, refer to section 25.8.2, SS Function in Master Mode.

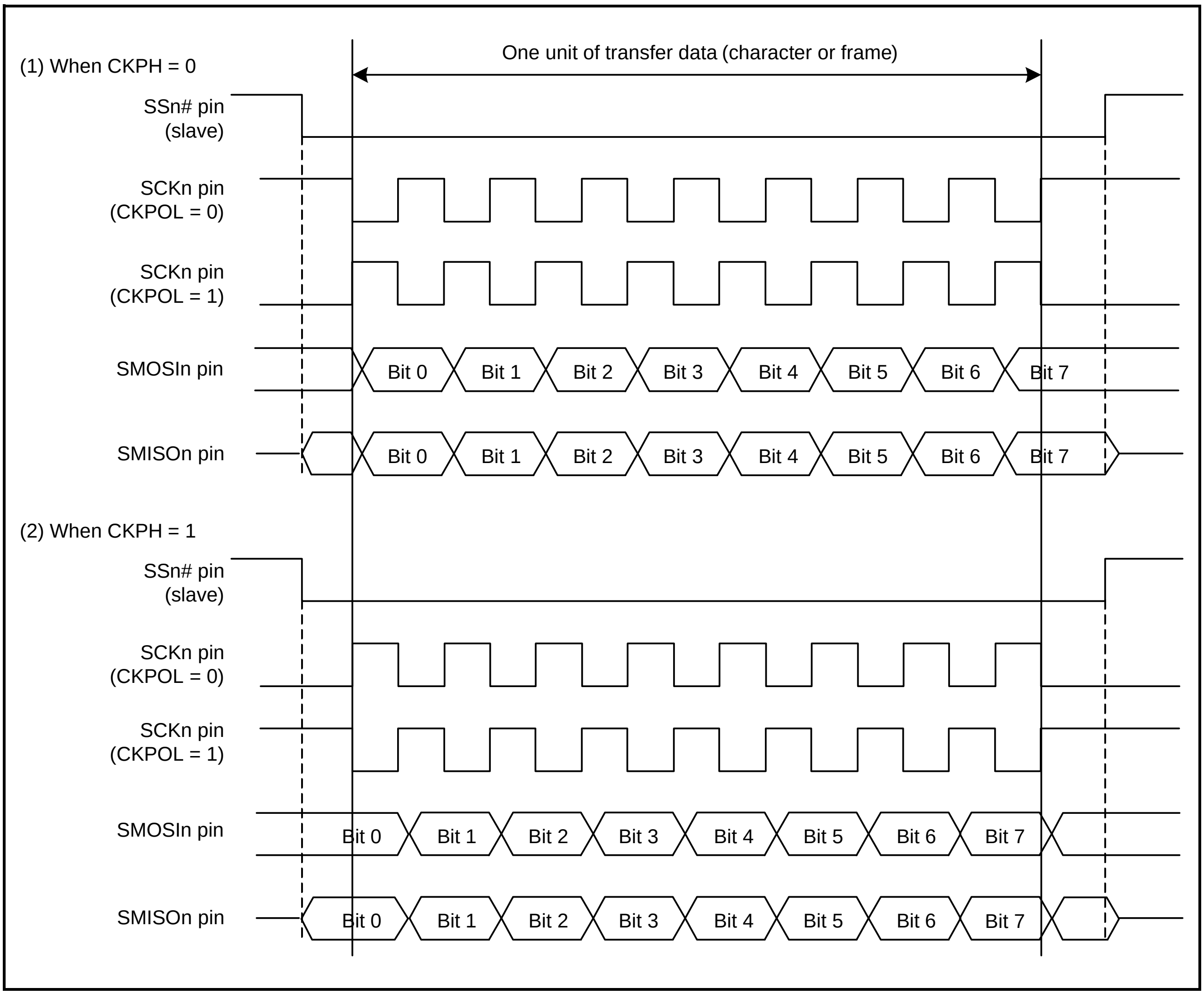


Figure 25.55 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

25.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 25.22, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR, are not initialized.
Note that changing the value of the TE bit from 1 to 0 or from 0 to 1 will lead to the generation of a transmit data empty interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

25.8.6 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.

25.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 25.56 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to $\frac{256}{160}$, and the bit rate is corrected to $\frac{160}{256}$. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the internal base clock.

Note: Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

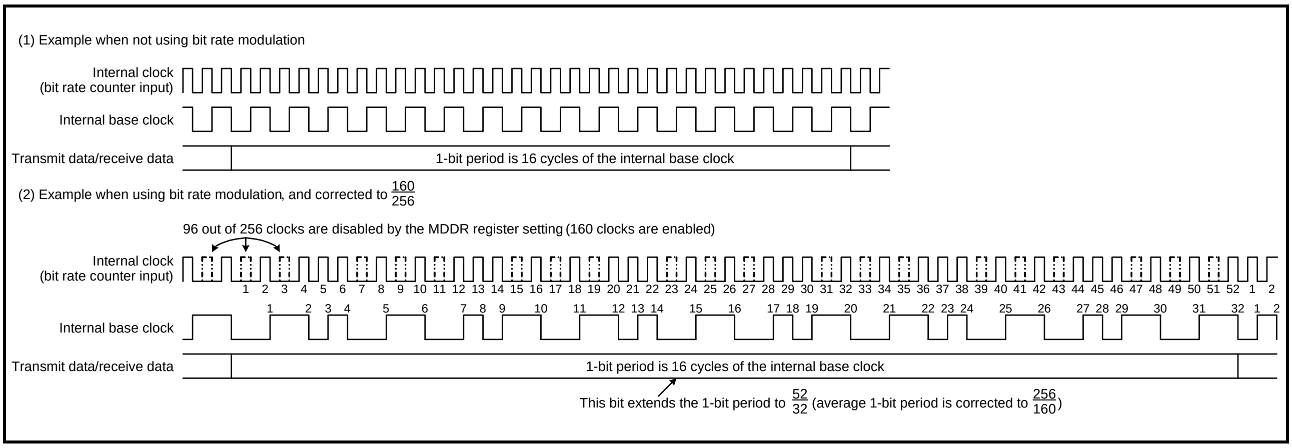


Figure 25.56 Example of the Base Clock When the Bit Rate Modulation Function is Used

25.10 Noise Cancellation Function

Figure 25.57 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 I^2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

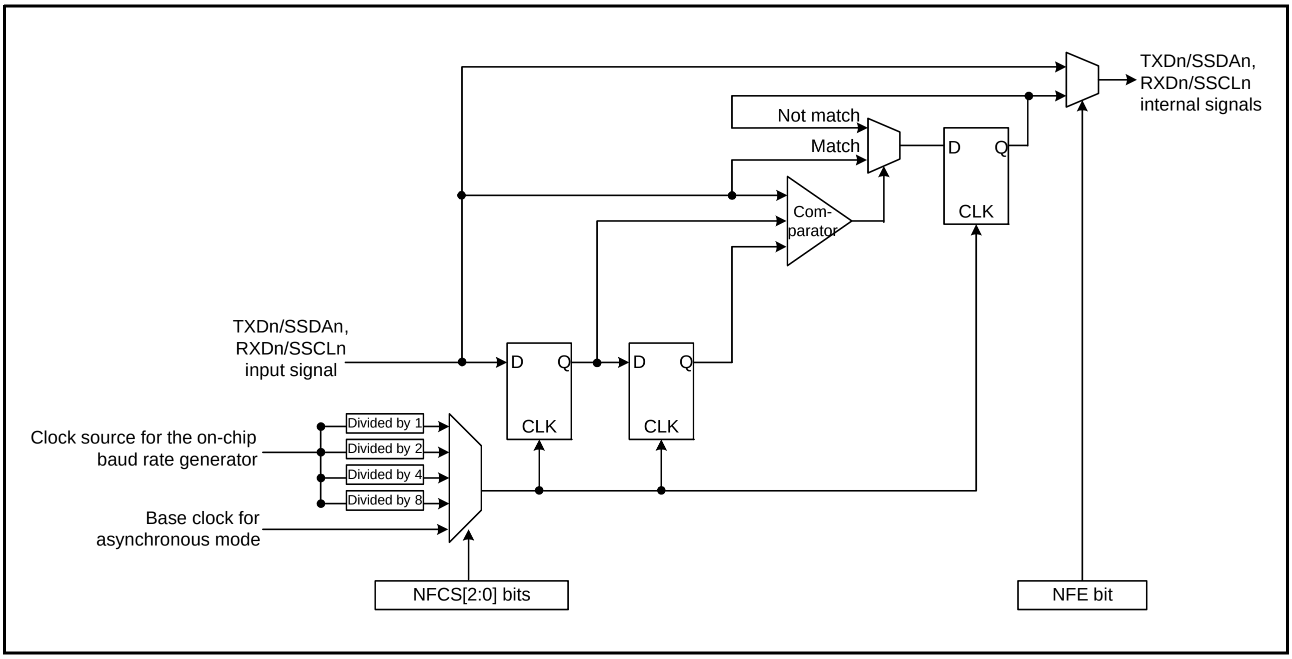


Figure 25.57 Block Diagram of Digital Noise Filter Circuit

25.11 Interrupt Sources

25.11.1 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 SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR) can also be used to discard an internally retained interrupt request.

25.11.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 25.27 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.
If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR. 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 to handle data transfer.
A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2
When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.
Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.
If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR. An RXI interrupt request can activate the DTC 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. In the case where asynchronous mode and 9-bit data length are selected
Note 2. To temporarily disable TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 25.27 Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	High ↑
RXI	Receive data full	RDRF	Possible	
TXI	Transmit data empty	TDRE	Possible	
TEI	Transmit end	TEND	Not possible	Low

25.11.3 Interrupts in Smart Card Interface Mode

Table 25.28 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 25.28 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	High ↑
RXI	Receive data full	—	Possible	
TXI	Transmit data empty	TEND	Possible	Low

Data transmission/reception using the DTC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in the SSR register is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC activation. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in the SSR register is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in the SCR register to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to 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 allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

25.11.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 25.29. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.
The DTC can also be used to handle transfer in simple I2C mode.
When the value of the IICINTM bit in the SIMR2 register is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC 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 beforehand, the TXI request will activate the DTC to handle transfer of the transmit data.
When the value of the IICINTM bit in the SIMR2 register 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 beforehand, the RXI request will activate the DTC to handle transfer of the received data.
Also, if the DTC is used for data transfer in reception or transmission, be sure to set up and enable the DTC before setting up the SCI.
When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 25.29 SCI Interrupt Sources

Name	**Interrupt Source**	Interrupt Flag	DTC Activation	Priority
RXI	Reception, ACK detection	—	Possible	High ↑
TXI	Transmission, NACK detection	—	Possible*1	
STI	Completion of generation of a start, restart, or stop condition	IICSTIF	Not possible	Low

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

25.12 Usage Notes

25.12.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

25.12.2 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 the SSR register is set to 1 (framing error has occurred), and the PER flag in the SSR register may also be set to 1 (parity error has occurred). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

25.12.3 Mark State and Generating Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (serial transmission is enabled), the I/O port function is used to set the TXDn pin to output high and set the pin mode to a general I/O port pin, and thus place the communication line in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output low and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

25.12.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (ORER) in the SSR register is set to 1, even if data is written to the TDR register. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the RE bit in the SCR register is set to 0 (serial reception is disabled).

25.12.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

25.12.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update TDR by the CPU or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (see Figure 25.58).

(2) Continuous transmission

(a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (see Figure 25.58).

(b) When updating TDR after bit 7 has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (see Figure 25.58).

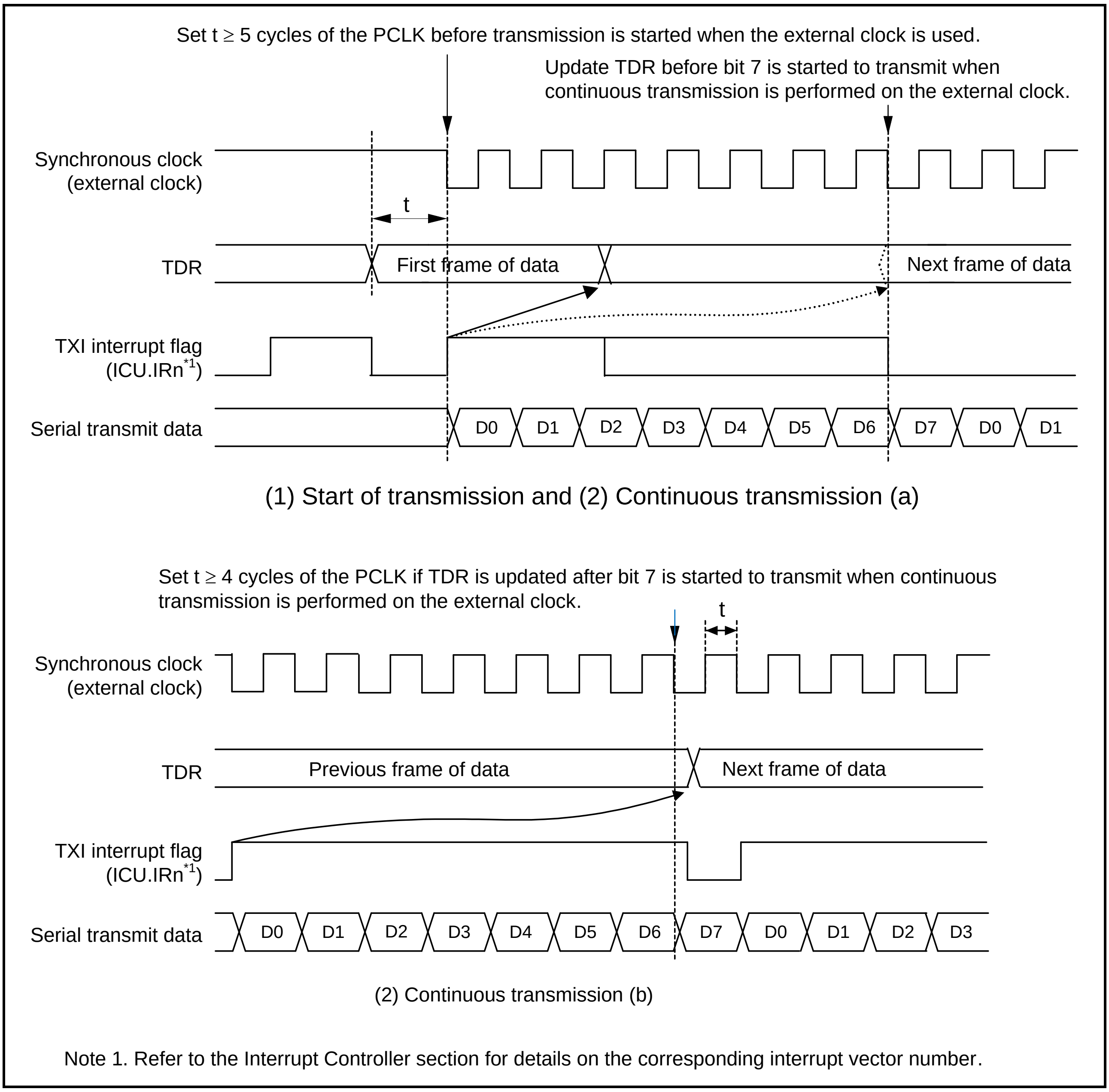


Figure 25.58 Restrictions on Use of External Clock in Clock Synchronous Transmission

25.12.7 Restrictions on Using DTC

When using the DTC to read RDR, RDRH, and RDRL, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

25.12.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUb).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR bit) in the interrupt controller to 0.

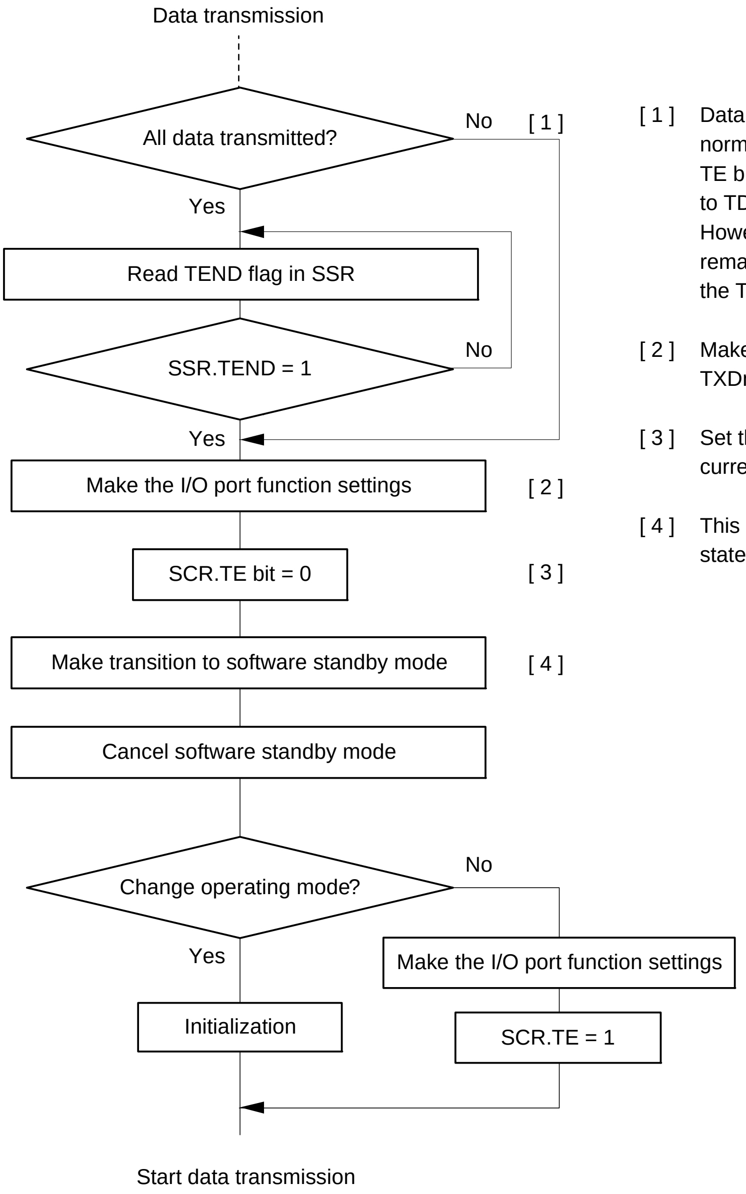
25.12.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the TEND bit in the SSR. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.
To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.
Figure 25.59 shows a sample flowchart for transition to software standby mode during transmission. Figure 25.60 and Figure 25.61 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 the SCR register). 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 25.62 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] Make the I/O port function settings to switch the TXDn pin to operate as a general I/O port.

[3] Set 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 25.59 Example of Flowchart for Transition to Software Standby Mode during Transmission

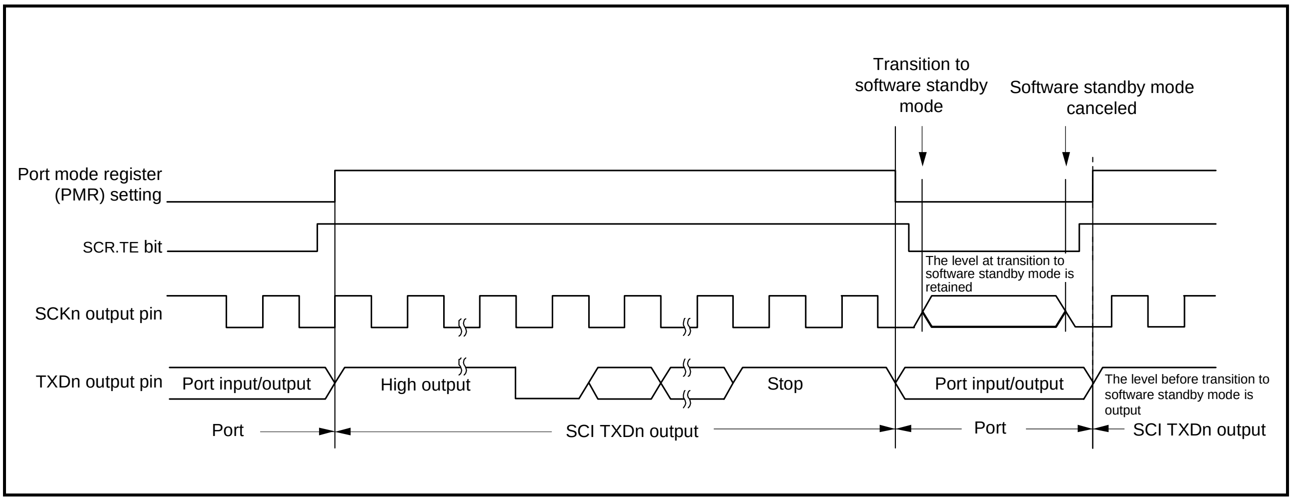


Figure 25.60 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

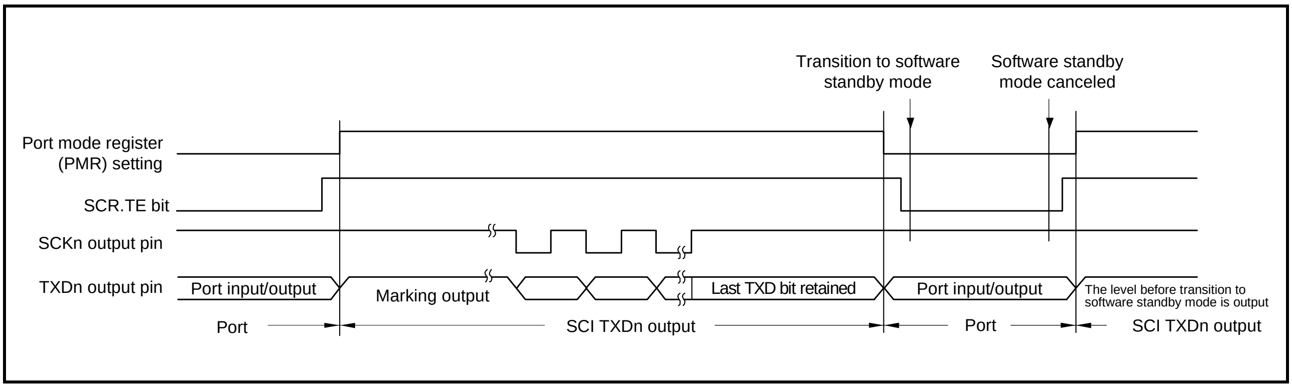


Figure 25.61 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

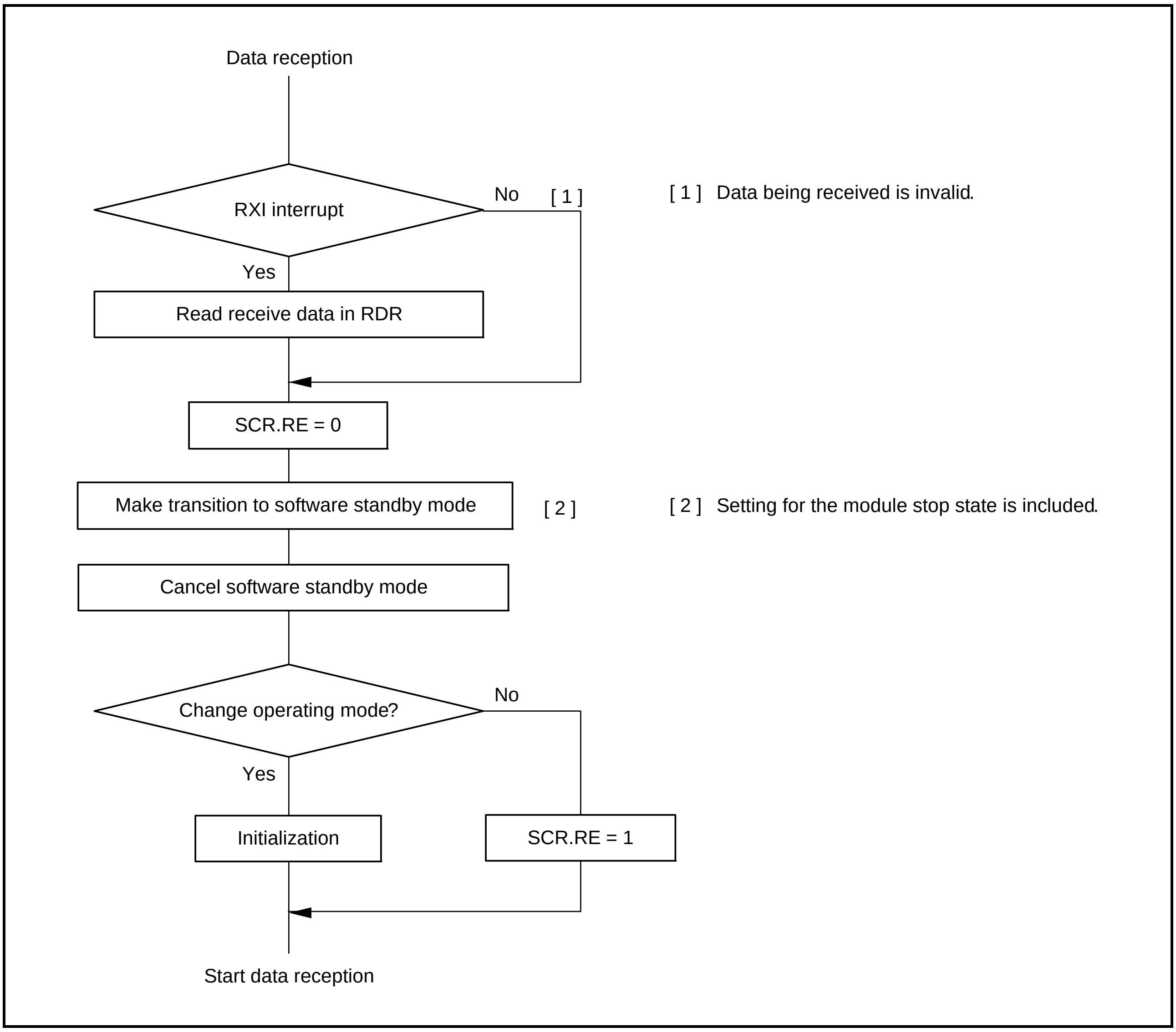


Figure 25.62 Example of Flowchart for Transition to Software Standby Mode during Reception

25.12.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

25.12.11 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 25.63. If the TE and RE bits in the SCR become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

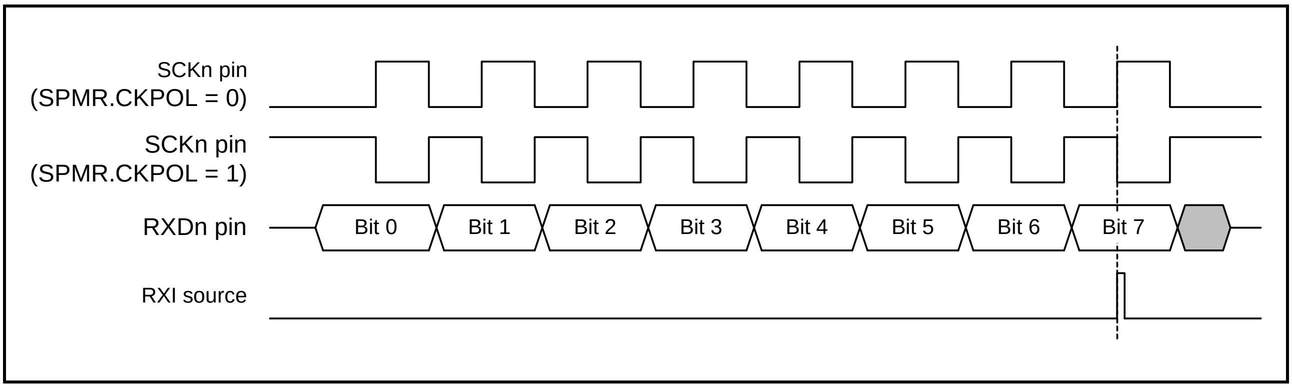


Figure 25.63 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal 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.

25.12.12 Note on Transmit Enable Bit (TE Bit)

When setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is “TXDn”, output of the pin becomes high impedance.

Prevent the TXDn line from becoming high impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.

(2) Change the pin function to “general-purpose I/O port, output” before setting the SCR.TE bit to 0.
Set the SCR.TE bit to 1 before changing the pin function to “TXDn”.

26. I2C-bus Interface (RIICa)

This MCU has a single-channel I2C-bus interface (RIIC).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

Table 26.1 lists the specifications of the RIIC, Figure 26.1 shows a block diagram of the RIIC, and Figure 26.2 shows an example of I/O pin connections to external circuits (I2C-bus configuration example). Table 26.2 lists the I/O pins of the RIIC.

Table 26.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode is supported (up to 400 kbps)
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4 to 96%.
Issuing and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledge bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit. • For reception, the acknowledge bit is automatically transmitted. If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.
Wait function	• In reception, the following periods of waiting can be obtained by holding the SCL clock at the low level: Waiting between the eighth and ninth clock cycles Waiting between the ninth clock cycle and the first clock cycle of the next transfer
SDA output delay function	Timing of the output of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	• For multi-master operation Operation to synchronize the SCL clock in cases of conflict with the SCL signal from another master is possible. When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line. In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line. • Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions). • Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable. • Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL clock.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Error in transfer or occurrence of events Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmit end

Table 26.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode

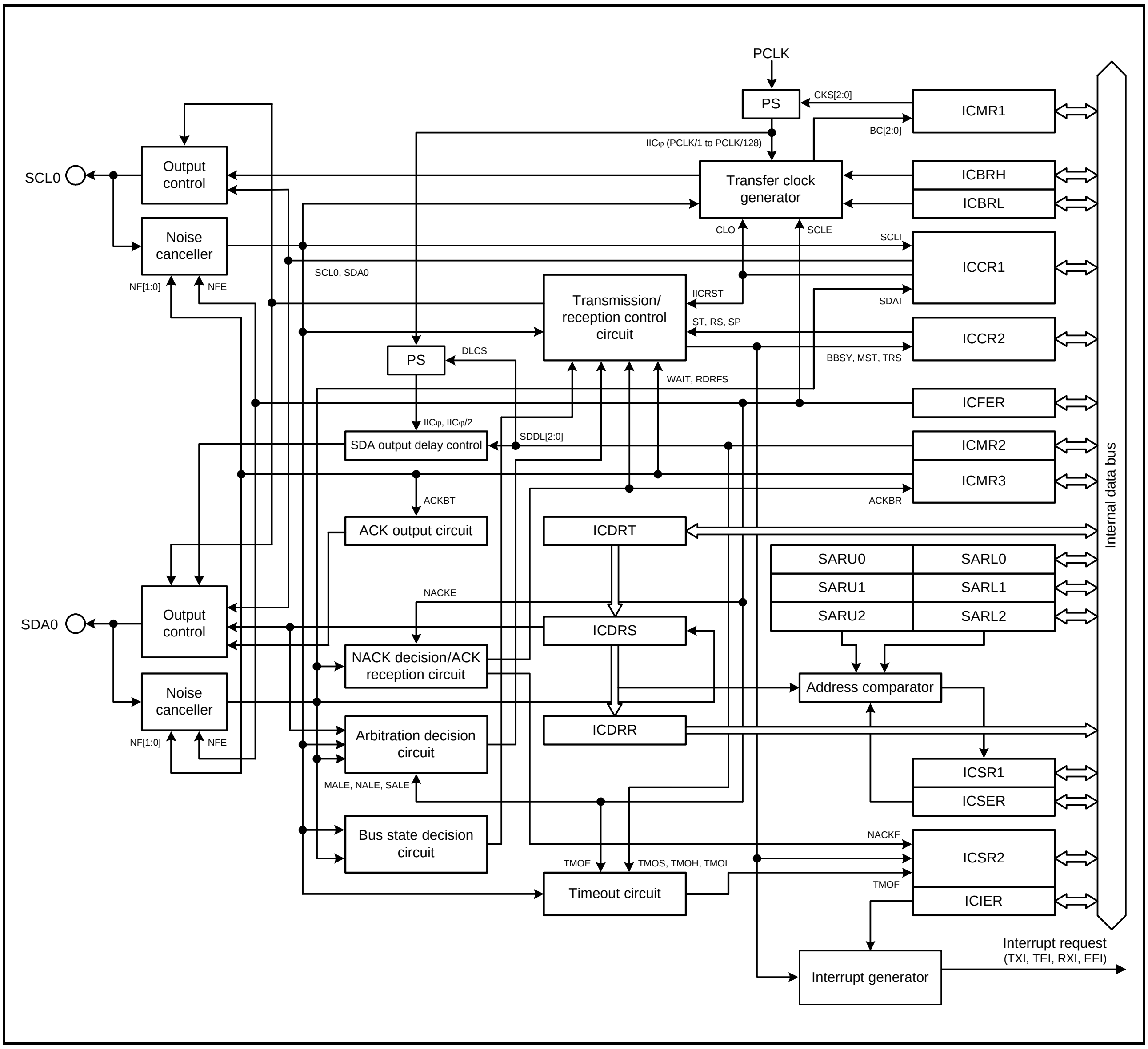


Figure 26.1 RIIC Block Diagram

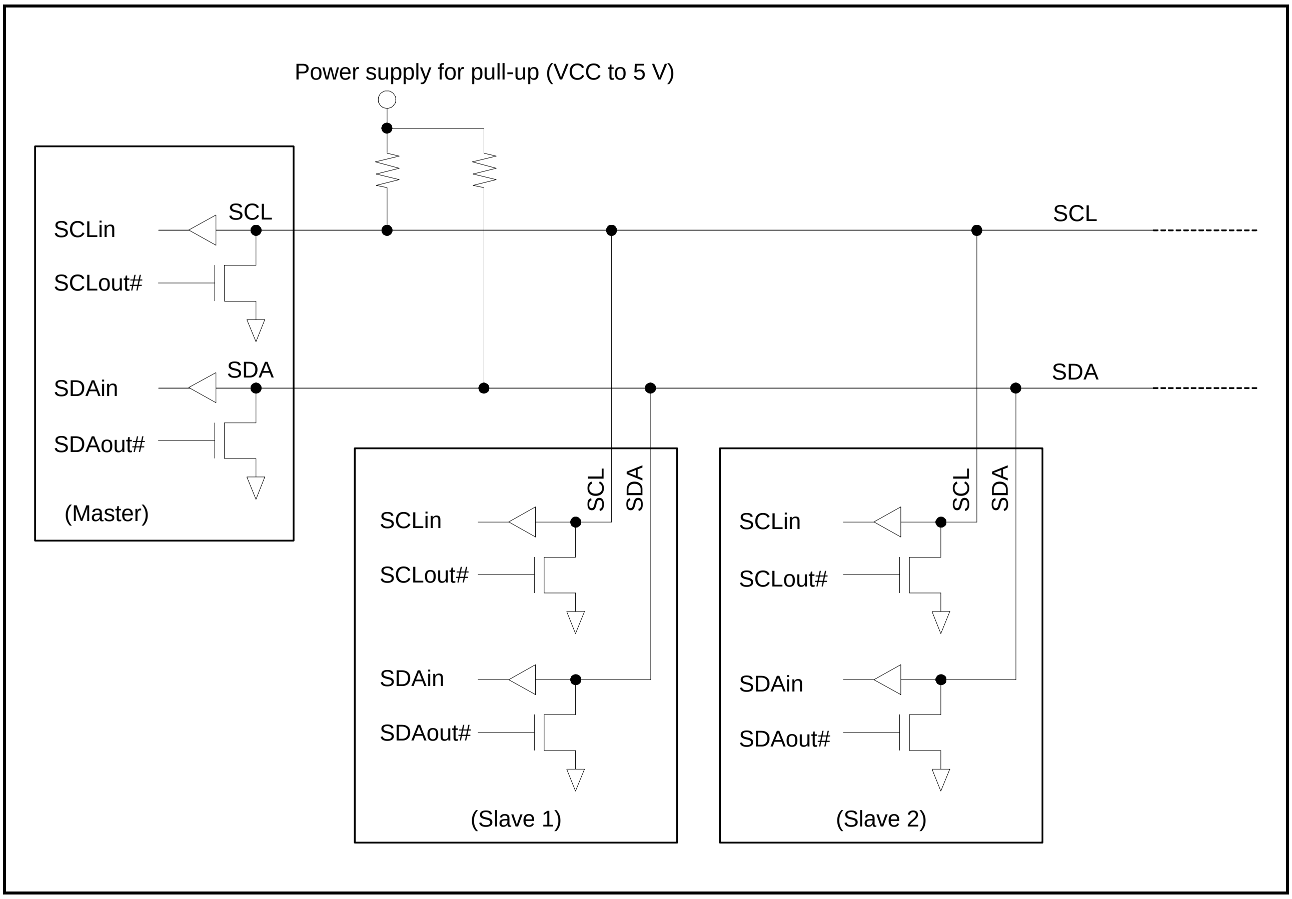


Figure 26.2 I/O Pin Connection to the External Circuit (I2C-bus Configuration Example)

The input level of the signals for RIIC is CMOS when I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when SMBus is selected (ICMR3.SMBS bit is 1).

Table 26.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin

26.2 Register Descriptions

26.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA0 pin low. 1: The RIIC has released the SDA0 pin. • Write: 0: The RIIC drives the SDA0 pin low. 1: The RIIC releases the SDA0 pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCL0 pin low. 1: The RIIC has released the SCL0 pin. • Write: 0: The RIIC drives the SCL0 pin low. 1: The RIIC releases the SCL0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Extra SCL Clock Cycle Output	0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL0/SDA0 output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.
When writing to these bits, also write 0 to the SOWP bit.
The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.
Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.
When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 26.11.2, 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 26.3 lists the resets of the RIIC.
The RIIC reset resets all registers and internal states of the RIIC, and the internal reset resets the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 26.14, Resets and Register and Function States When Issuing Each Condition.
An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL0 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 26.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 26.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

26.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Issuance Request	0: Does not request to issue a start condition. 1: Requests to issue a start condition.	R/W
b2	RS	Restart Condition Issuance Request	0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.	R/W
b3	SP	Stop Condition Issuance Request	0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free state).

For details on the start condition issuance, refer to section 26.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.
When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition issuance, refer to section 26.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (requests to issue a restart condition) is written to the RS bit in slave mode, the restart condition is not issued but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.
When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition issuance, refer to section 26.10, 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 ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of TRS bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally according to the restart condition issuance request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (a start condition is detected with ICCR2.BBSY flag is 1 and ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by issuing of a start condition and issuing or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is 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 ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C-bus is occupied (bus busy state) or released (bus free state).

This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been issued.

When the SDA0 line changes from low to high under the condition of SCL0 line = high, this bit is set to 0 after the bus free time (specified in the ICBRL register) 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 the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

26.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock source (IICφ) for the RIIC. b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred bytes when the SCL0 line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

26.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCL0 line is at a low level. 1: Count-up is enabled while the SCL0 line is at a low level.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is at a high level. 1: Count-up is enabled while the SCL0 line is at a high level.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0 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 bit is 1 (IICφ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICφ cycles 0 1 0: 3 or 4 IICφ cycles 0 1 1: 5 or 6 IICφ cycles 1 0 0: 7 or 8 IICφ cycles 1 0 1: 9 or 10 IICφ cycles 1 1 0: 11 or 12 IICφ cycles 1 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 26.11.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

Set the SDA output delay time to meet the I^2C-bus specification (within the data enable time/acknowledge enable time*[1]) or the SMBus specification (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 specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 26.5, SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: Standard-mode (Sm))
900 ns (up to 400 kbps: Fast-mode (Fm))

26.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Receive Acknowledge	0: 0 is received as the acknowledge bit (ACK reception). 1: 1 is received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is sent as the acknowledge bit (ACK transmission). 1: 1 is sent as the acknowledge bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL0 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 SCL0 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 the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 26.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 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 ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the 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 ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL0 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 SCL0 line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL0 line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the I2C-bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL0 line is held low from the falling edge of the ninth clock cycle until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

26.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Circuit Enable	0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.
For details on the timeout function, refer to section 26.11.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.
When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.
When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.
For details on the NACK reception transfer suspension function, refer to section 26.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.
When the SCLE bit is set to 0 (no SCL synchronous circuit used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCL0 line state. For this reason, if the bus load of the I^2C-bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.
This bit must not be set to 0 except for checking the output of the set transfer rate.

26.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.
When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.
When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.
When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.
When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.
When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.
When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.
For details on the device-ID address detection, refer to section 26.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.
When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.
When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

26.2.8 I2C-bus Interface Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Transmit end interrupt (TEI) request is disabled. 1: Transmit end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Request Enable)

This bit is used to enable or disable transmit end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

26.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

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

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

(host address detection is enabled)
This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

26.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmit End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

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

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.

[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The RIIC can also 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 SDA0 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 SDA0 line is driven low while the internal SDA output is at a high level (the SDA0 pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA0 line level
- When the ICCR2.ST bit is set to 1 (start condition issuance request) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL 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 SDA0 line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 26.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

ICFER			ICSR2		
MALE	**NALE**	**SALE**	**AL**	**Error**	**Arbitration-Lost Generation Source**
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDA0 line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End Flag)

[Setting condition]

- At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1 while the ICFER.NACKE bit 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.

26.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	A slave address is set.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set.	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

26.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

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

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	A slave address is set.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

26.2.13 I^{2}C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 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 (refer to section 26.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))

26.2.14 I²C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-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 ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I²C transfer rate and the SCL clock duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCL0 line rising time [tr] + SCL0 line falling time [tf]}
Duty cycle = {SCL0 line rising time [tr]*2 + (ICBRH + 1) / IICφ} / {SCL0 line falling time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCL0 line rising time [tr] and SCL0 line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I²C-bus specification from NXP Semiconductors.

Table 26.5 lists examples of ICBRH/ICBRL settings.

Table 26.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			32			33		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)	110b	22 (F6h)	26 (FAh)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)	100b	17 (F1h)	20 (F4h)
100	010b	2 (E2h)	3 (E3h)	011b	15 (EFh)	18 (F2h)	011b	16 (F0h)	19 (F3h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)	001b	9 (E9h)	21 (F5h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)		
	40		
	CKS[2:0]	ICBRH	ICBRL
10	111b	13 (7Dh)	15 (7Fh)
50	100b	21 (F5h)	24 (F8h)
100	011b	19 (F3h)	23 (F7h)
400	001b	11 (7Bh)	25 (F9h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rising time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns
SCL0 line falling time (tf): 400 kbps or less (Sm/Fm): 300 ns
For the specified values of SCL0 line rising time (tr) and SCL0 line falling time (tf), see the I2C-bus specification from NXP Semiconductors.

26.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

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

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
the ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

26.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

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

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
the ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

26.2.17 I2C-bus Shift Register (ICDRS)

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

ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDA0 pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
ICDRS register cannot be accessed directly.

26.3 Operation

26.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledge. The first byte following a start condition or restart condition is an address byte used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 26.3 shows the I2C-bus format, and Figure 26.4 shows the I2C-bus timing.

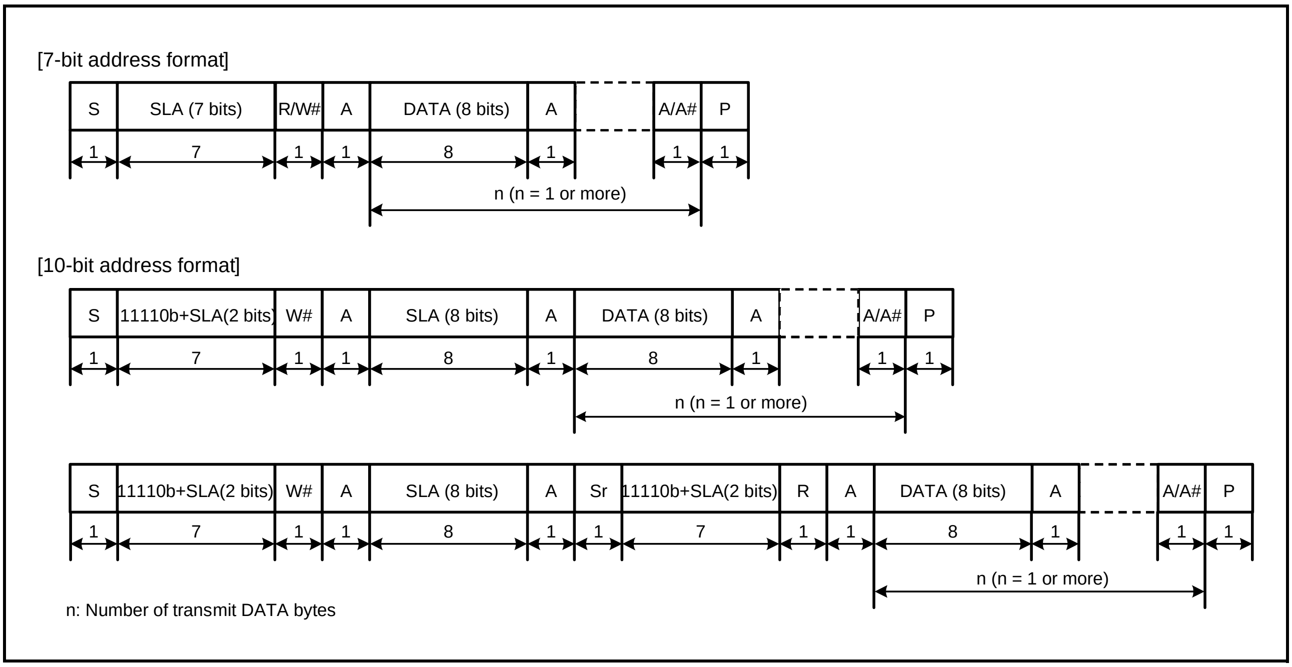


Figure 26.3 I2C-bus Format

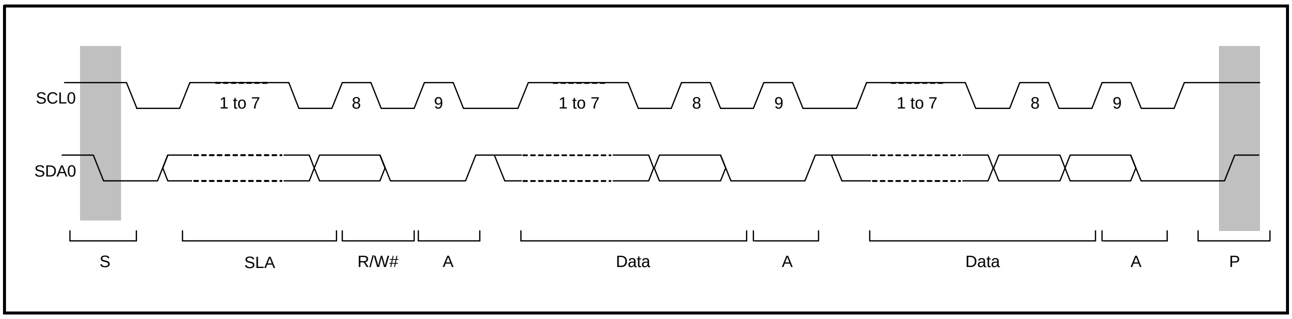


Figure 26.4 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA0 line low from high level while the SCL0 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 SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receive device drives the SDA0 line high.

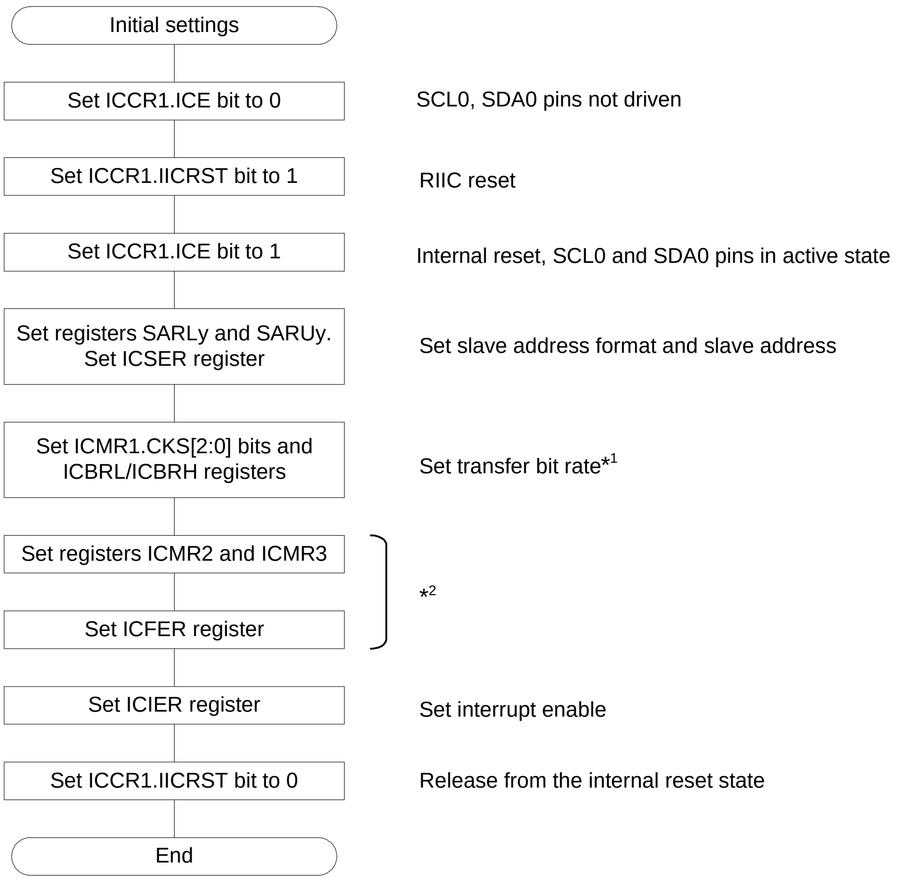
Sr: Restart condition. The master device drives the SDA0 line low from the high level after the setup time has elapsed with the SCL0 line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low level while the SCL0 line is at a high level.

26.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 26.5.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 26.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.



y = 0 to 2

Note 1. When the RIIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as necessary.

Figure 26.5 Example of RIIC Initialization Flowchart

26.3.3 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 acknowledgments. Figure 26.6 shows an example of usage of master transmission and Figure 26.7 to Figure 26.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 26.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL0 line low until the data for transmission are ready or a stop condition is issued.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

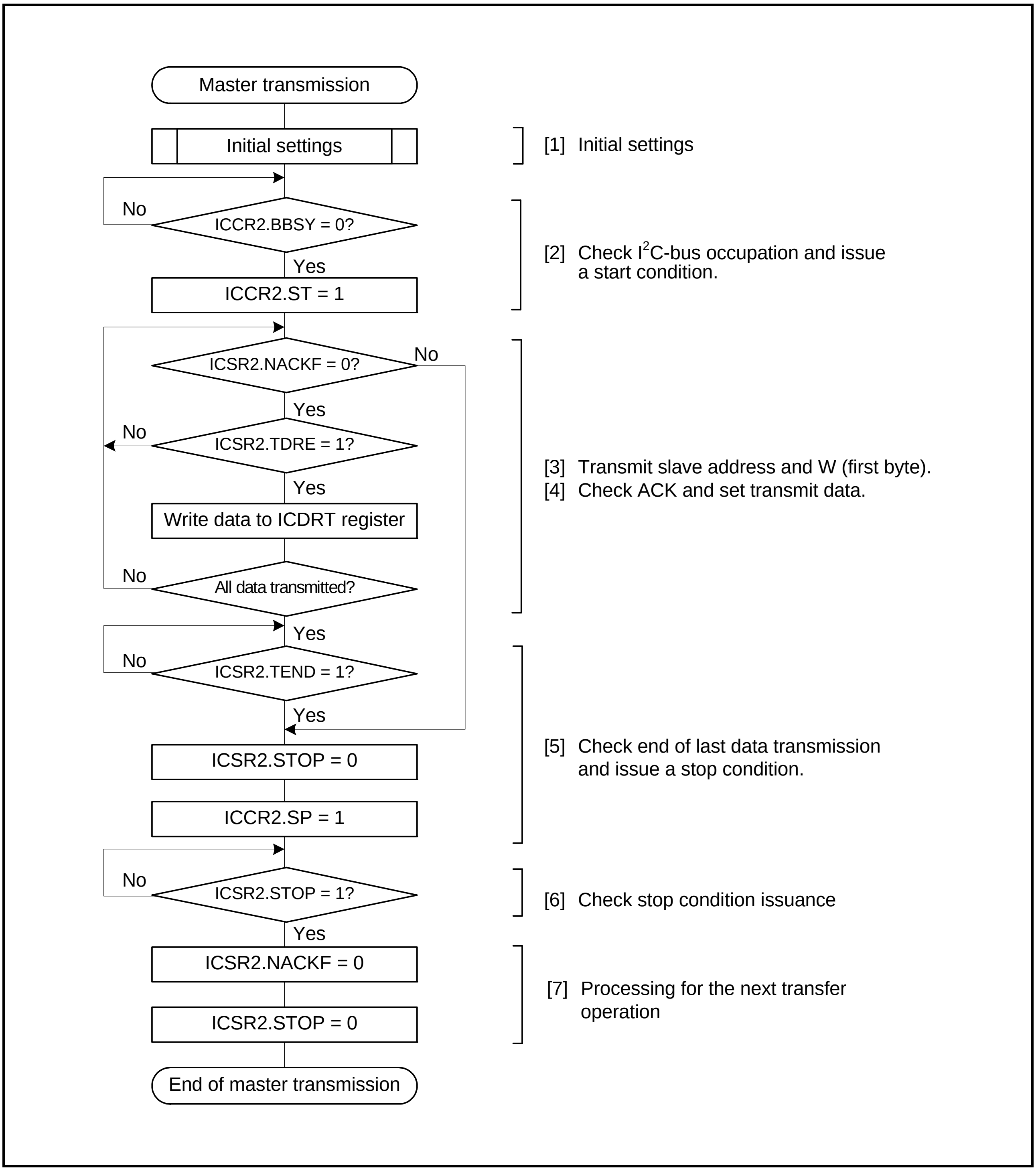


Figure 26.6 Example of Master Transmission Flowchart

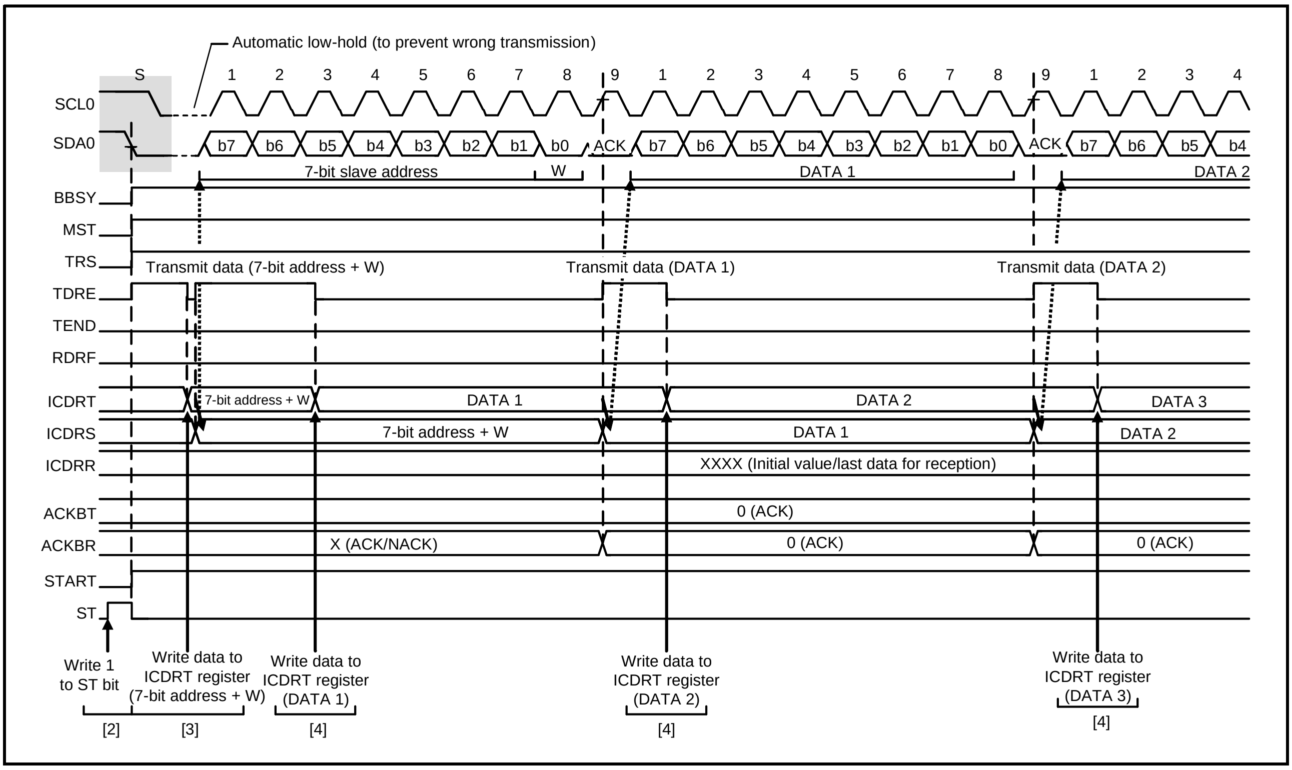


Figure 26.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

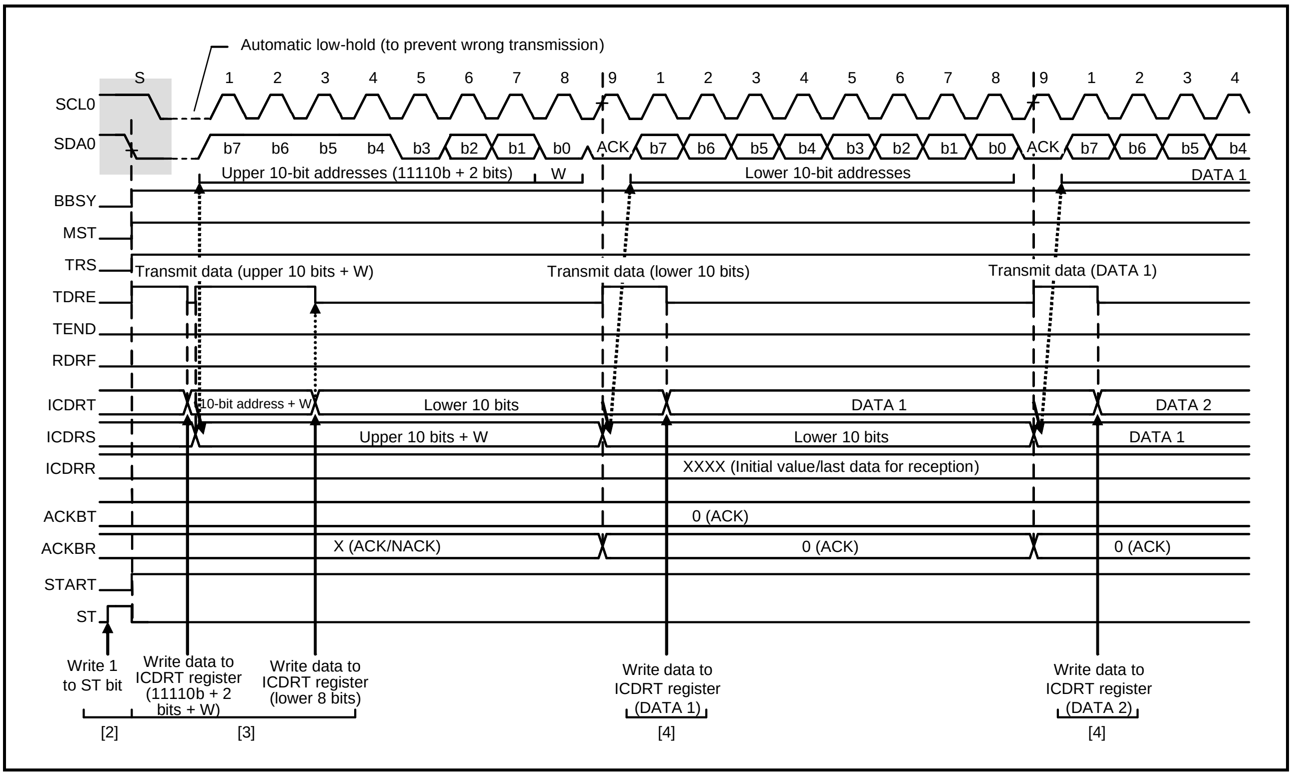


Figure 26.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

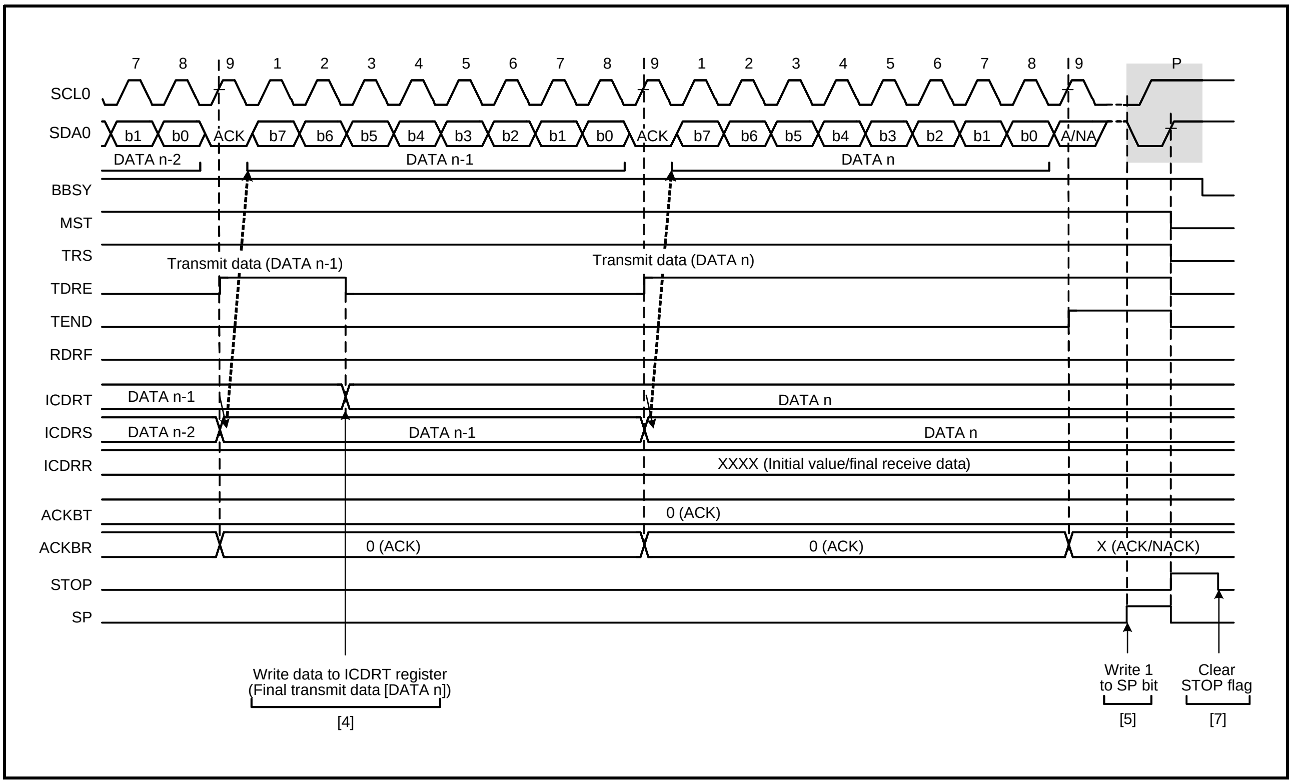


Figure 26.9 Master Transmit Operation Timing (3)

26.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. 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 26.10 and Figure 26.11 show examples of usage of master reception (7-bit address format) and Figure 26.12 to Figure 26.14 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 26.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (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 ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of SCL clock, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL clock and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (stop condition issuance request) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL0 line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

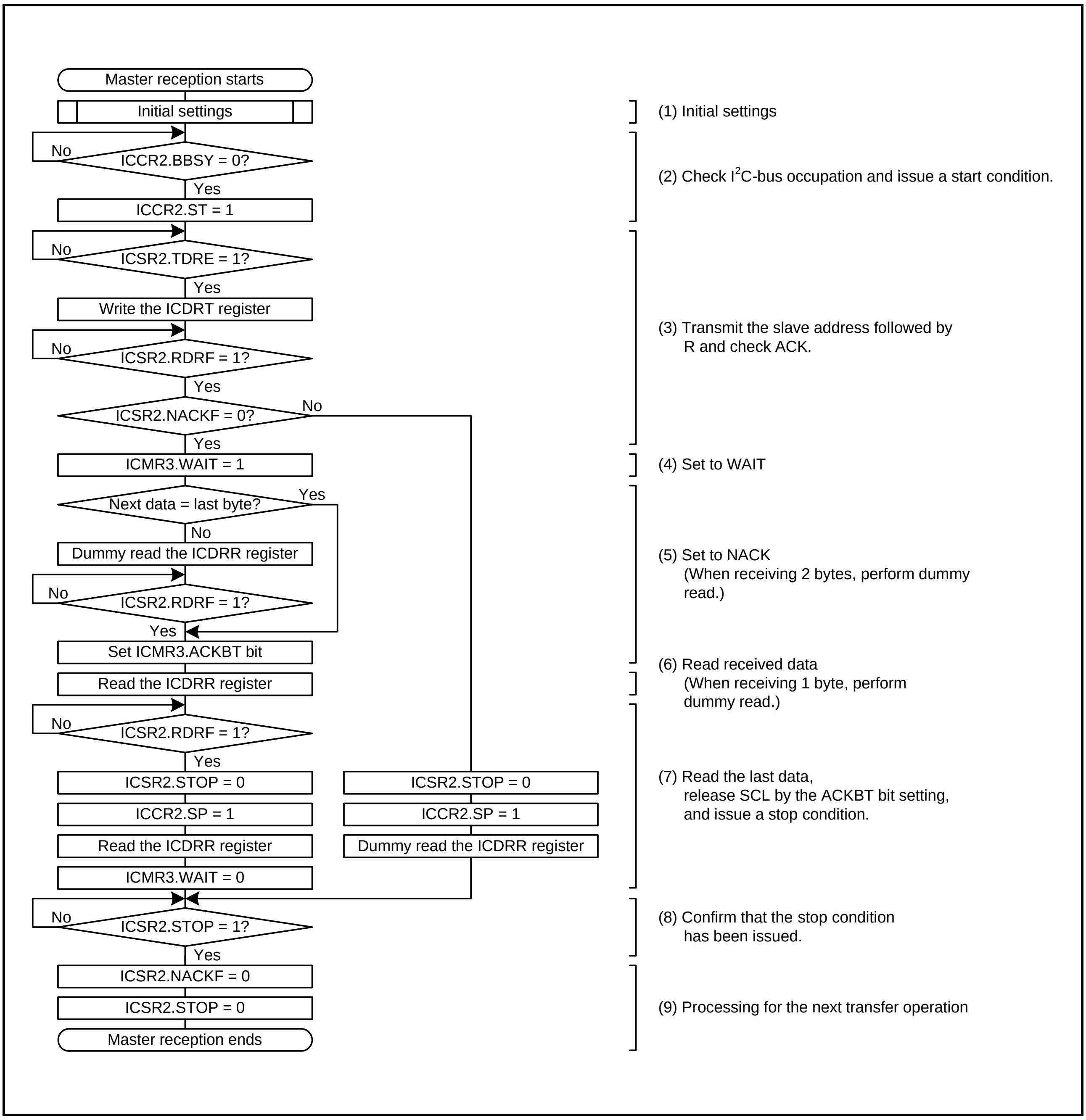


Figure 26.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

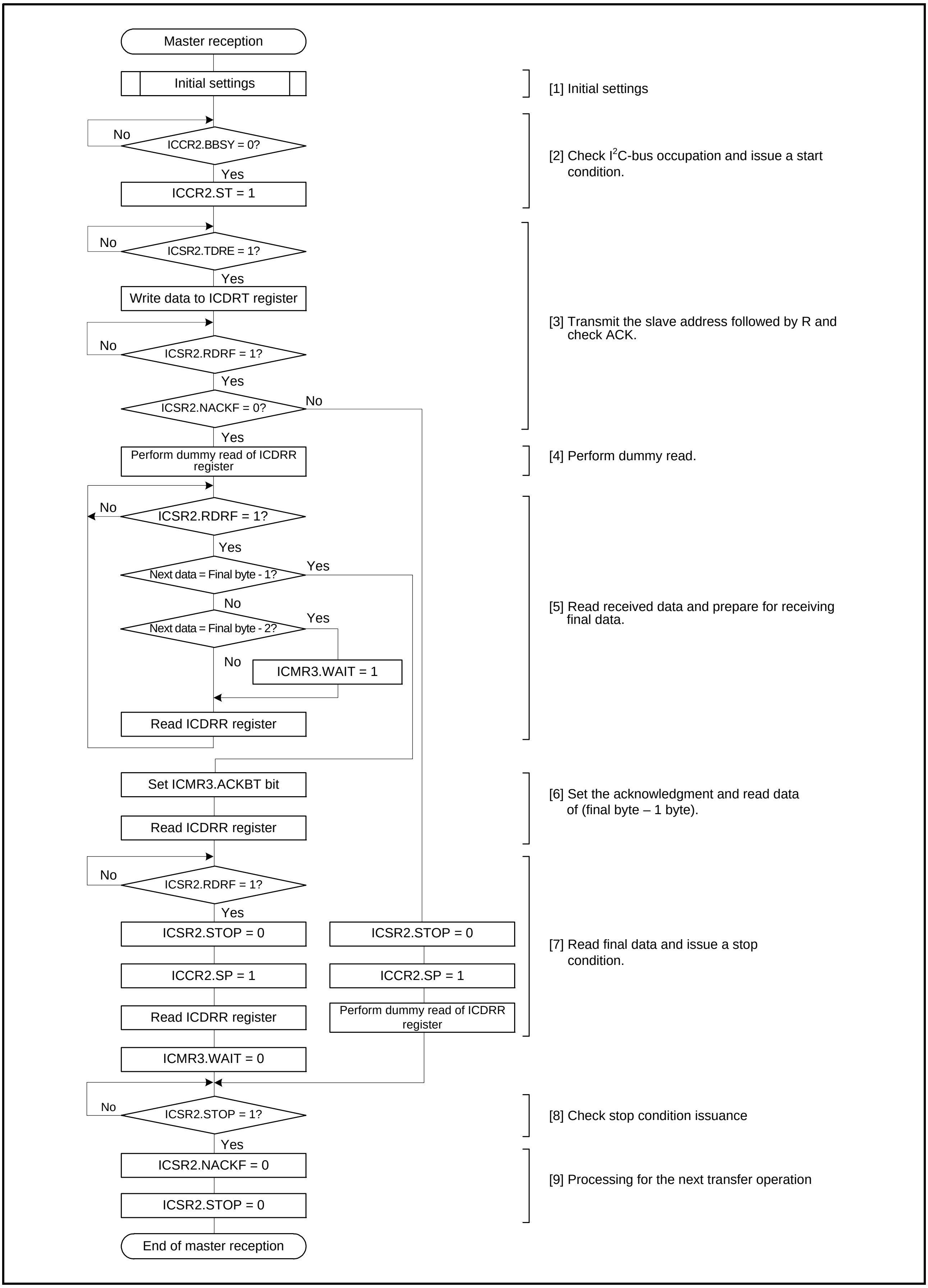


Figure 26.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

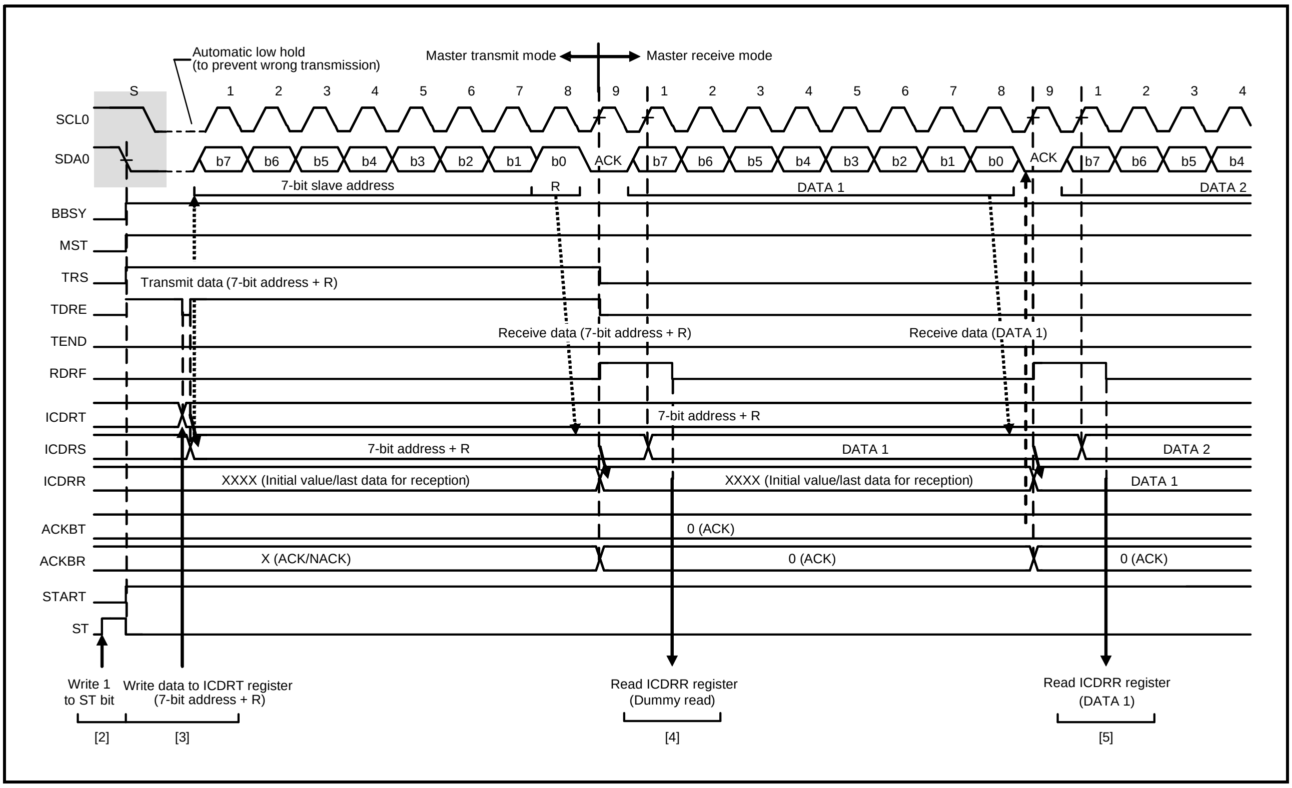


Figure 26.12 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

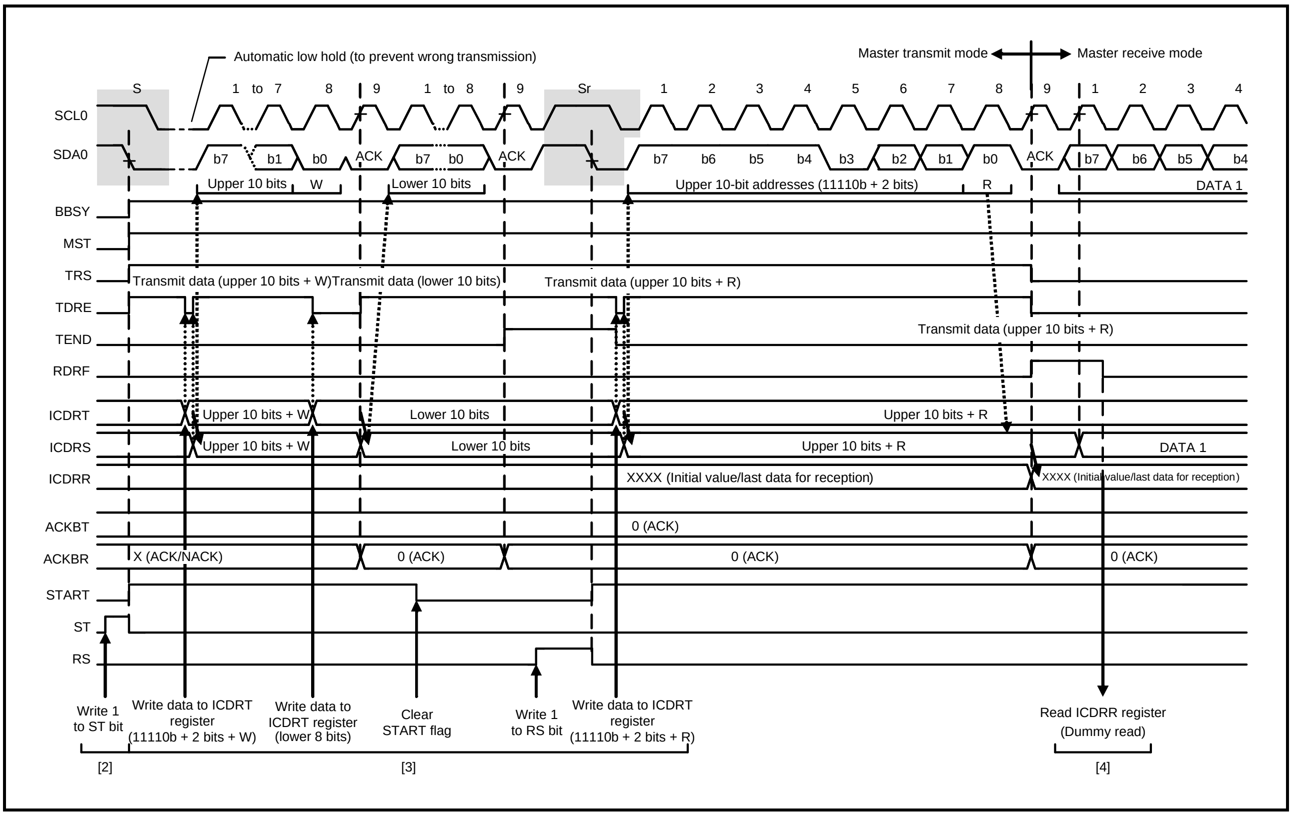


Figure 26.13 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

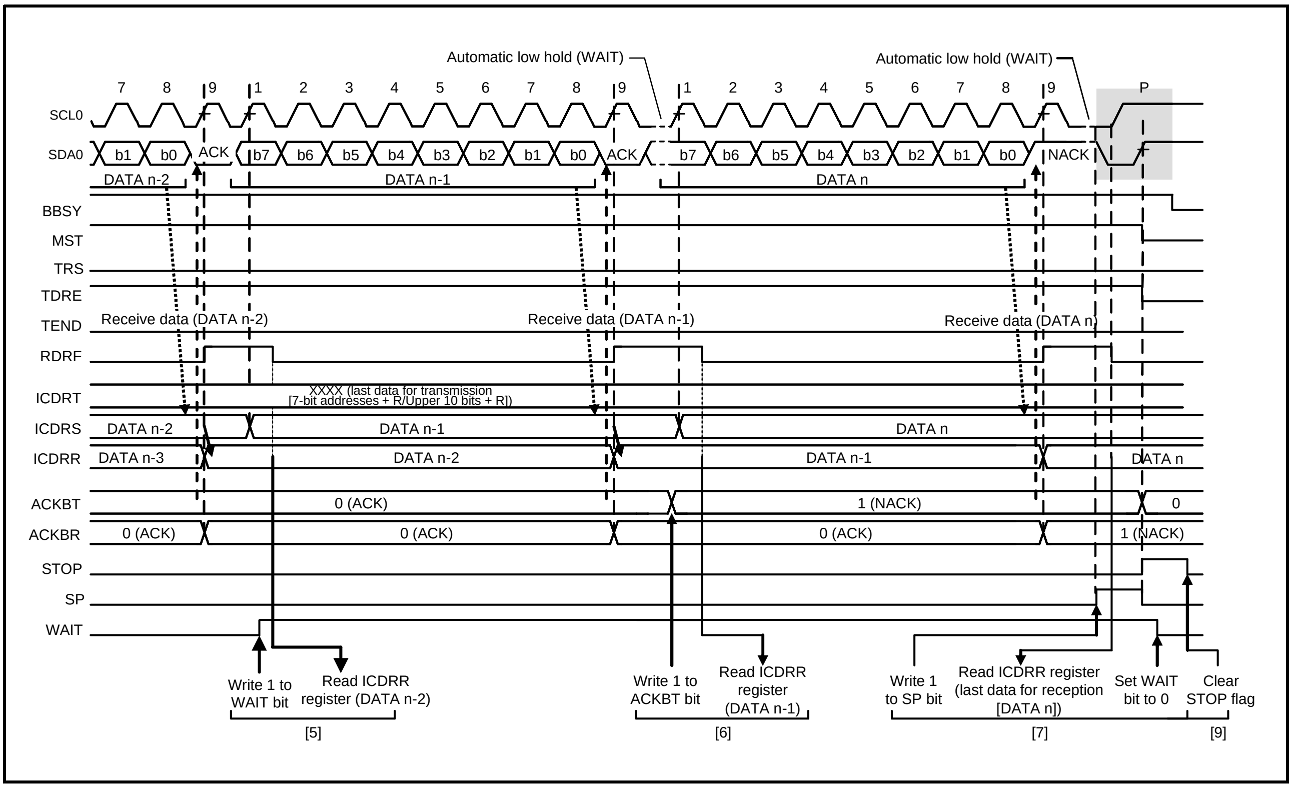


Figure 26.14 Master Receive Operation Timing (3) (When RDRFS bit is 0)

26.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 26.15 shows an example of usage of slave transmission and Figure 26.16 and Figure 26.17 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 26.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL0 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 the ICDRR register to complete the processing. This releases the SCL0 line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

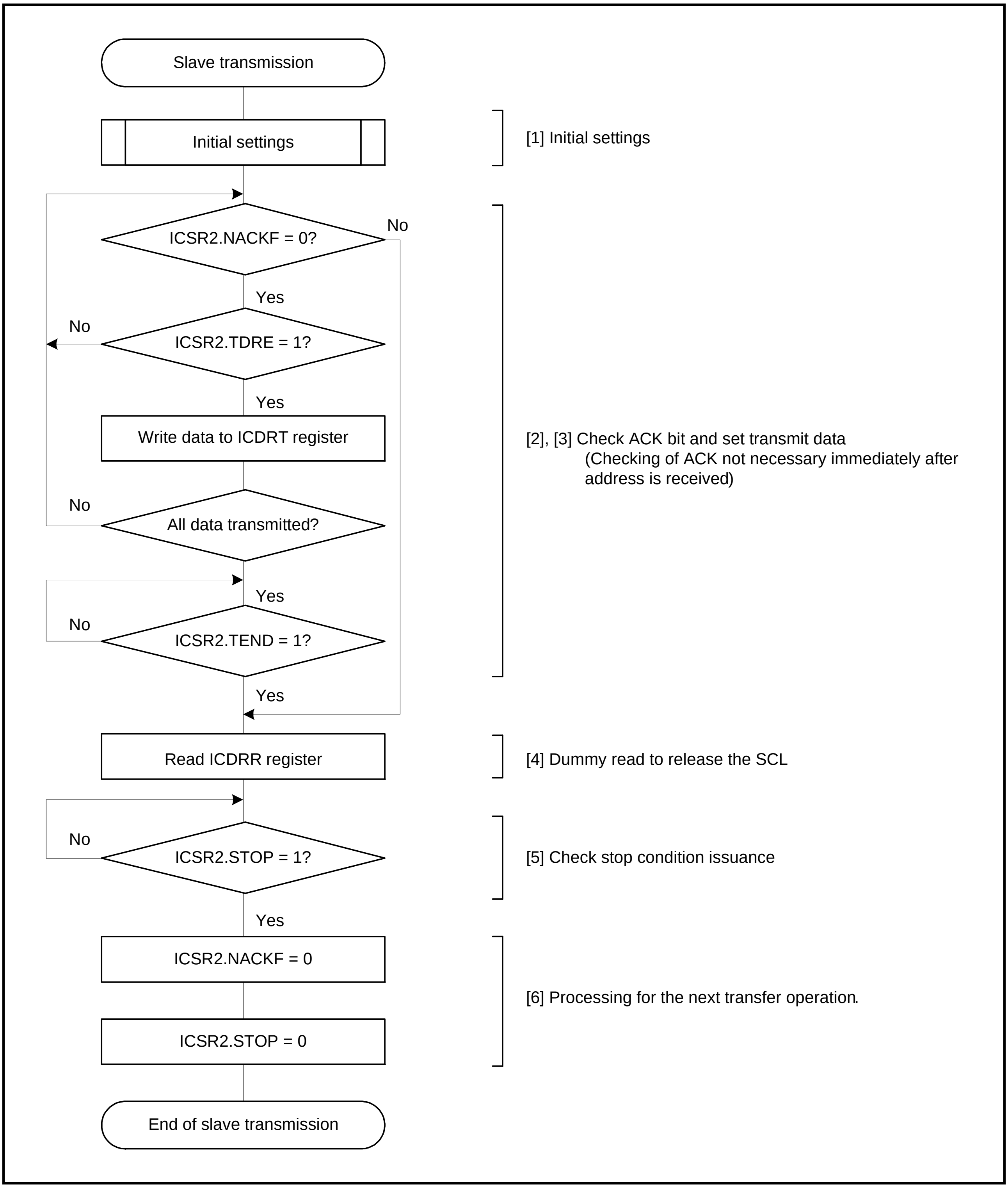


Figure 26.15 Example of Slave Transmission Flowchart

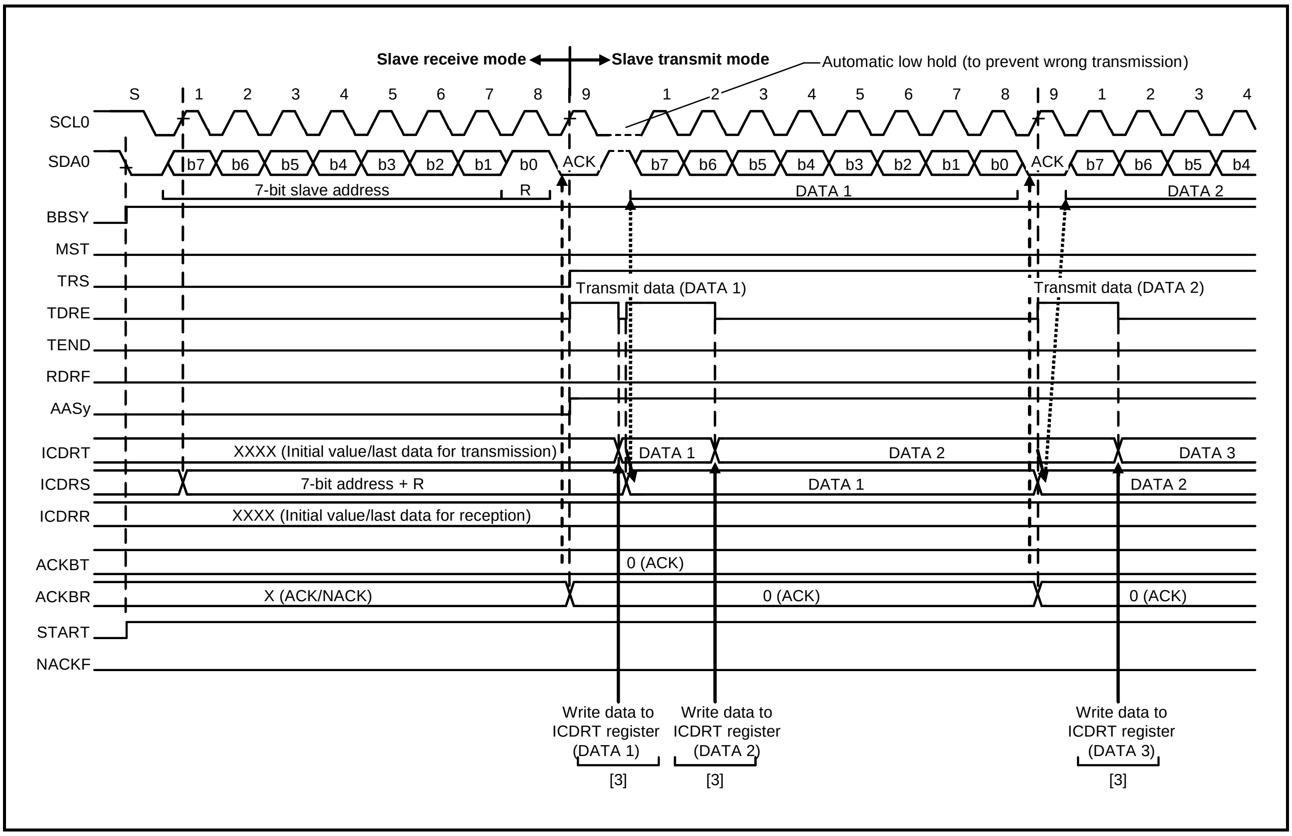


Figure 26.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

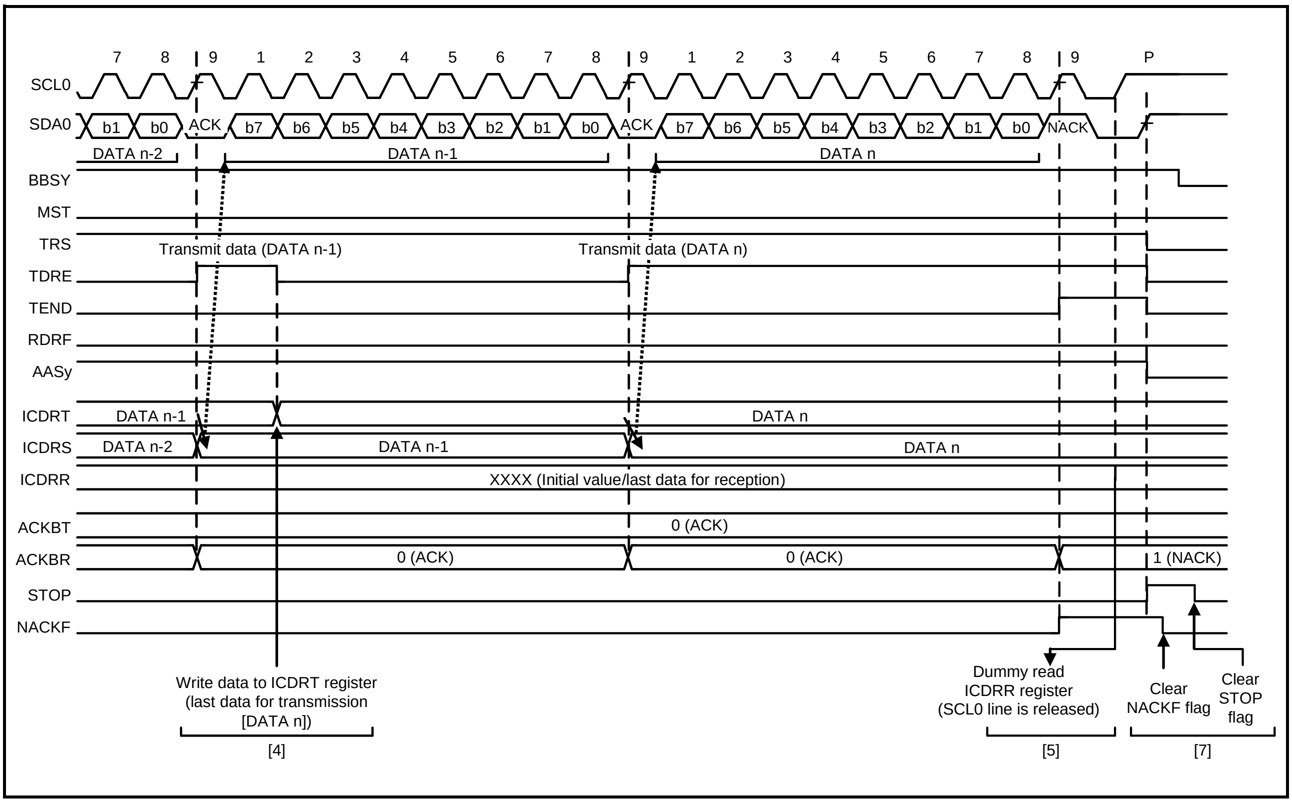


Figure 26.17 Slave Transmit Operation Timing (2)

26.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgments as a slave device.
Figure 26.18 shows an example of usage of slave reception and Figure 26.19 and Figure 26.20 show the timing of operations in slave reception.
The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 26.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read the ICDRR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).

(4) When the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCL0 line from being held at the low level.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read the ICDRR register until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

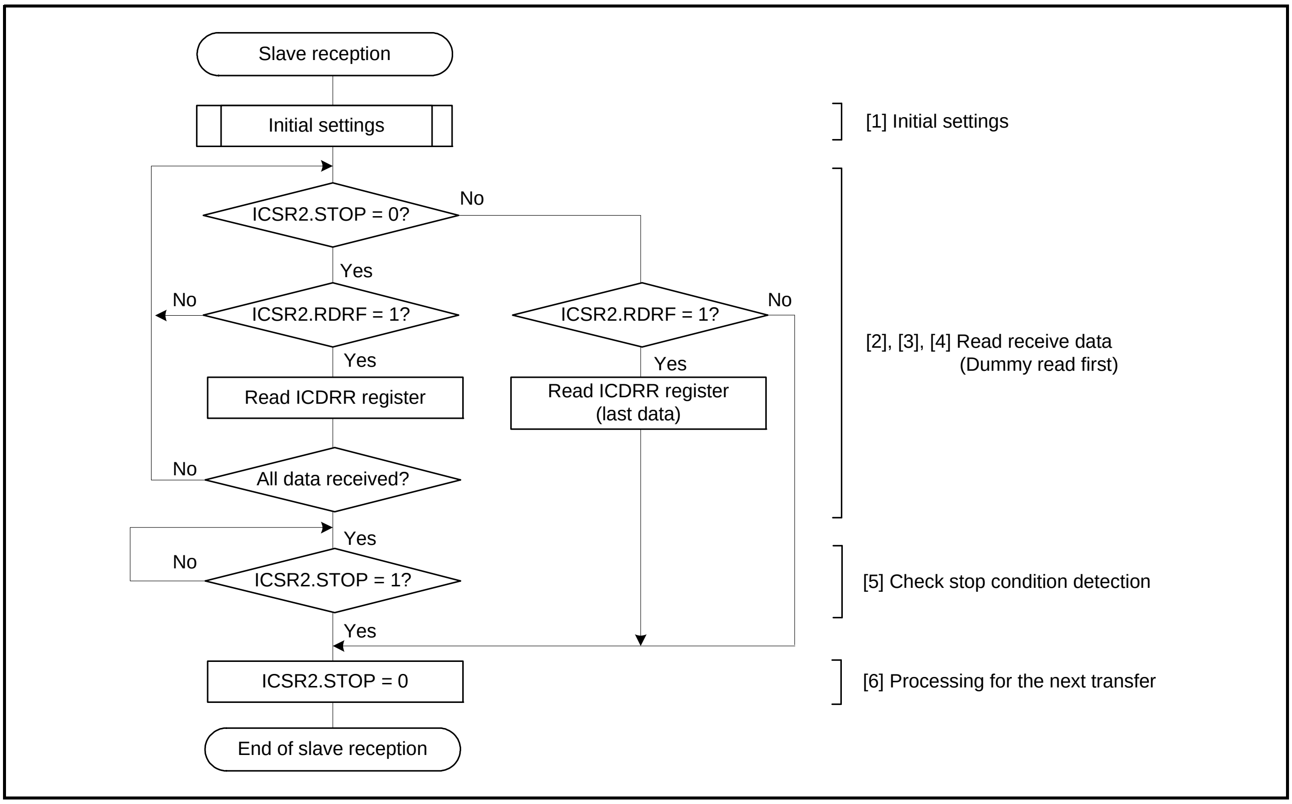


Figure 26.18 Example of Slave Reception Flowchart

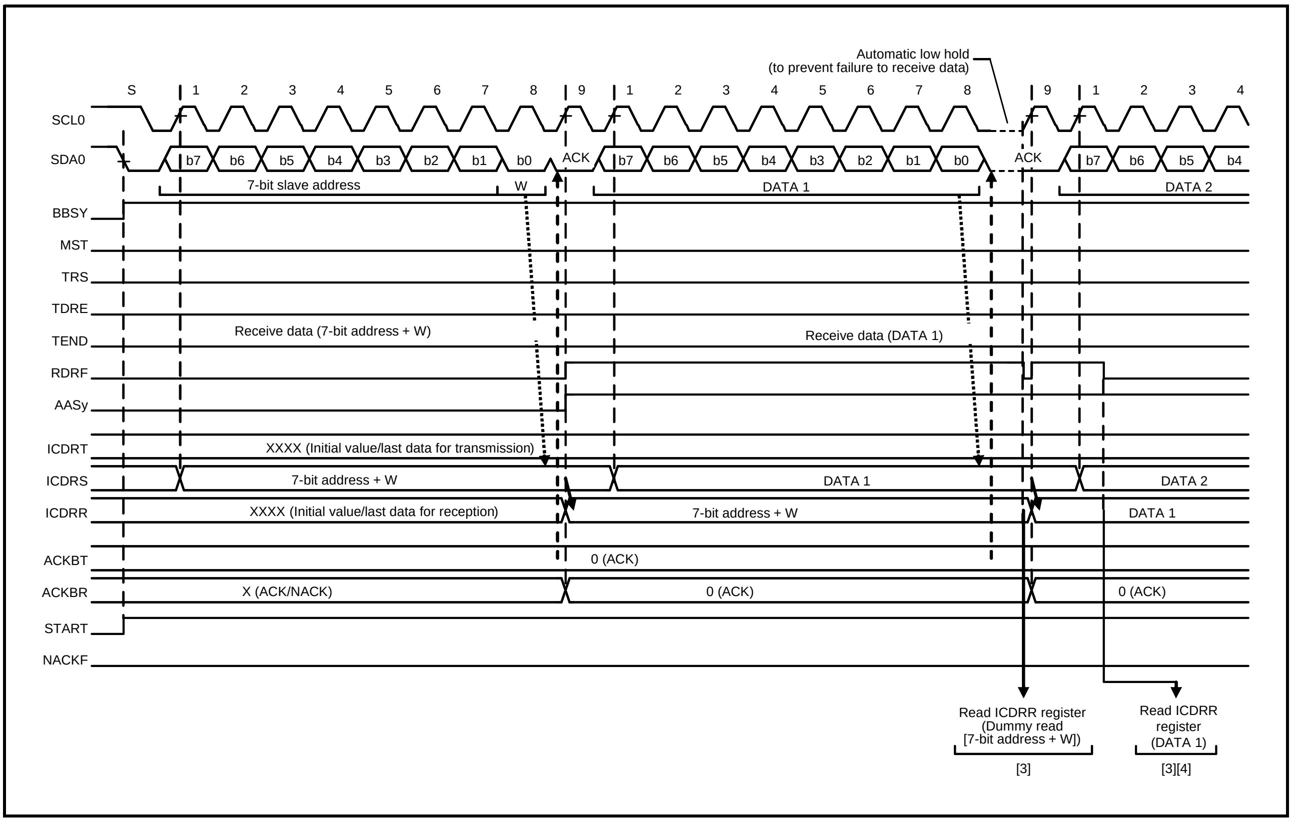


Figure 26.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

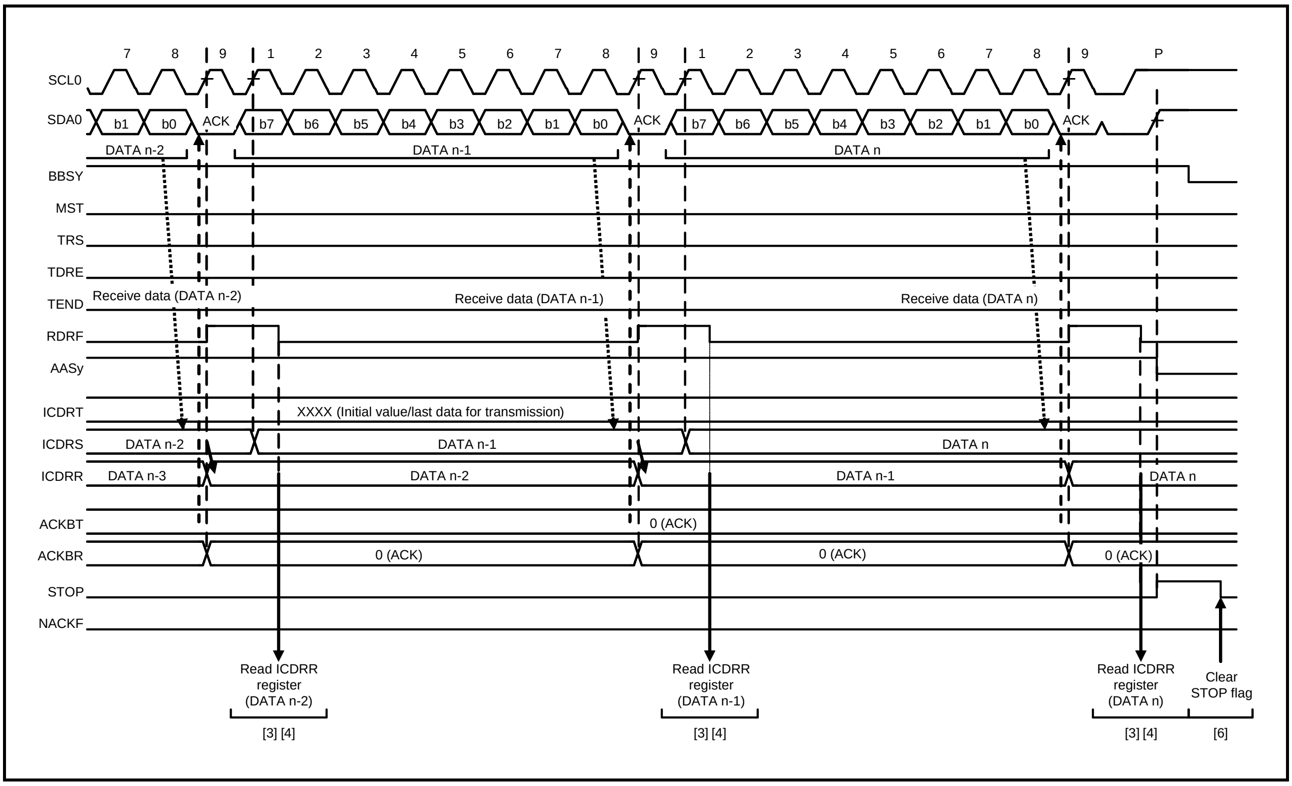


Figure 26.20 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

26.4 SCL Synchronization Circuit

In generation of the SCL clock, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low level period specified in the ICBRL register, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I^{2}C-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 SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 line 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 SCL0 line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL0 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 ICFER.SCLE bit is set to 1.

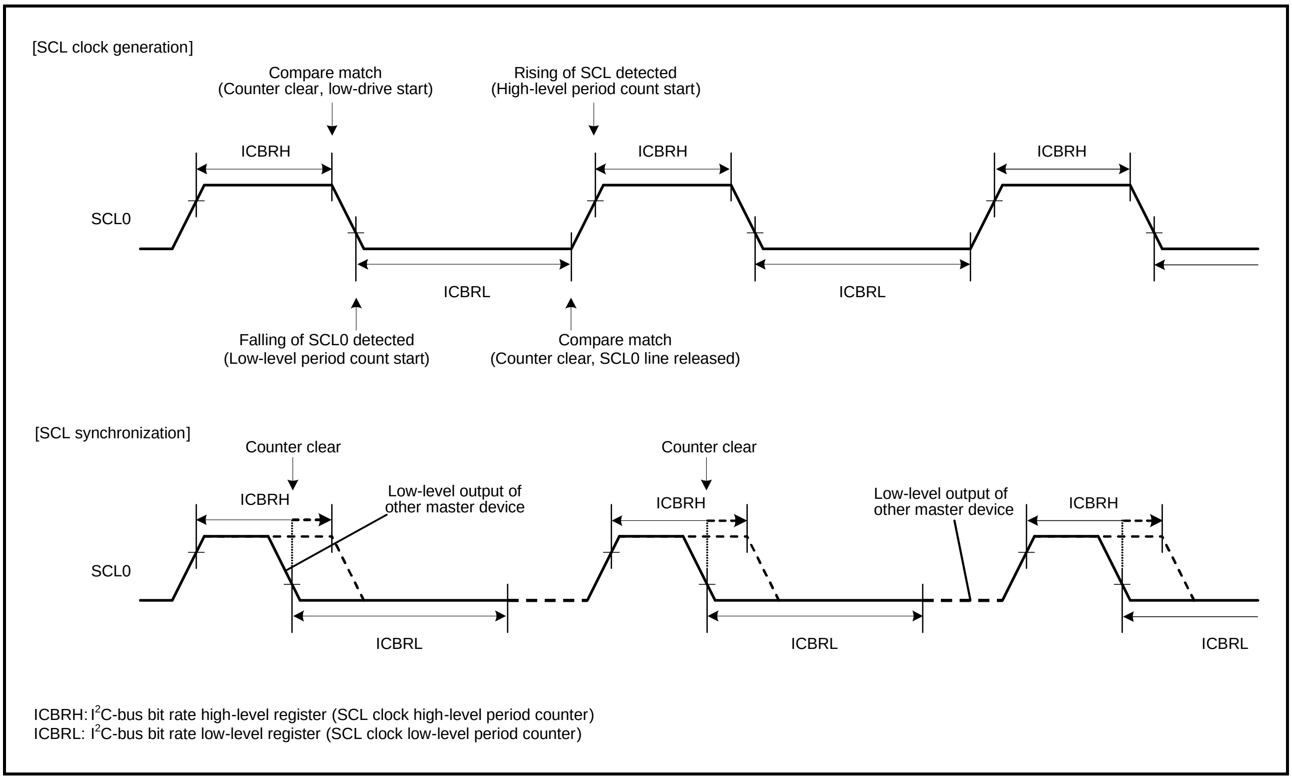


Figure 26.21 Generation and Synchronization of the SCL Signal from the RIIC

26.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay function, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay function is enabled (i.e. while the ICMR2.SDDL[2:0] bits are set to any value other than 000b), the ICMR2.DLCS bit 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 ICMR2.SDDL[2:0] bits. 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.

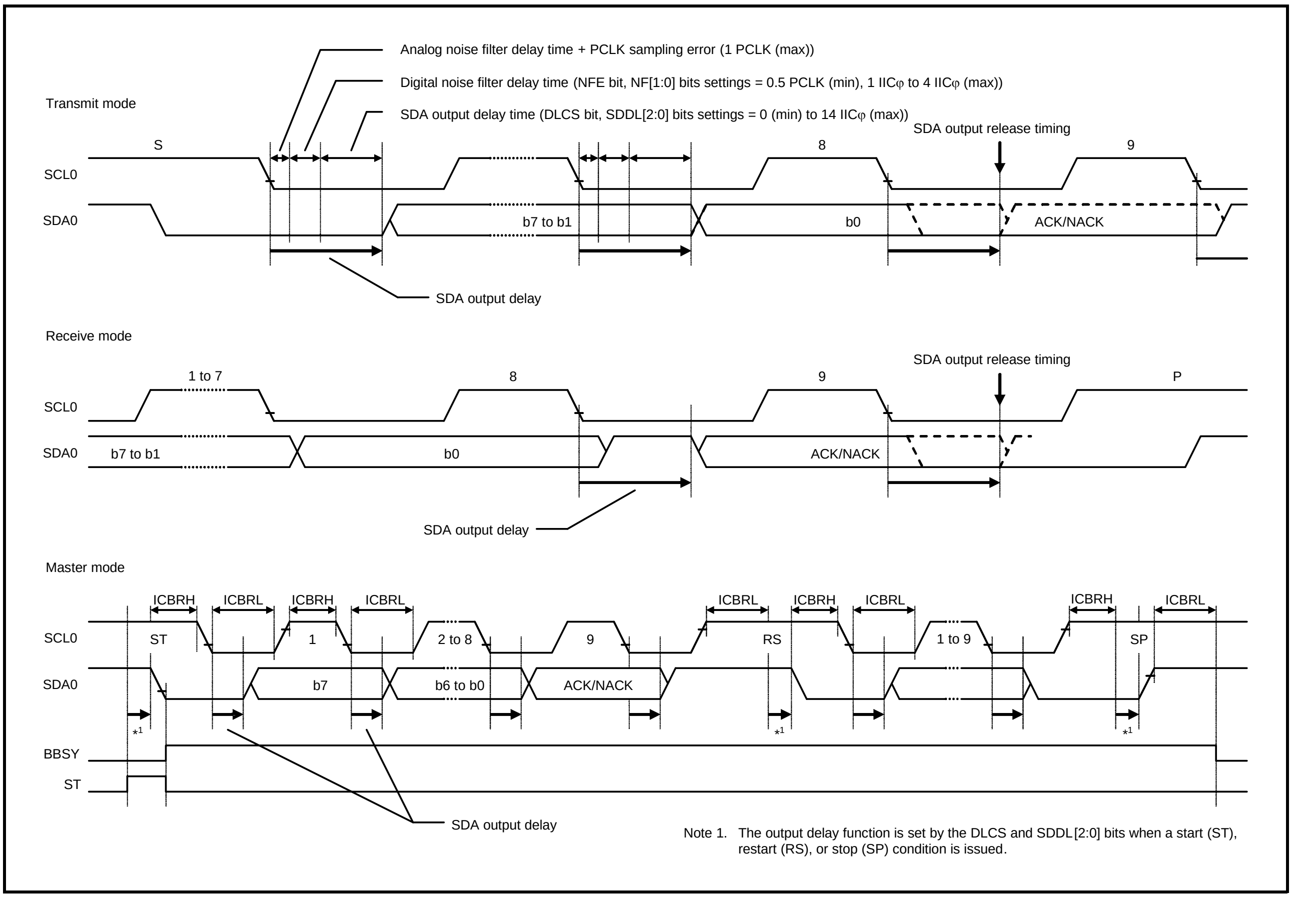


Figure 26.22 SDA Output Delay Function

26.6 Digital Noise Filter Circuits

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 26.23 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 ICMR3.NF[1:0] bits. 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 SCL0 pin (or SDA0 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 ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by setting the ICFER.NFE bit to 0) and use only the analog noise filter circuit.

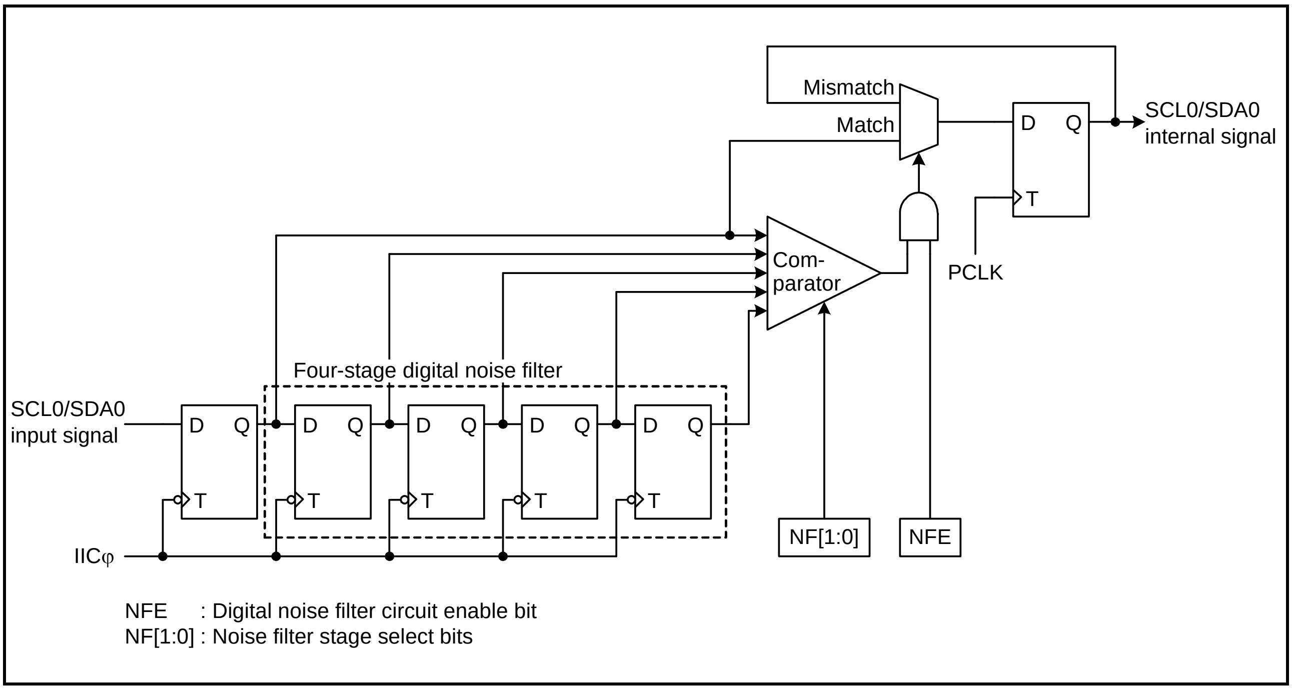


Figure 26.23 Block Diagram of Digital Noise Filter Circuit

26.7 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.

26.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the ICSR2.RDRF flag or the ICSR2.TDRE flag 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 26.24 to Figure 26.26 show the AASy flag set timing in three cases.

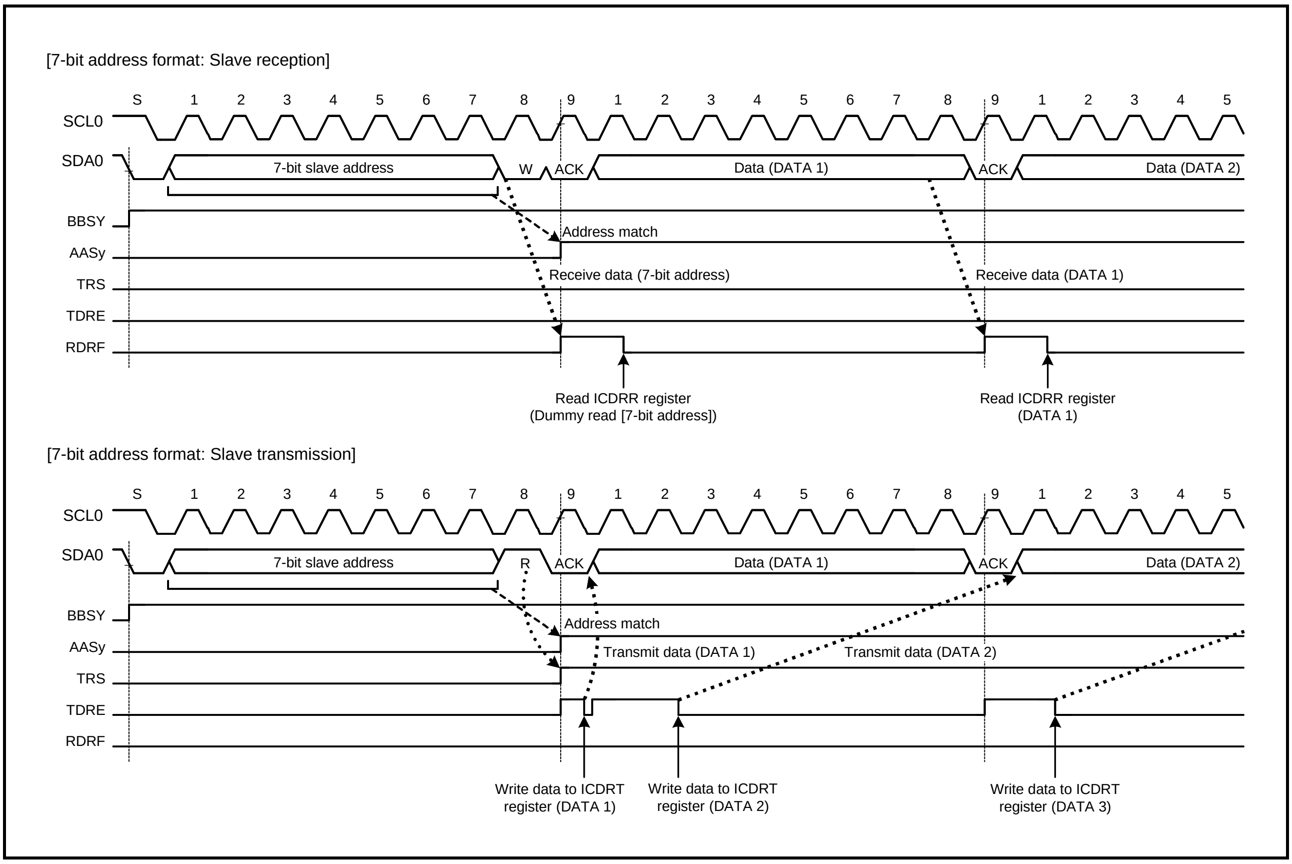


Figure 26.24 AASy Flag Set Timing with 7-Bit Address Format Selected

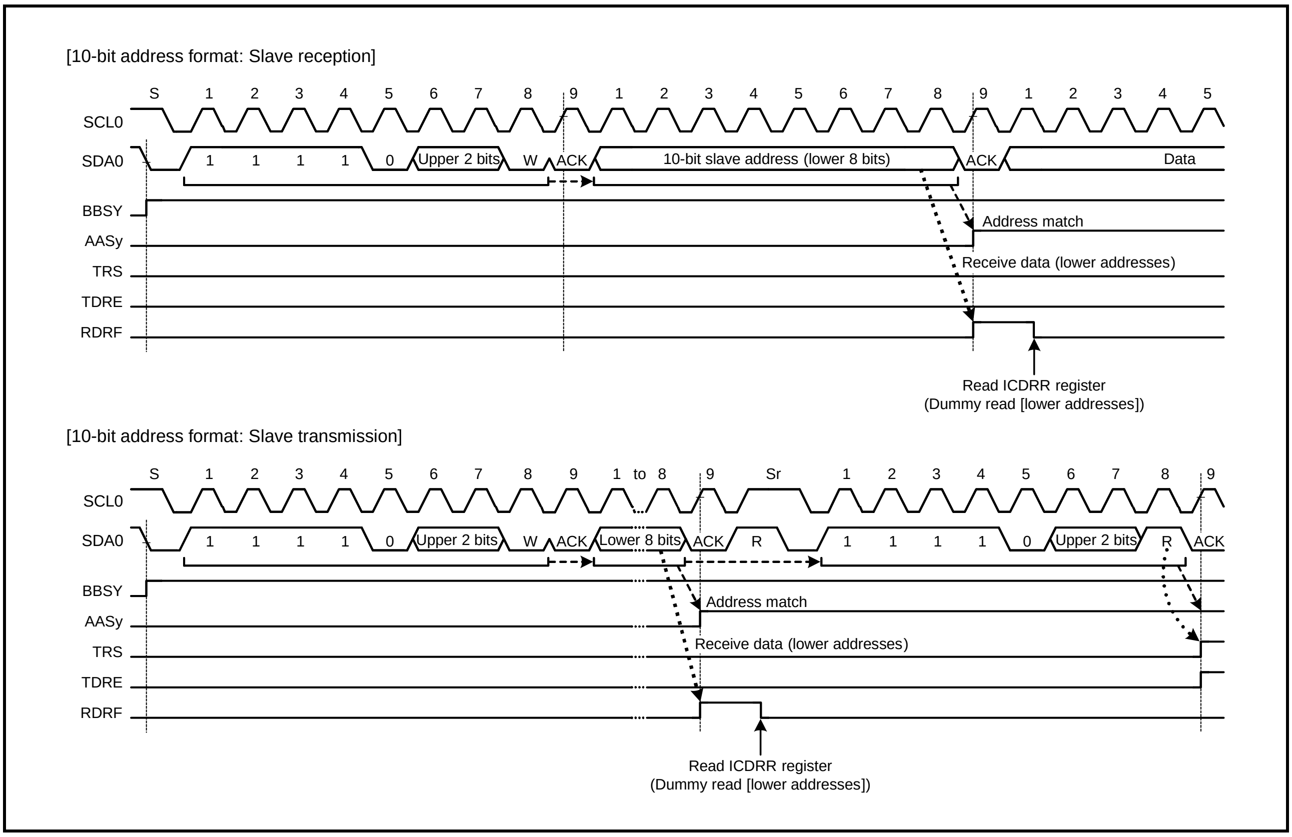


Figure 26.25 AASy Flag Set Timing with 10-Bit Address Format Selected

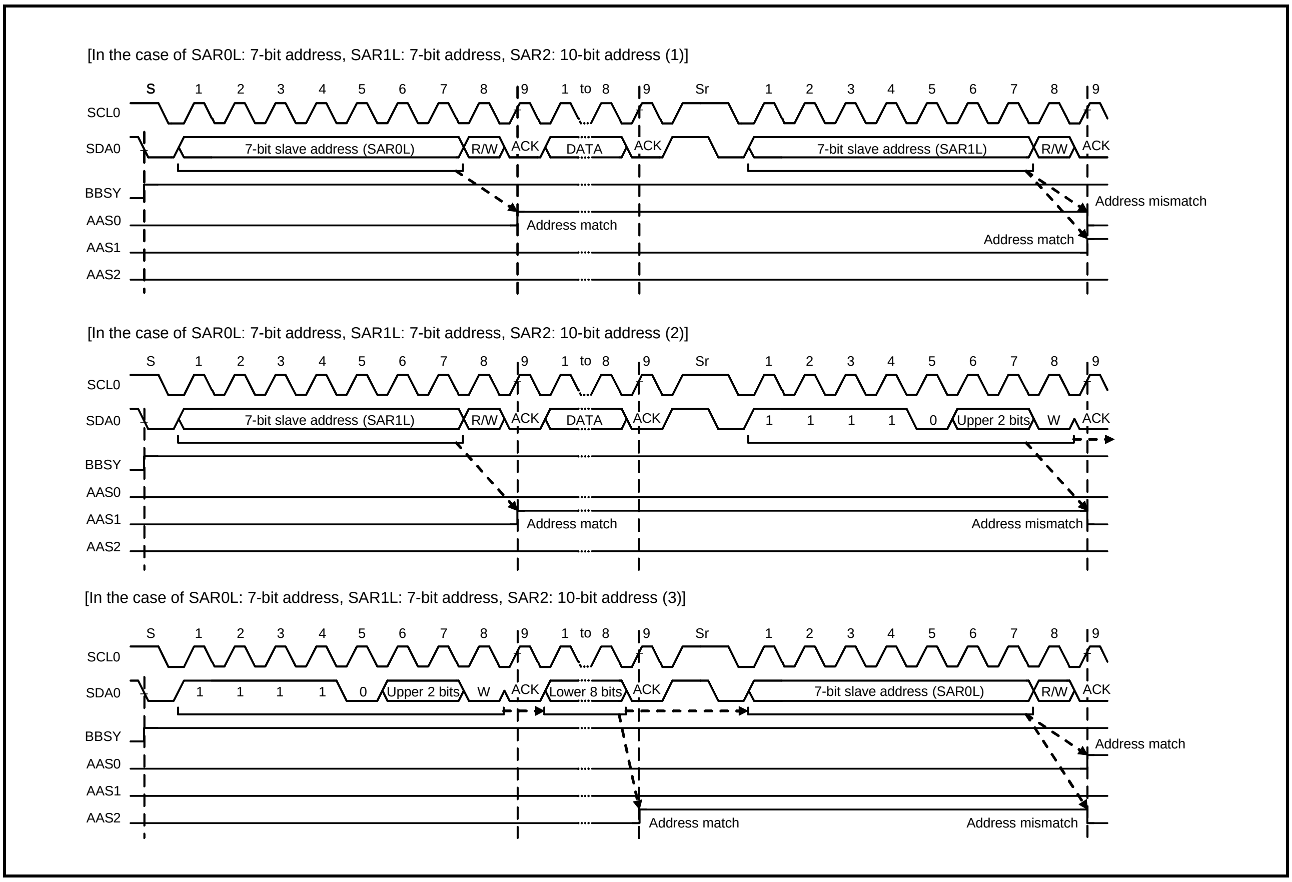


Figure 26.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

26.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth 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.

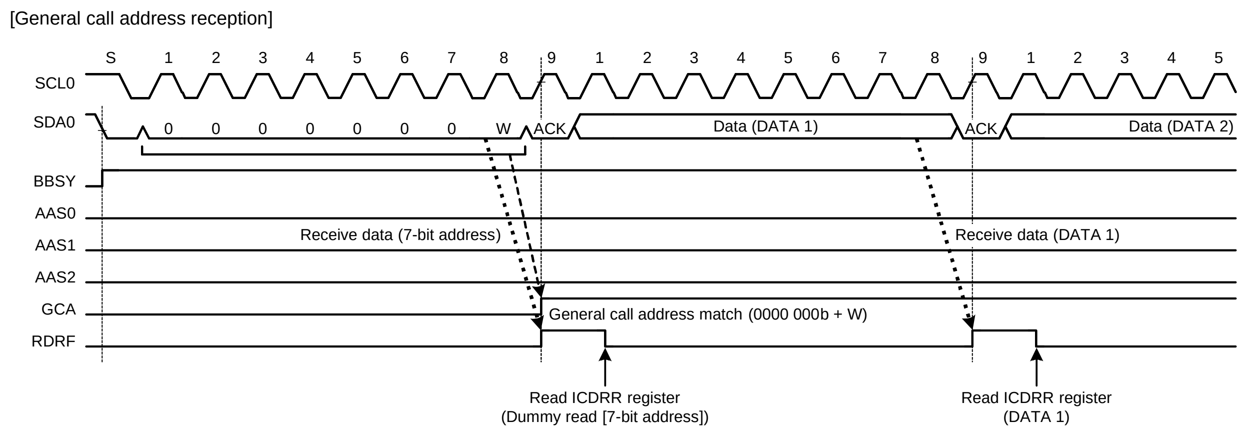


Figure 26.27 Timing of GCA Flag Setting during Reception of General Call Address

26.7.3 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 ICSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the ICSR1.DID flag to 1 on the rising edge of the eighth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.
After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.
In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.
Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

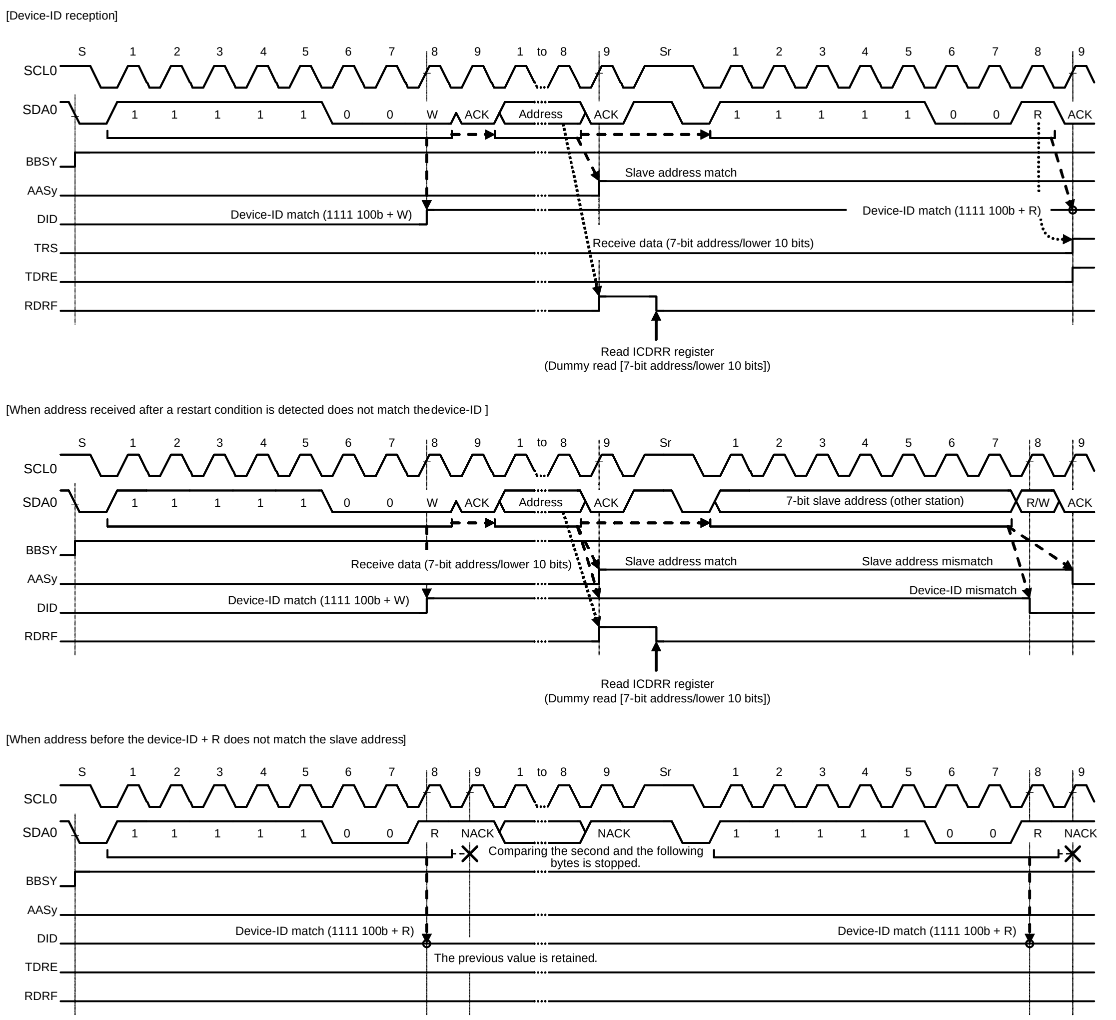


Figure 26.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

26.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).
When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.
If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

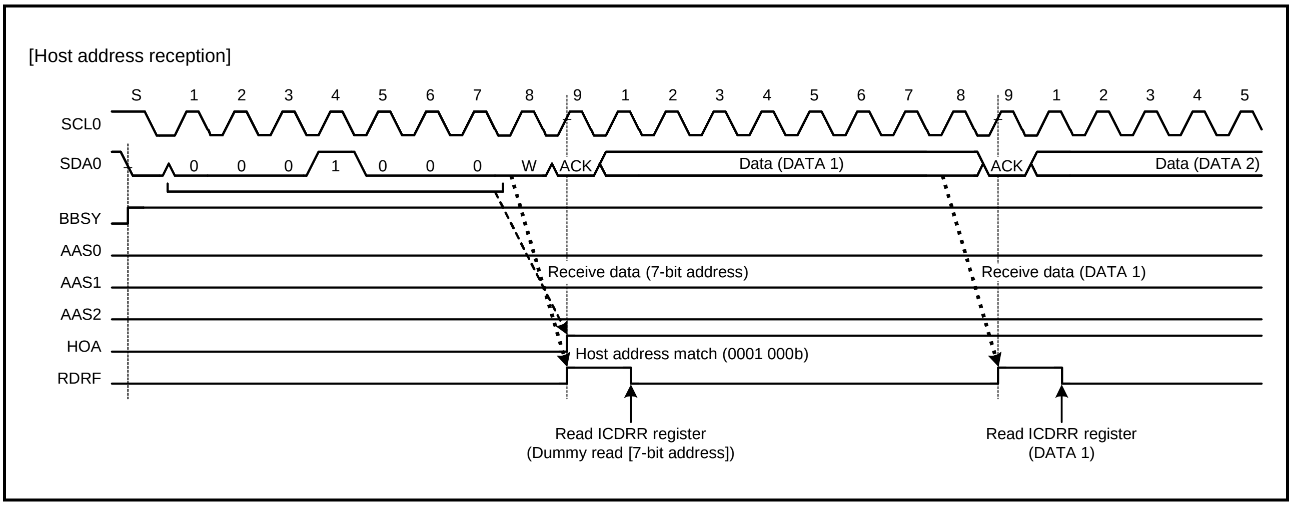


Figure 26.29 HOA Flag Set Timing during Reception of Host Address

26.8 Automatic Low-Hold Function for SCL

26.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I^2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCL0 line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

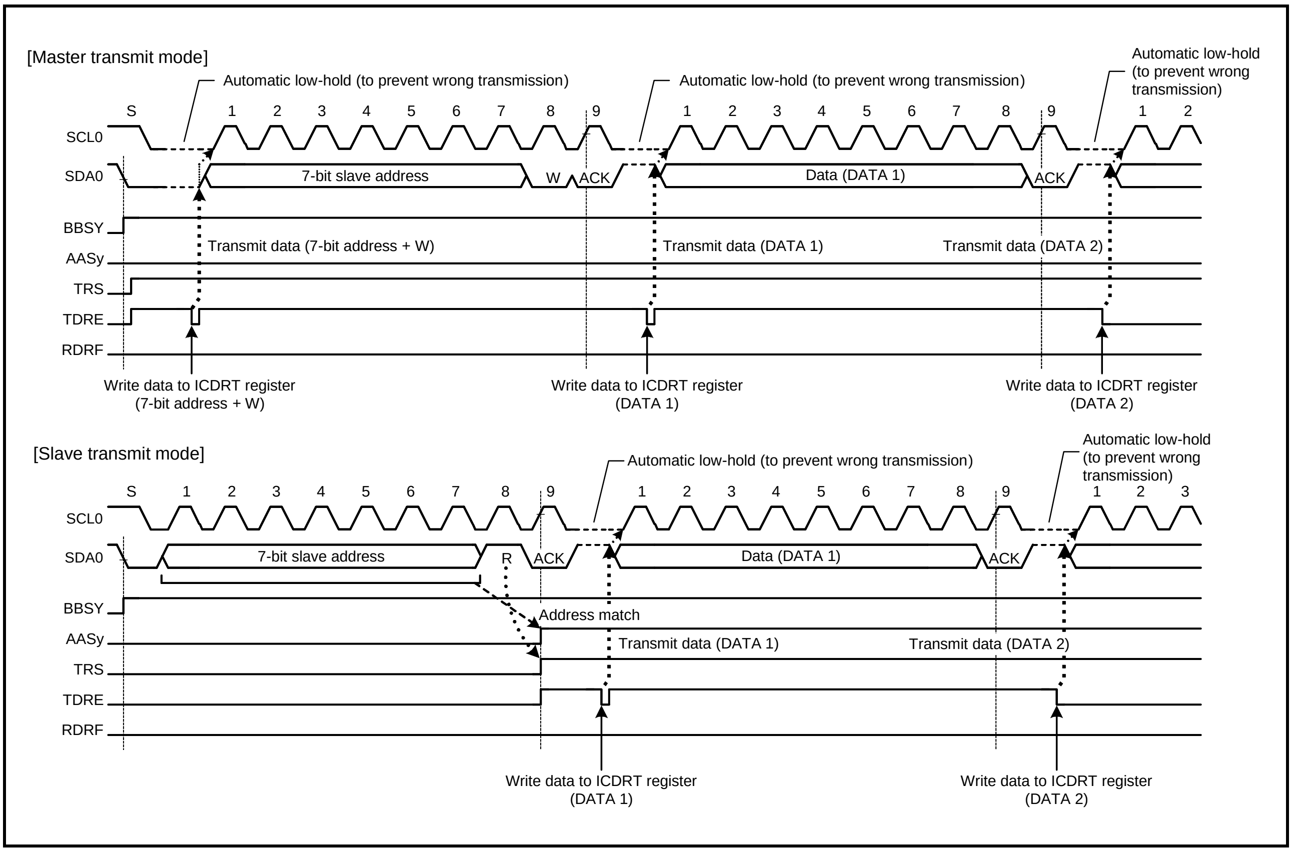


Figure 26.30 Automatic Low-Hold Operation in Transmit Mode

26.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA0 line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (ICSR2.NACKF flag is 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, after setting the NACKF flag to 0, issue a restart condition, or issue a stop condition and then issue a start condition again.

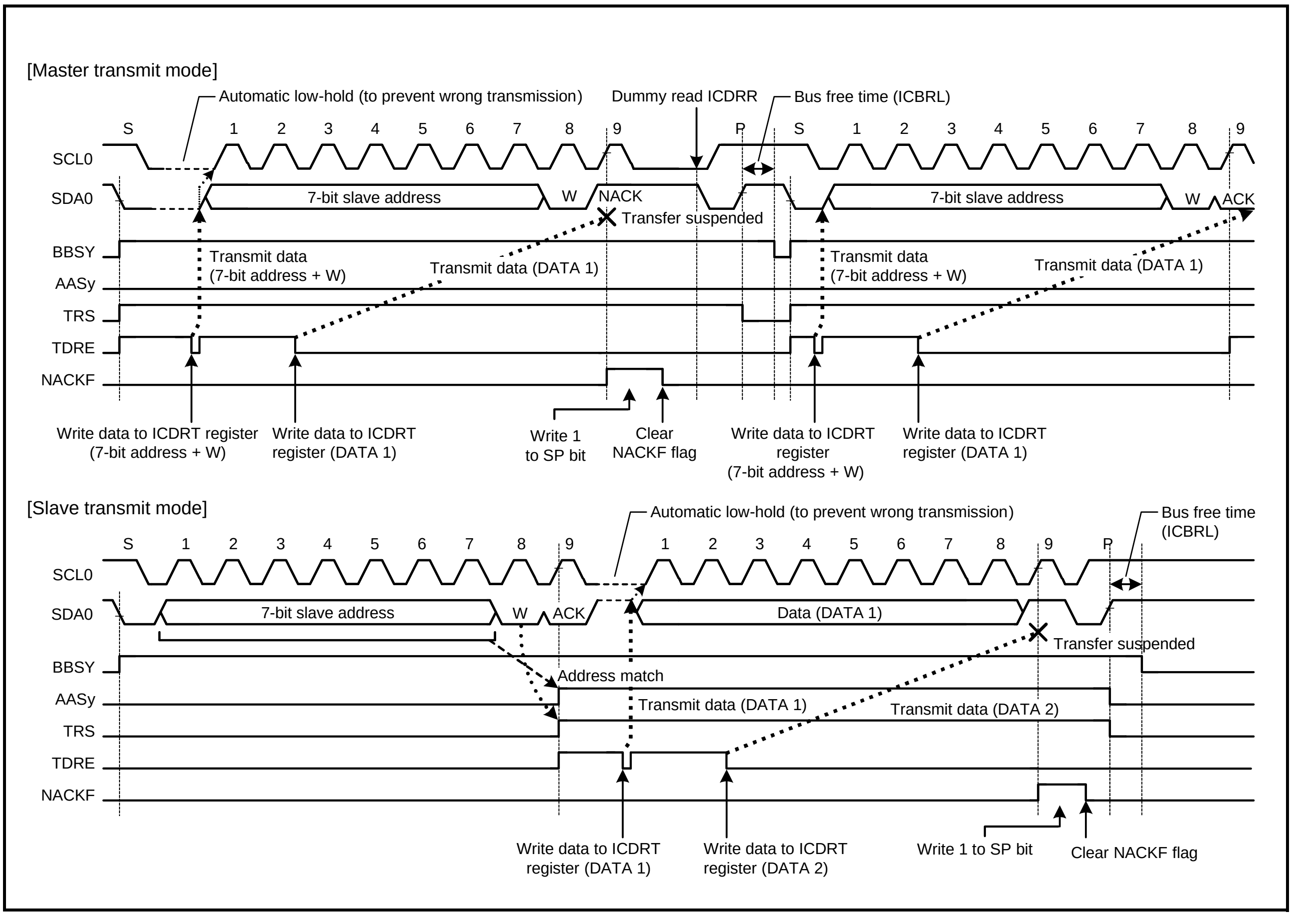


Figure 26.31 Suspension of Data Transfer When NACK is Received (NACKE = 1)

26.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the 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 SCL0 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 the ICDRR register, which enables bytewise receive operation.

The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL0 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 ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

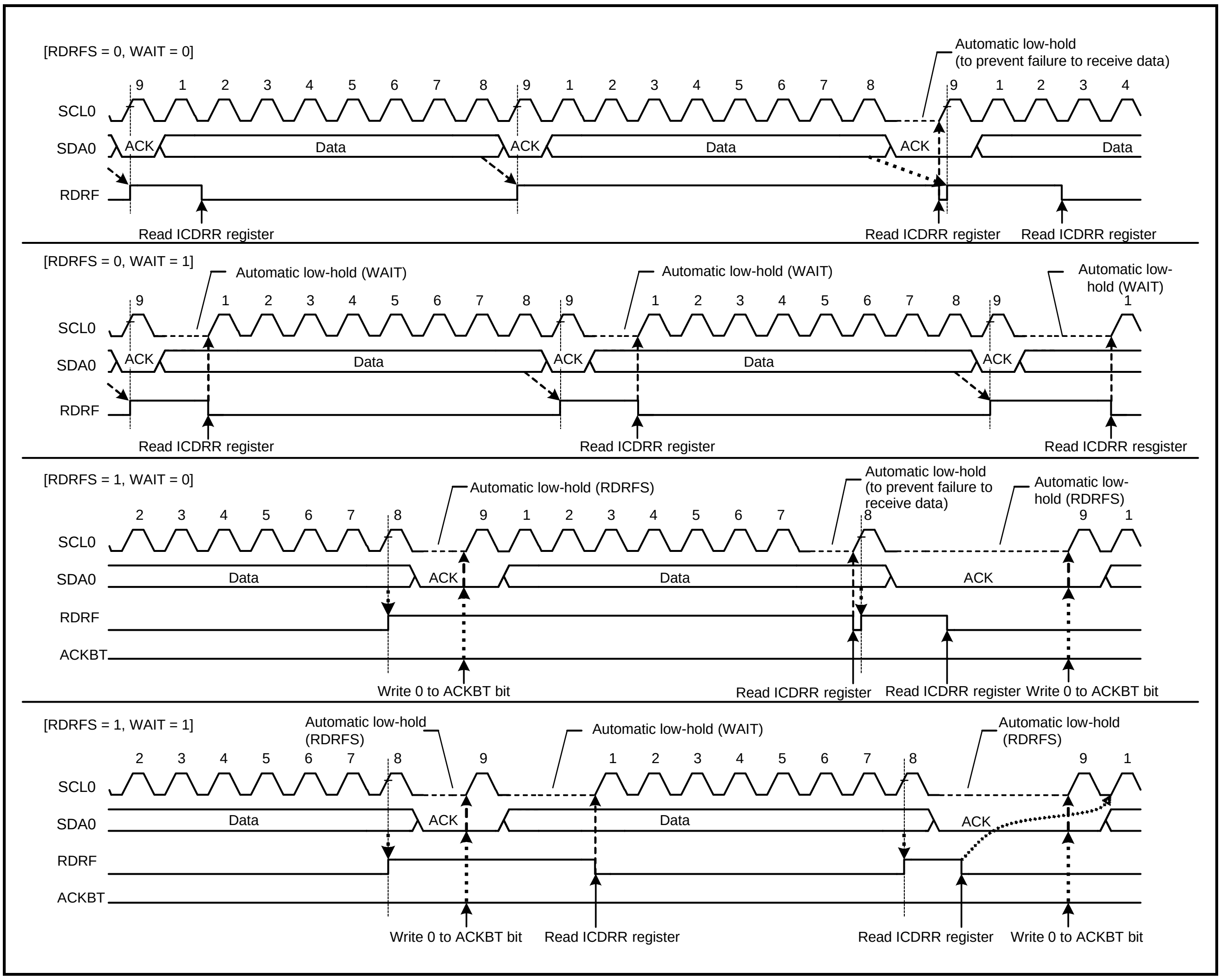


Figure 26.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

26.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

26.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to issue a start condition. However, if the SDA0 line has already been driven low by another master device issuing a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.
When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 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 ICFER.MALE bit 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 SDA0 line after a start condition was issued by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous issuing of a start condition)
- Setting of the ICCR2.ST bit 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 SDA0 line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

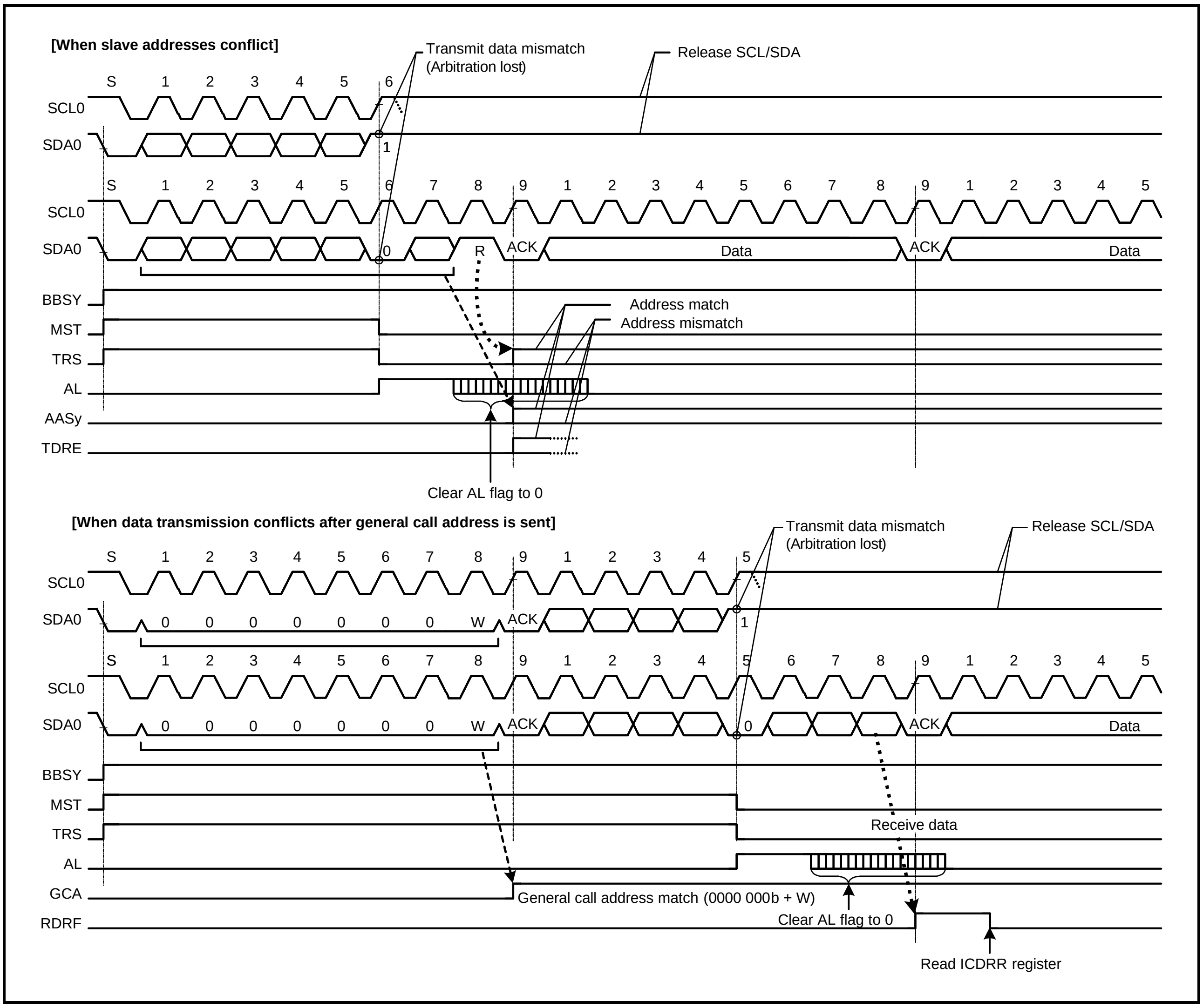


Figure 26.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

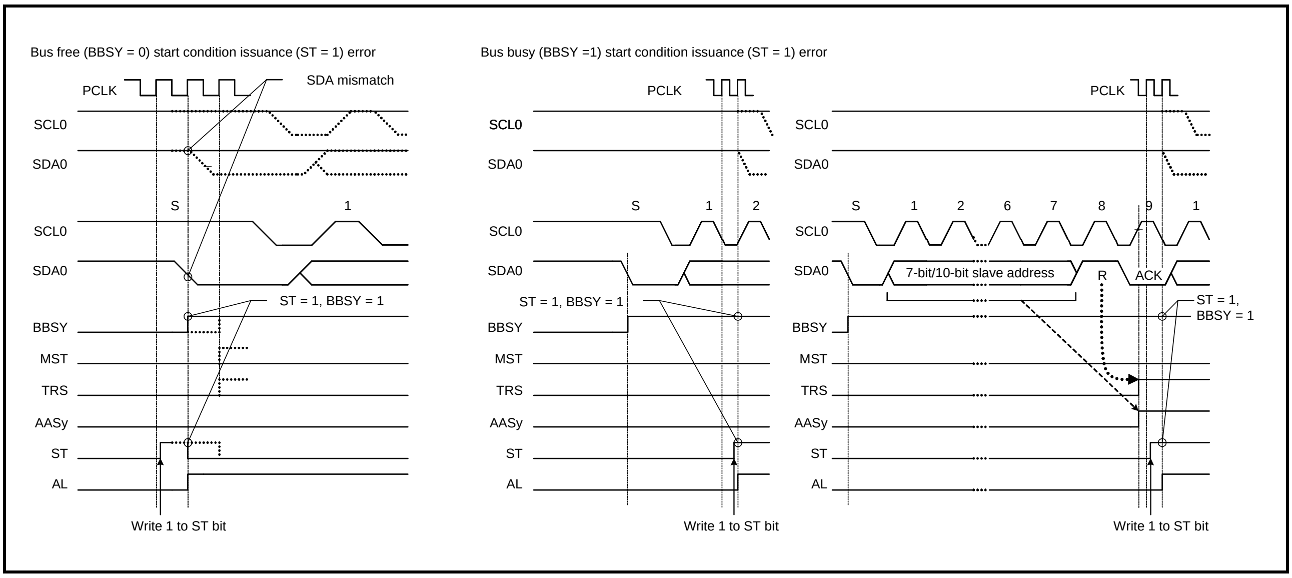


Figure 26.34 Arbitration-Lost When a Start Condition is Issued (MALE = 1)

26.9.2 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 SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 26.35 shows an example of arbitration-lost detection during transmission of NACK.

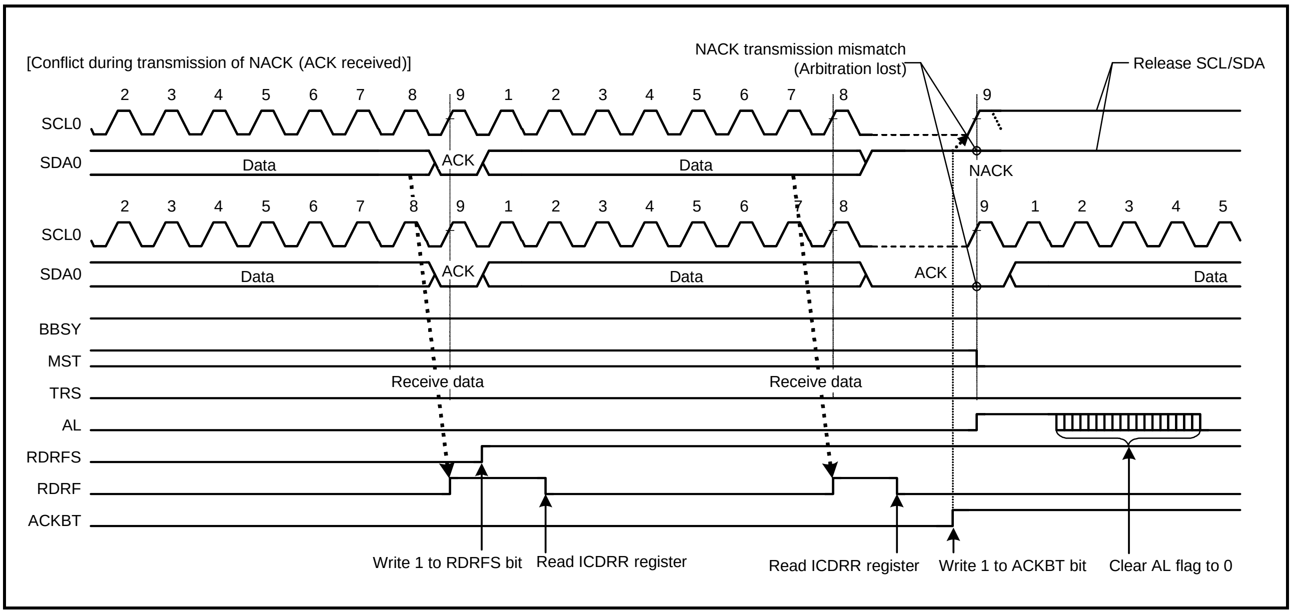


Figure 26.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the ICFER.NALE bit 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 SDA0 line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

26.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low level is detected on the SDA0 line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When it loses slave arbitration, the RIIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of FFh).

The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit 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 SDA0 line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

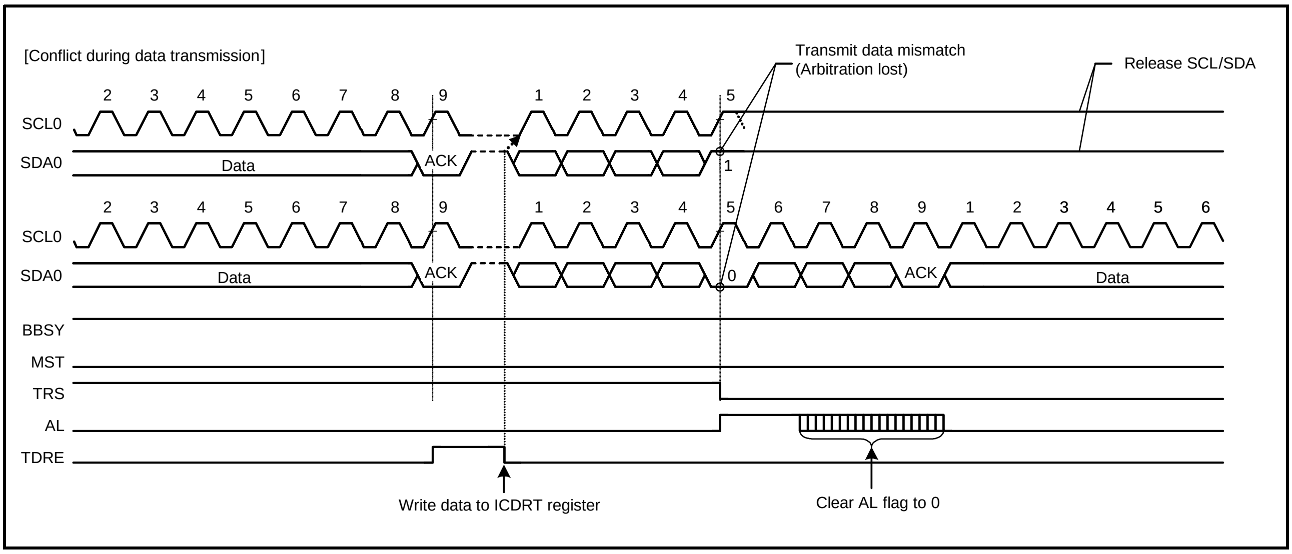


Figure 26.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

26.10 Start Condition/Restart Condition/Stop Condition Issuing Function

26.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ICCR2.ST bit 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 ICCR2.BBSY flag is 0 (bus free state). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.
A start condition is issued in the following sequence.

[Start condition issuance]
(1) Drive the SDA0 line low (high level to low level).
(2) Ensure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCL0 line low (high level to low level).
(4) Detect low level of the SCL0 line and ensure the low-level period of SCL0 line set in the ICBRL register.

26.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the ICCR2.RS bit 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 ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is issued in the following sequence.

[Restart condition issuance]
(1) Release the SDA0 line.
(2) Ensure the low-level period of SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low level to high level).
(4) Detect a high level of the SCL0 line and ensure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDA0 line low (high level to low level).
(6) Ensure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCL0 line low (high level to low level).
(8) Detect a low level of the SCL0 line and ensure the low-level period of SCL0 line set in the ICBRL register.

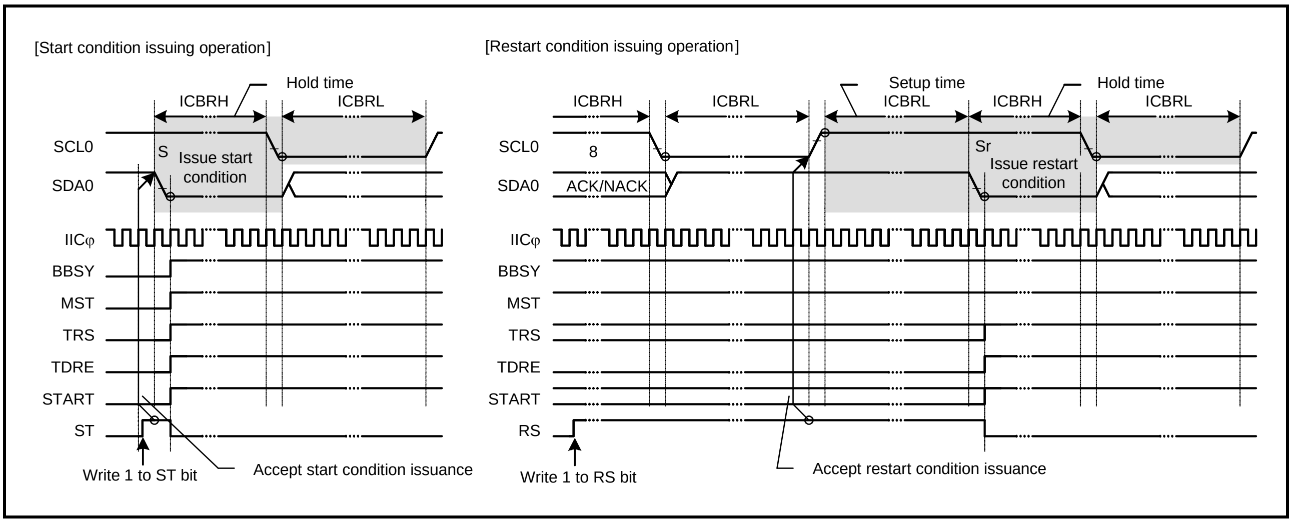


Figure 26.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

26.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the ICCR2.SP bit 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 ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA0 line low (high level to low level).
- Ensure the low-level period of SCL0 line set in the ICBRL register.
- Release the SCL0 line (low level to high level).
- Detect a high level of the SCL0 line and ensure the time set in the ICBRH register and the stop condition setup time.
- Release the SDA0 line (low level to high level).
- Ensure the time set in the ICBRL register and the bus free time.
- Set the BBSY flag to 0 (to release the bus mastership).

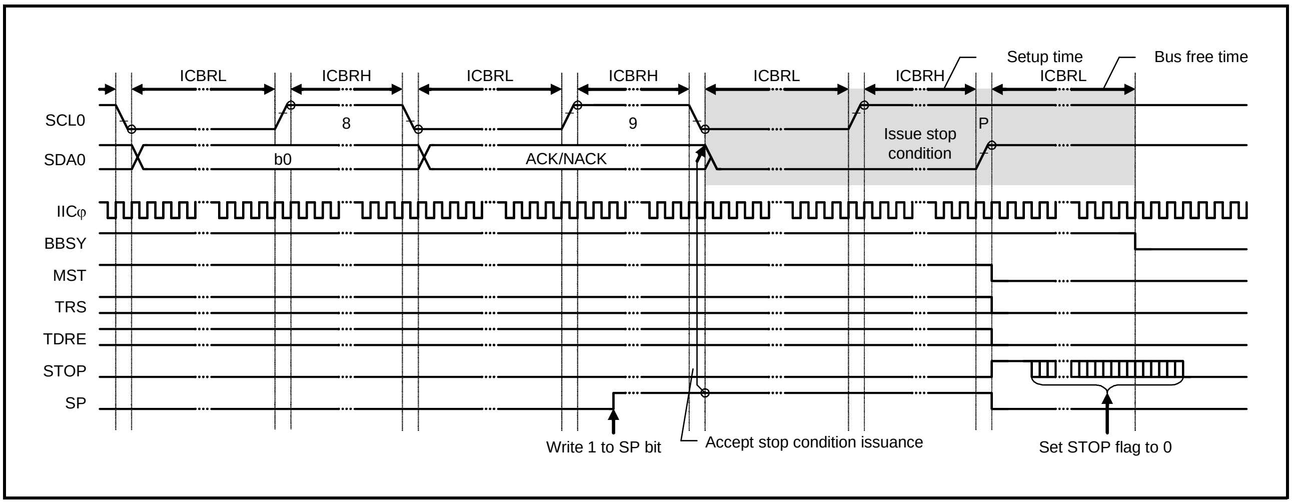


Figure 26.38 Stop Condition Issue Timing (SP Bit)

26.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I^{2}C-bus might hang with a fixed level on the SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL0 or SDA0 lines.

26.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL0 line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC’s own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCL0 line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

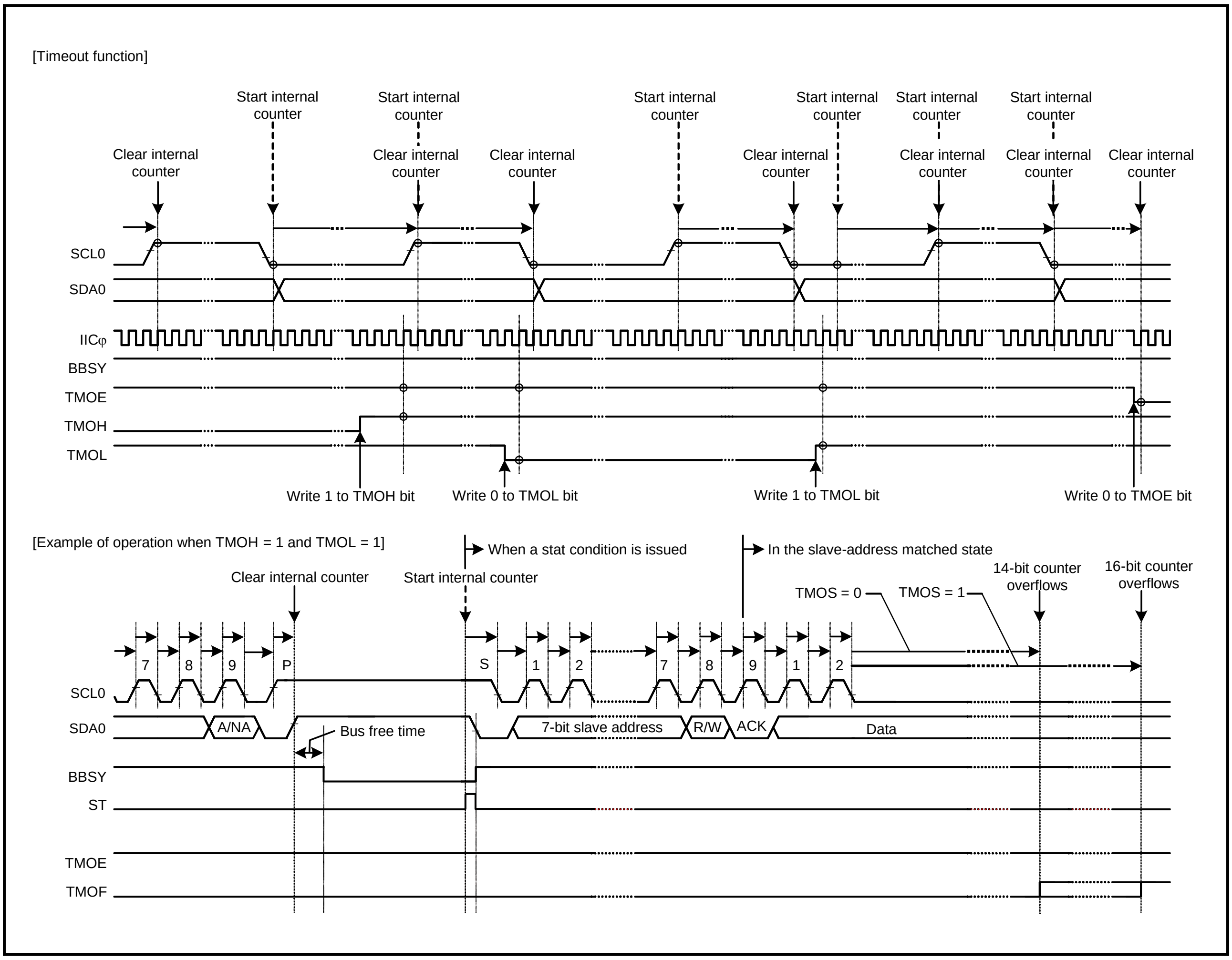


Figure 26.39 Timeout Function

26.11.2 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 SDA0 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 SDA0 line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL clock as the unit if the RIIC cannot issue a stop condition because the slave device is holding the SDA0 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 ICCR1.CLO bit 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 ICMR1.CKS[2:0] bits, and of registers ICBRH and ICBRL) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. Therefore, further extra clock cycles can be output consecutively by writing 1 to the CLO bit after confirming the CLO bit to be 0.
When the RIIC module is in master mode and the slave device is holding the SDA0 line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDA0 line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDA0 line by the slave device, complete communications by reissuing the stop condition.
Use this facility with the ICFER.MALE bit (master arbitration-lost detection disabled) set to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the ICCR1.SDAO bit does not match the state of the SDA0 line, so take care on this point.

[Output conditions for using the ICCR1.CLO bit]

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCL0 line low

Figure 26.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

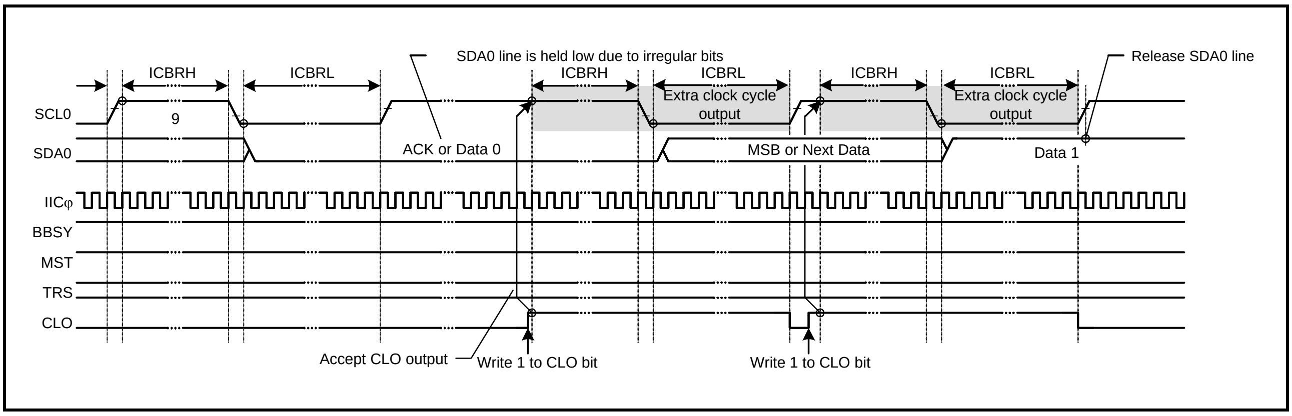


Figure 26.40 Extra SCL Clock Cycle Output Function (CLO Bit)

26.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After issuing a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL0 and SDA0 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 (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 26.14, Resets and Register and Function States When Issuing Each Condition.

26.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

26.12.1 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 specification.
If the time measured with the MTU or TMR exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCL0 pin and SDA0 pin and make the SCL0/SDA0 pin outputs high-impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL 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 specification or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

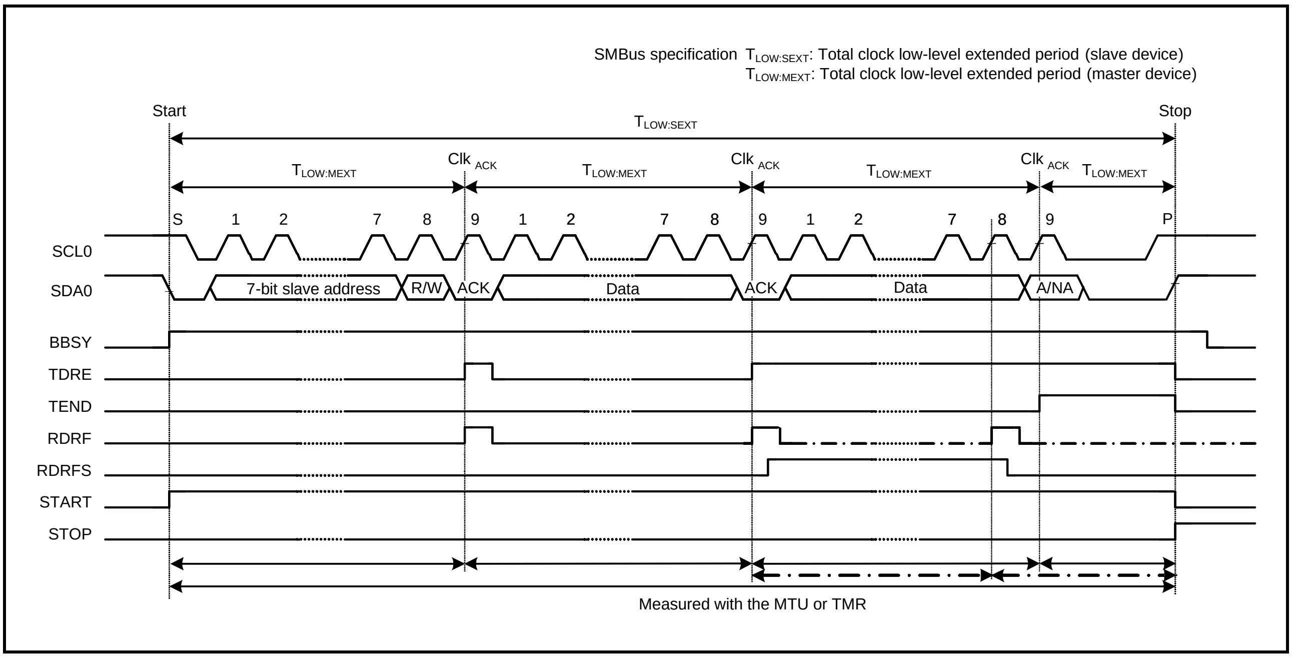


Figure 26.41 SMBus Timeout Measurement

26.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 28, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock cycle.

26.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

26.13 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 26.6 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of activating data transfer by the DTC.

Table 26.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Activation	Priority	Interrupt Condition
EEI	Transfer error/ event generation	AL	Not possible	High	AL = 1 • ALIE = 1
		NACKF		↑	NACKF = 1 • NAKIE = 1
		TMOF			TMOF = 1 • TMOIE = 1
		START			START = 1 • STIE = 1
		STOP			STOP = 1 • SPIE = 1
RXI*2	Receive data full	RDRF	Possible		RDRF = 1 • RIE = 1
TXI*1	Transmit data empty	TDRE	Possible		TDRE = 1 • TIE = 1
TEI*3	Transmit end	TEND	Not possible	Low	TEND = 1 • TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Since TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Since RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

26.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source). An interrupt request that was being retained internally is output to the ICU 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 corresponding interrupt enable bit in the ICIER register.

26.14 Resets and Register and Function States When Issuing Each Condition

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 26.7 lists the register and function states when issuing each reset or condition.

Table 26.7 Register and Function States When Issuing Each Reset or Condition

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	ICE, IICRST	At a reset	Retained	Retained	Retained	Retained
	SCLO, SDAO		At a reset	At a reset		
	Others			Retained		
ICCR2	BBSY	At a reset	At a reset	Retained	Retained	Retained
	ST			At a reset	At a reset	Retained
	Others					At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Retained
	Others			Retained	Retained	
ICMR2		At a reset	At a reset	Retained	Retained	Retained
ICMR3		At a reset	At a reset	Retained	Retained	Retained
ICFER		At a reset	At a reset	Retained	Retained	Retained
ICSER		At a reset	At a reset	Retained	Retained	Retained
ICIER		At a reset	At a reset	Retained	Retained	Retained
ICSR1		At a reset	At a reset	At a reset	Retained	At a reset
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Retained	At a reset
	START				Retained	
	STOP				Retained	Retained
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		At a reset	At a reset	Retained	Retained	Retained
ICBRH, ICBRL		At a reset	At a reset	Retained	Retained	Retained
ICDRT		At a reset	At a reset	Retained	Retained	Retained
ICDRR		At a reset	At a reset	Retained	Retained	Retained
ICDRS		At a reset	At a reset	At a reset	Retained	Retained
Timeout function		At a reset	At a reset	Operation	Operation	Operation
Bus free time measurement		At a reset	At a reset	Operation	Operation	Operation

26.15 Usage Notes

26.15.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control register B, refer to section 11, Low Power Consumption.

26.15.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

27. Serial Peripheral Interface (RSPIa)

In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

This MCU includes one channel 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 27.1 lists the specifications of the RSPI, and Figure 27.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

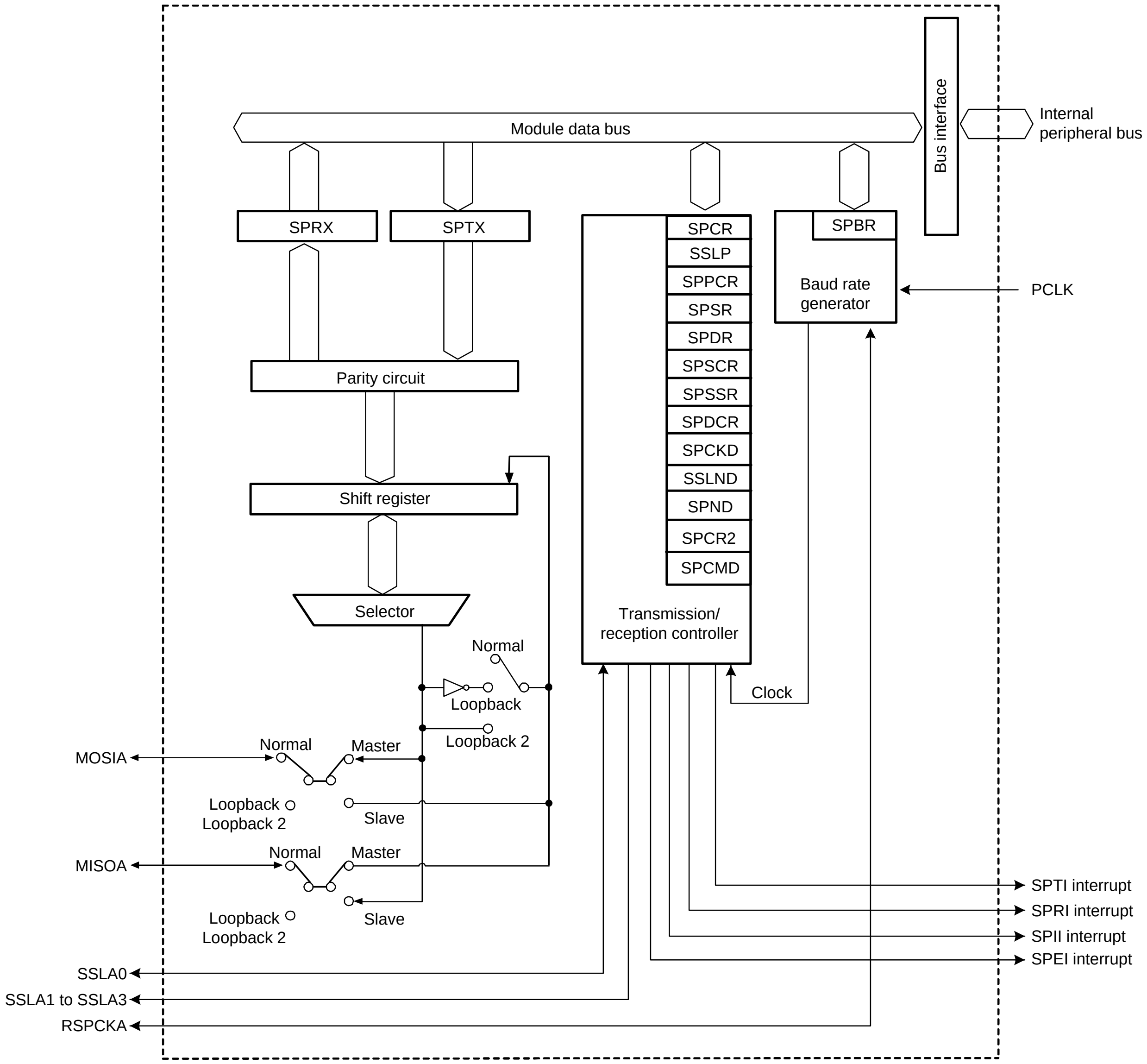
Table 27.1 RSPI Specifications (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Transmit-only operation is available. • Communication mode: Full-duplex or transmit-only can be selected. • Switching of the polarity of RSPCK • Switching of the phase of RSPCK
Data format	• MSB first/LSB first selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. • 128-bit transmit/receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096). • In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8). Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection*1 • Parity error detection
SSL control function	• Four SSL pins (SSLA0 to SSLA3) for each channel • In single-master mode, SSLA0 to SSLA3 pins are output. • In multi-master mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused. • In slave mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function

Table 27.1 RSPI Specifications (2/2)

Item	Description
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt RSPI error interrupt (mode fault, overrun, or parity error) RSPI idle interrupt (RSPI idle)
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.



SPCR:	RSPI control register	SSLND:	RSPI slave select negation delay register
SPCR2:	RSPI control register 2	SPND:	RSPI next-access delay register
SSLP:	RSPI slave select polarity register	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SPPCR:	RSPI pin control register	SPBR:	RSPI bit rate register
SPSR:	RSPI status register	SPTX:	RSPI transmit buffer
SPDR:	RSPI data register	SPRX:	RSPI receive buffer
SPSCR:	RSPI sequence control register	SPTI:	RSPI transmit interrupt
SPSSR:	RSPI sequence status register	SPRI:	RSPI receive interrupt
SPDCR:	RSPI data control register	SPII:	RSPI idle interrupt
SPCKD:	RSPI clock delay register	SPEI:	RSPI error interrupt

Figure 27.1 RSPI Block Diagram

Table 27.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 27.3.2, Controlling RSPI Pins for details.

Table 27.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output

27.2 Register Descriptions

27.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODF EN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI error interrupt requests	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	RSPI Receive Buffer Full Interrupt Enable	0: Disables the generation of RSPI receive buffer full interrupt requests 1: Enables the generation of RSPI receive buffer full interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Operation should not be performed if the CPHA bit is set to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 27.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 27.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 27.3.2, 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 (refer to section 27.3.8, Error Detection).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.
A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.
Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.
When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 27.3.8, 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 27.3.9, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

SPRIE Bit (RSPI Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive buffer full interrupt request.

27.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 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 should not be performed.

27.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 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 (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.
When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.
When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

27.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	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	SPTEF	Transmit Buffer Empty Flag	0: Transmit buffer has valid data 1: Transmit buffer has no valid data	R/(W) *2
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R/(W) *2

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. The write value should be 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR1.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 27.3.8.1, Overrun Error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (disables the RSPI function)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)

4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (disables the RSPI function)

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 (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then 0 is written to the MODF flag

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When data is transferred from the transmit buffer to the shift register

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register

27.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h

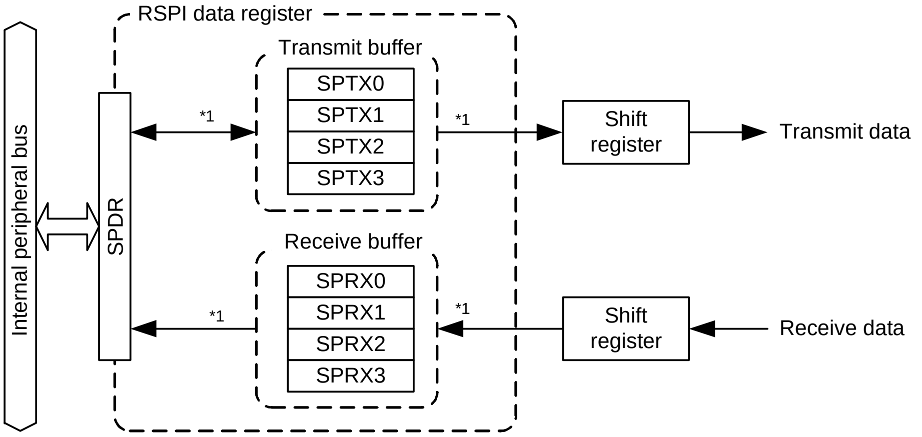
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Address(es): RSPI0.SPDR.H 0008 8384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1), access SPDR.
When accessing in words (the SPLW bit is 0), access the higher-order 16 bits (H) of SPDR.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 27.2 shows the Configuration of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 27.2 Configuration of SPDR

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of SPDR.
Data written to SPDR are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer.
The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus Interface

SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the RSPI longword access/word access specification bit in the RSPI data control register (SPDCR.SPLW). Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

(a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 27.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

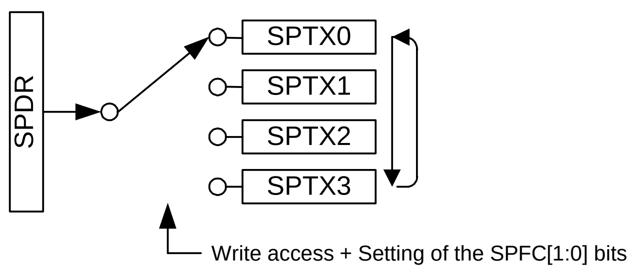


Figure 27.3 Configuration of SPDR (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

(b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.
The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.
Figure 27.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

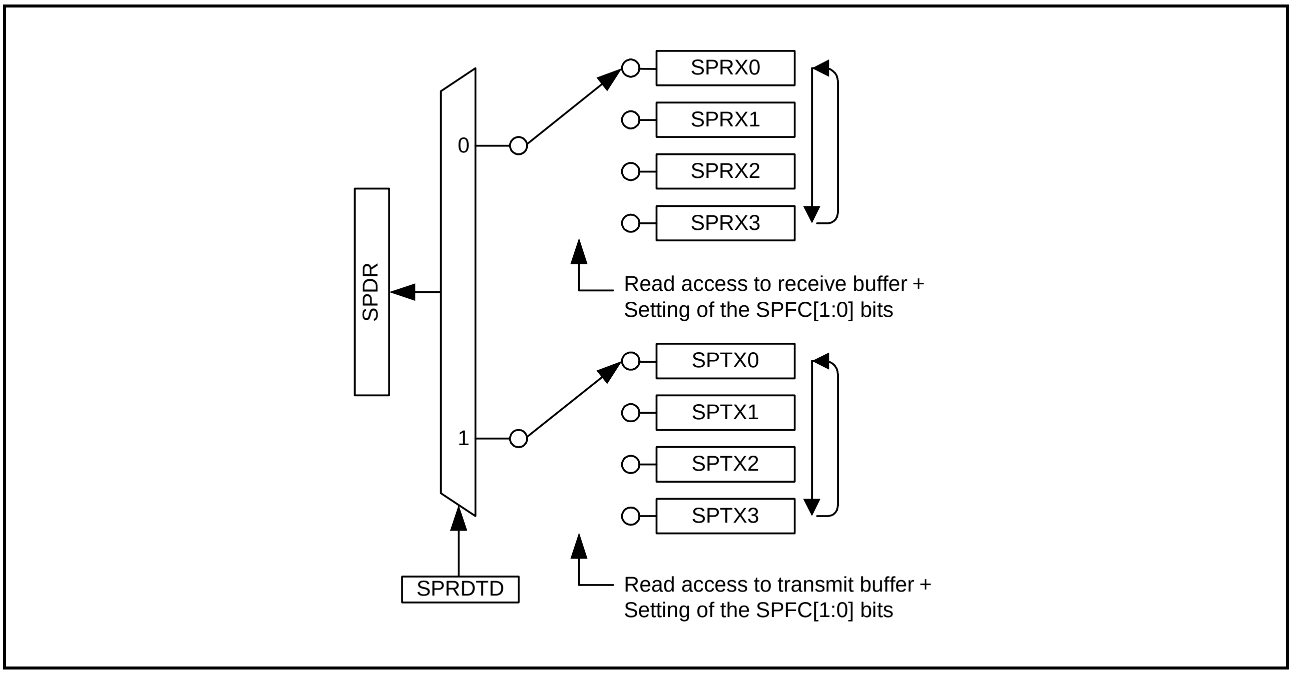


Figure 27.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.
The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.
The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

27.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 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→… The order in which the SPCMD0 to SPCMD07 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.

In slave mode, SPCMD0 is referred.

27.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 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, refer to section 27.3.10.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, refer to section 27.3.8, Error Detection. For the RSPI's sequence control, refer to section 27.3.10.1, Master Mode Operation.

27.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah

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

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n + 1) \times 2^N}$$

Table 27.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 27.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate		
			PCLK = 32 MHz	PCLK = 36MHz	PCLK = 40MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps
5	2	48	667 kbps	750 kbps	833 kbps
5	3	96	333 kbps	375 kbps	417 kbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps

27.2.9 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 Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W
b5	SPLW	RSPI Longword Access/ Word Access Specification	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.
When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.
When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the RSPI receive buffer full interrupt is generated.
Table 27.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations should not be performed.

Table 27.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data”
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 27.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR is in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operations should not be performed.

27.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 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 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 should not be performed.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

27.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 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 should not be performed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

27.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 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 should not be performed.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

27.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE, SPOE, or SCKASE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.
The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.
The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.
When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.
The SPOE bit is valid only when the SPPE bit is 1.

SPIIE Bit (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 27.3.8.1, Overrun Error.

27.2.14 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 should not be performed.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (refer to section 27.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 27.3.10.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPLW bit is 0, set the SPB[3:0] bits to “0100b” (8 bits) to “1111b” (16 bits).

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.
When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.
When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.
When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

27.3 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.

27.3.1 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 27.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 27.5 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 change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*2,*3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

27.3.2 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 27.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

Table 27.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi Bit for I/O Ports = 0	ODRn.Bi Bit for I/O Ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	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
	SSLA0 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
	SSLA0 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 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 27.7.

Table 27.7 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	Low
1	1	High

27.3.3 RSPI System Configuration Examples

27.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 27.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

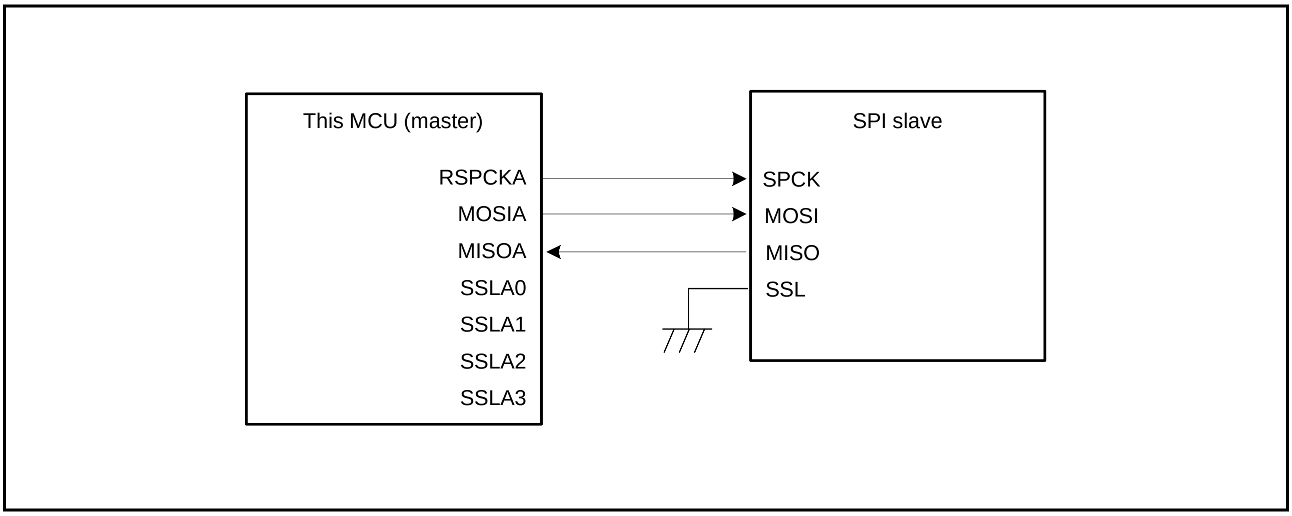


Figure 27.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

27.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 27.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 27.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

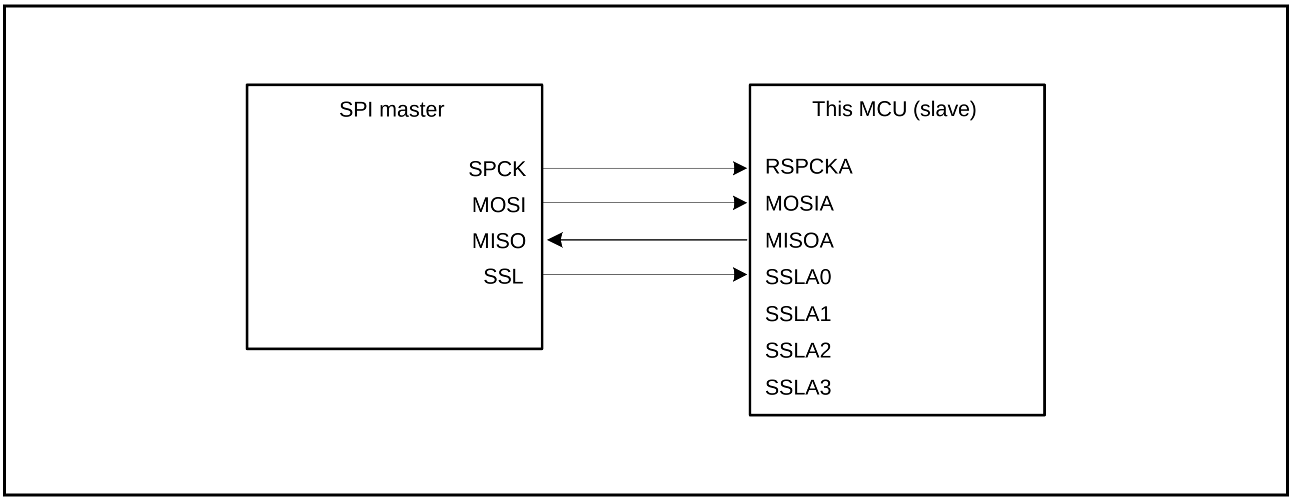


Figure 27.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

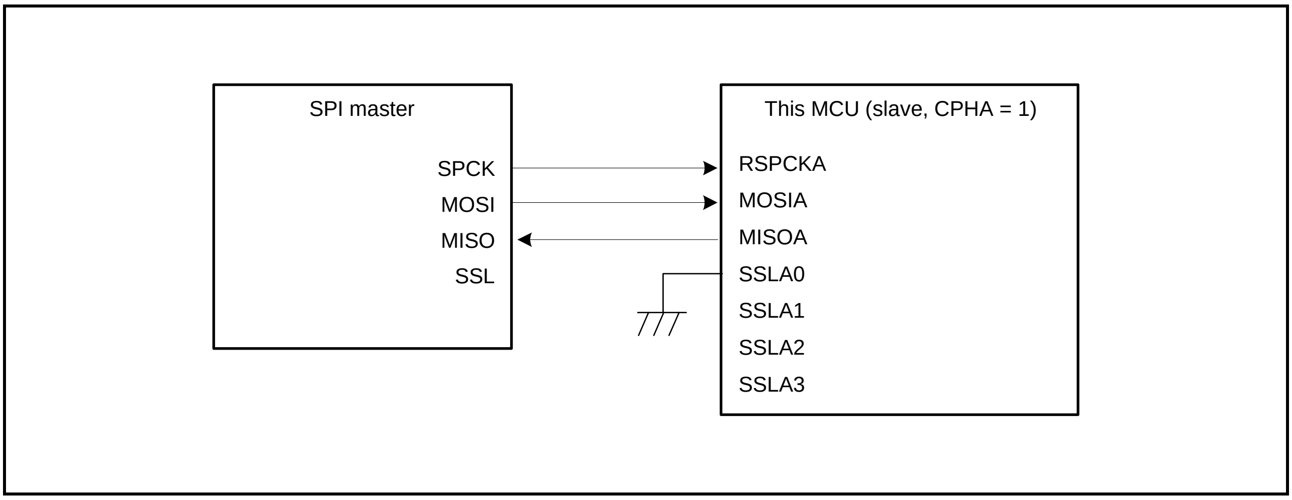


Figure 27.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

27.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 27.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 27.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKA, MOSIA, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

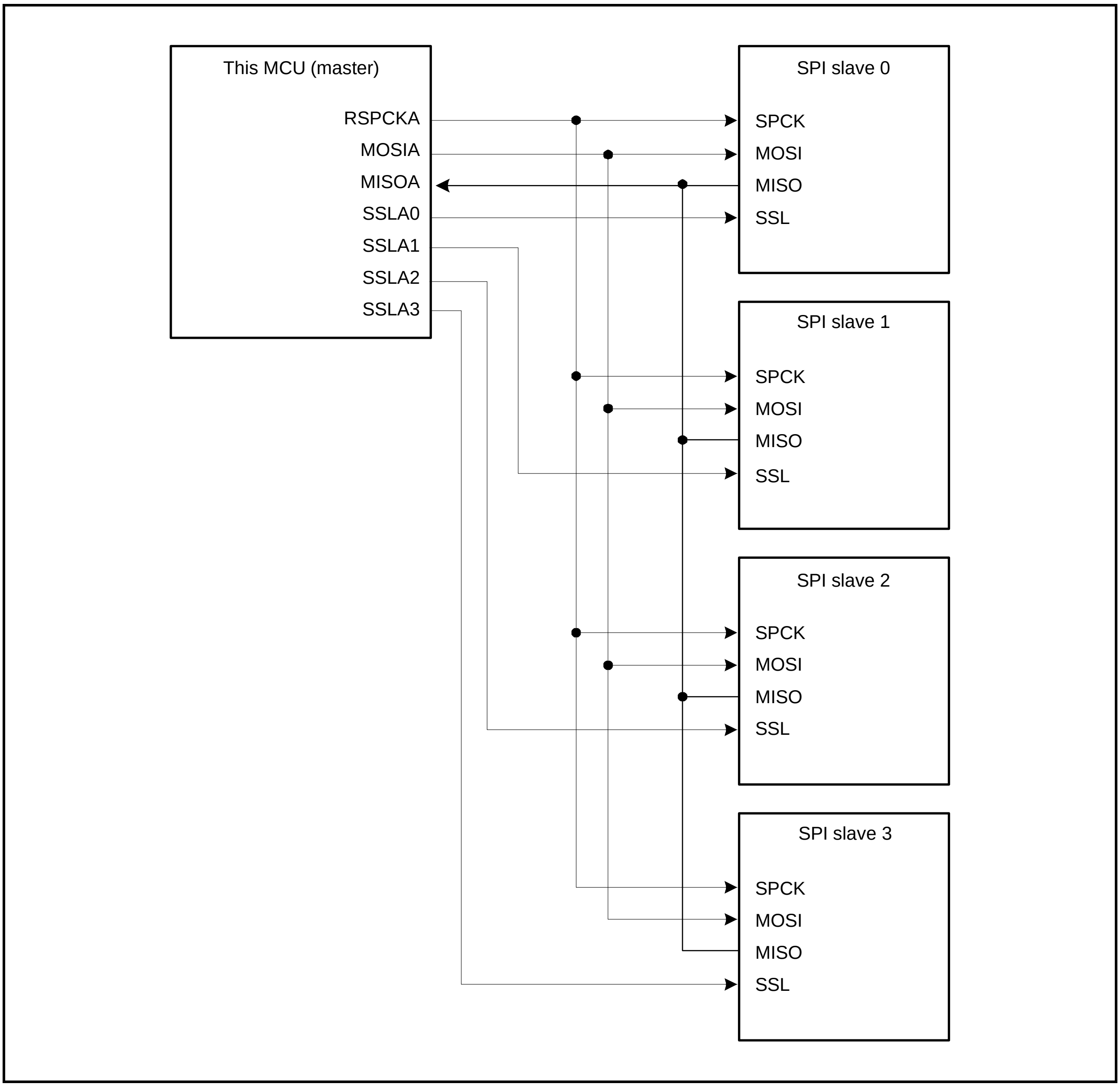


Figure 27.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

27.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 27.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 27.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

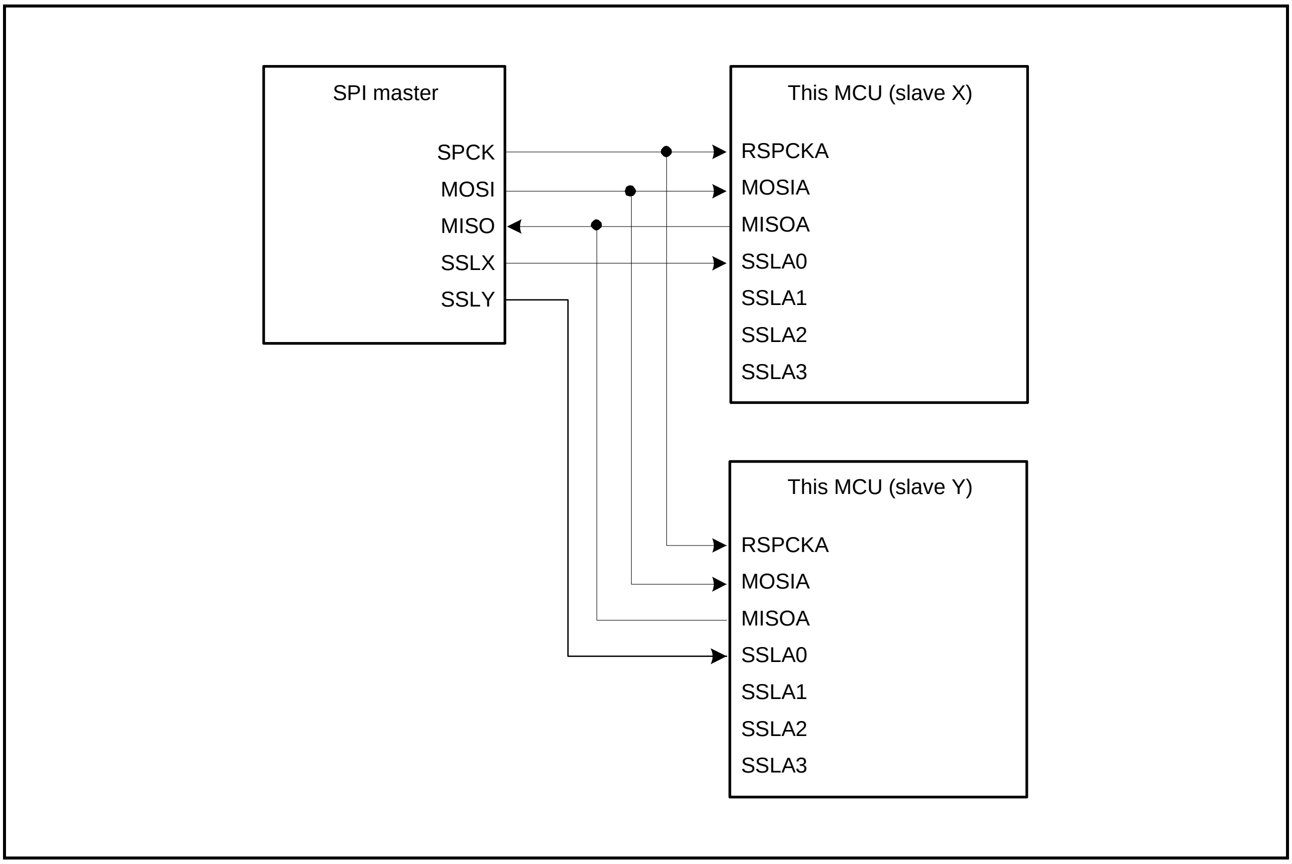


Figure 27.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

27.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 27.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 27.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

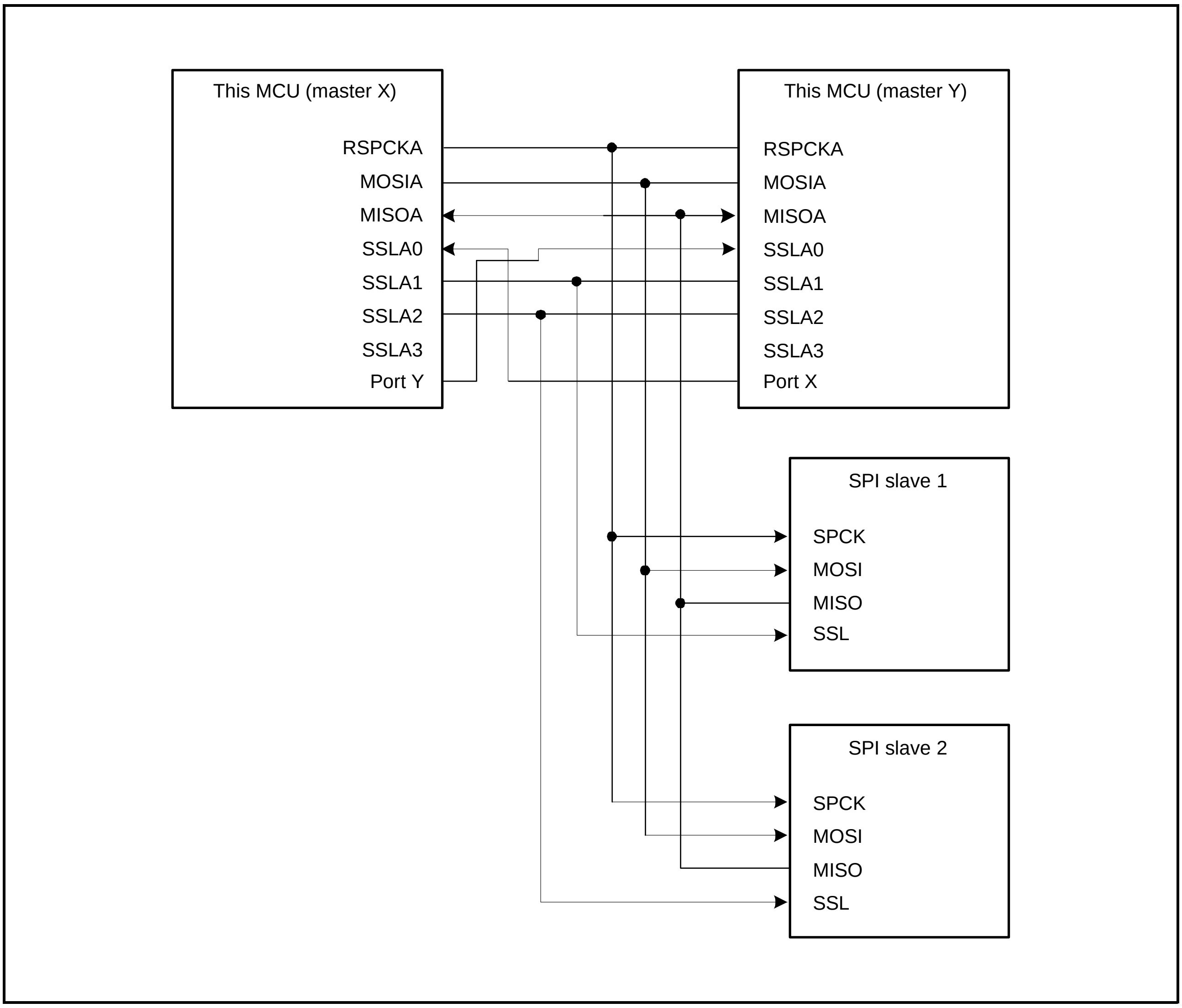


Figure 27.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

27.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 27.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

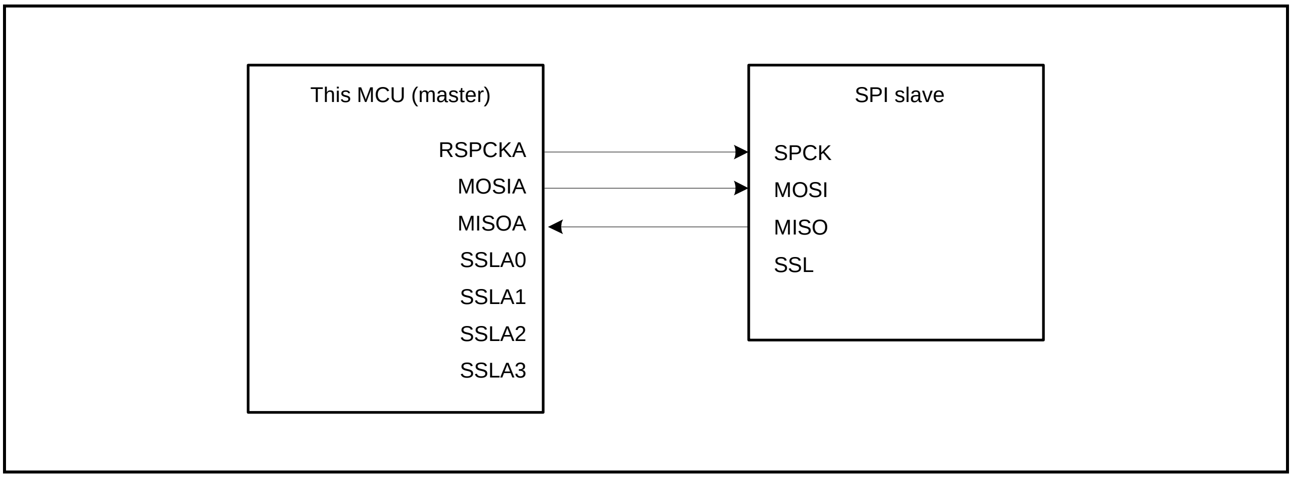


Figure 27.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

27.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 27.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

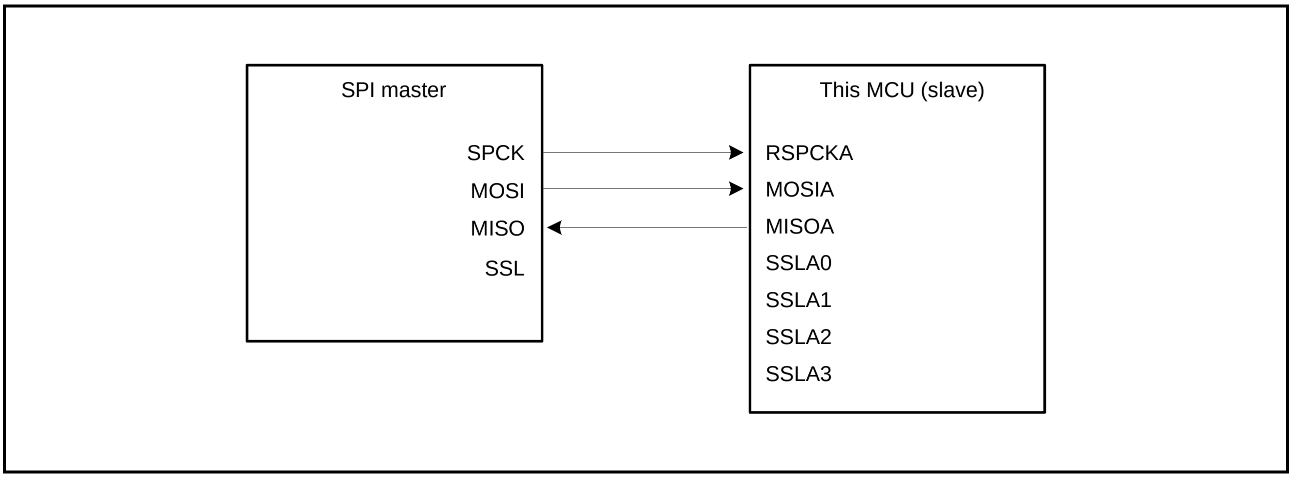


Figure 27.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

27.3.4 Data Format

The RSPI’s data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

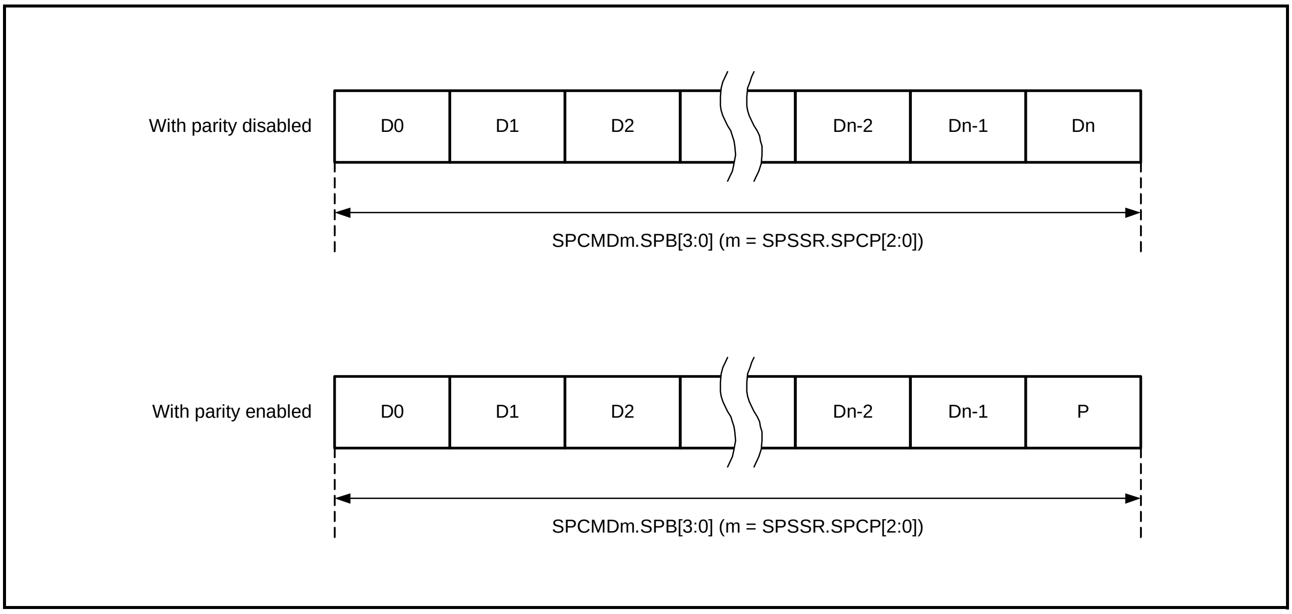


Figure 27.13 Outline of the Data Format (with Parity Disabled/Enabled)

27.3.4.1 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 27.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

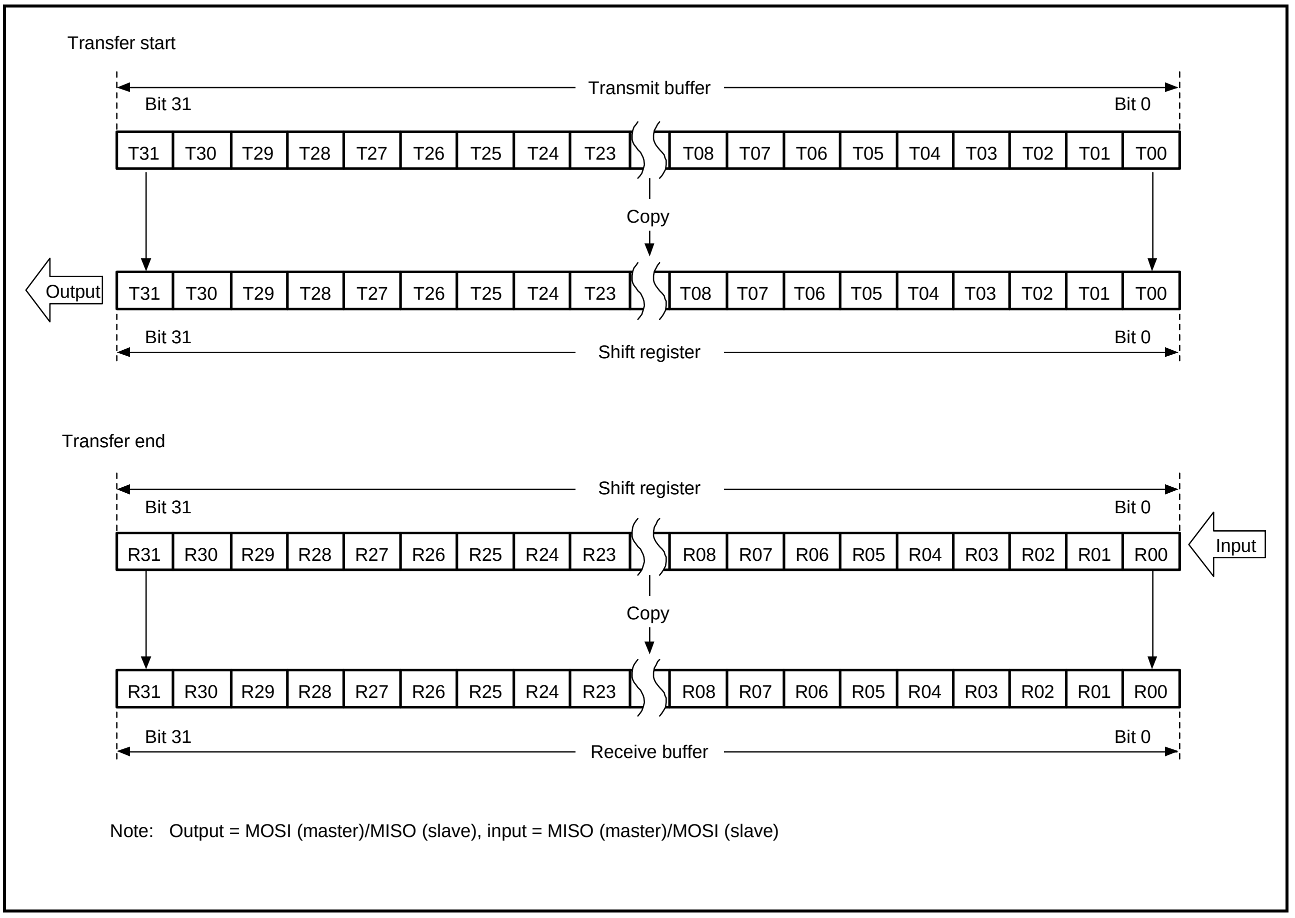
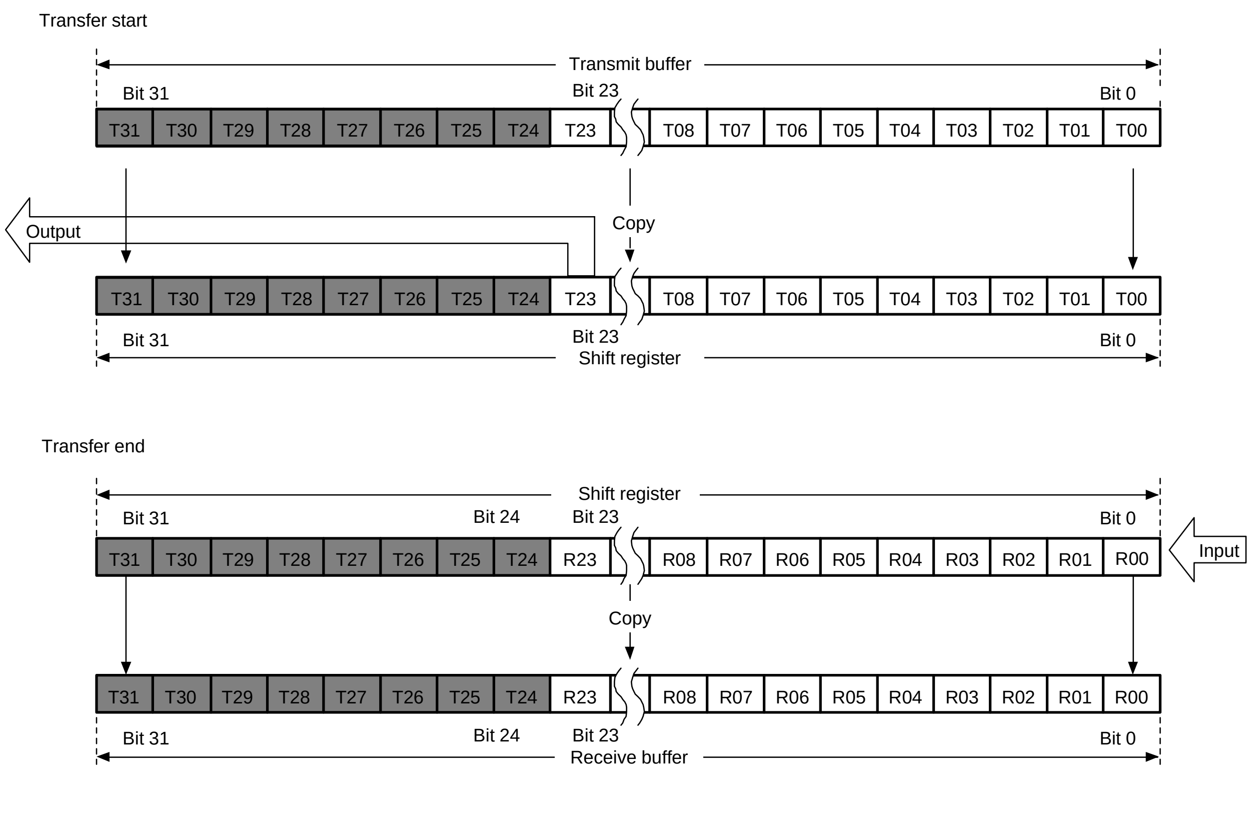


Figure 27.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 27.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 27.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 27.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

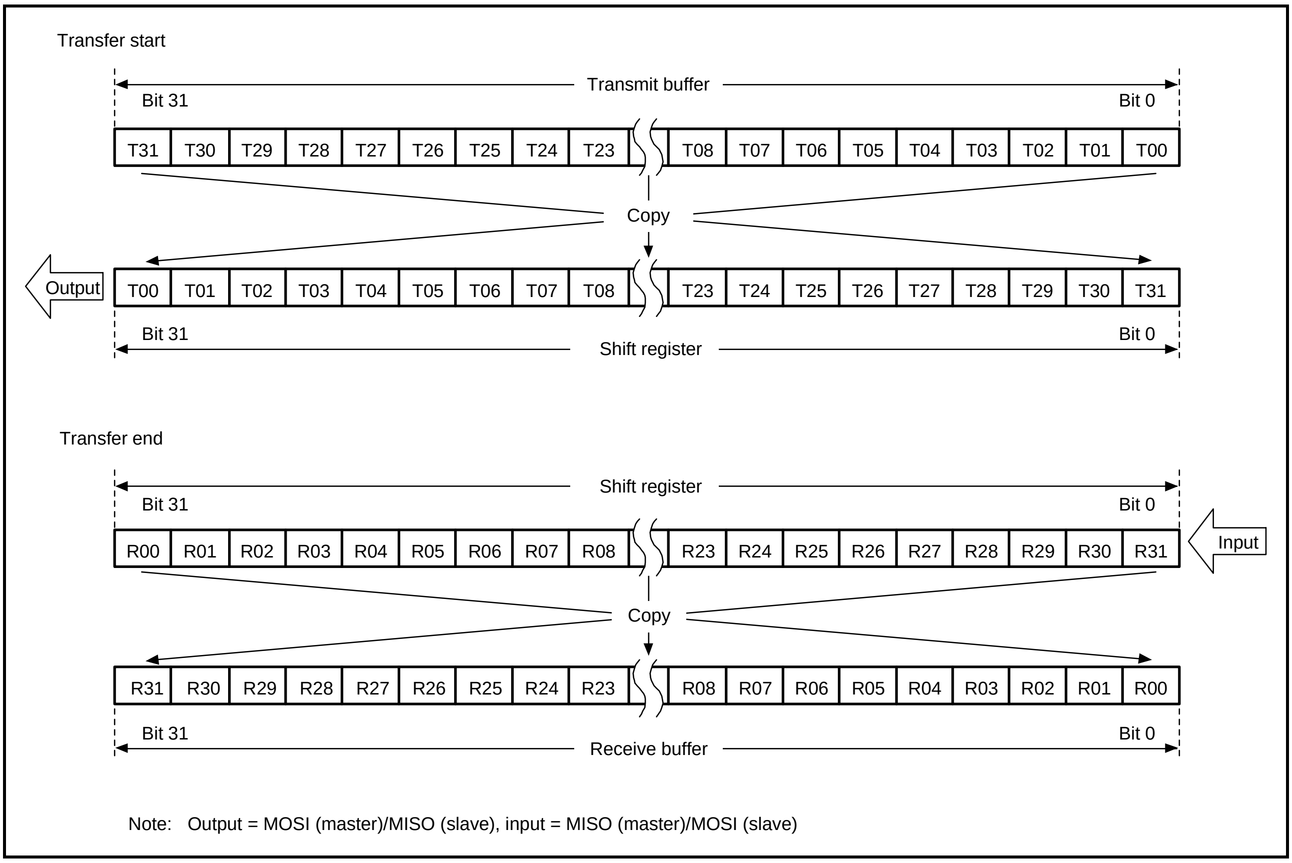


Figure 27.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 27.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

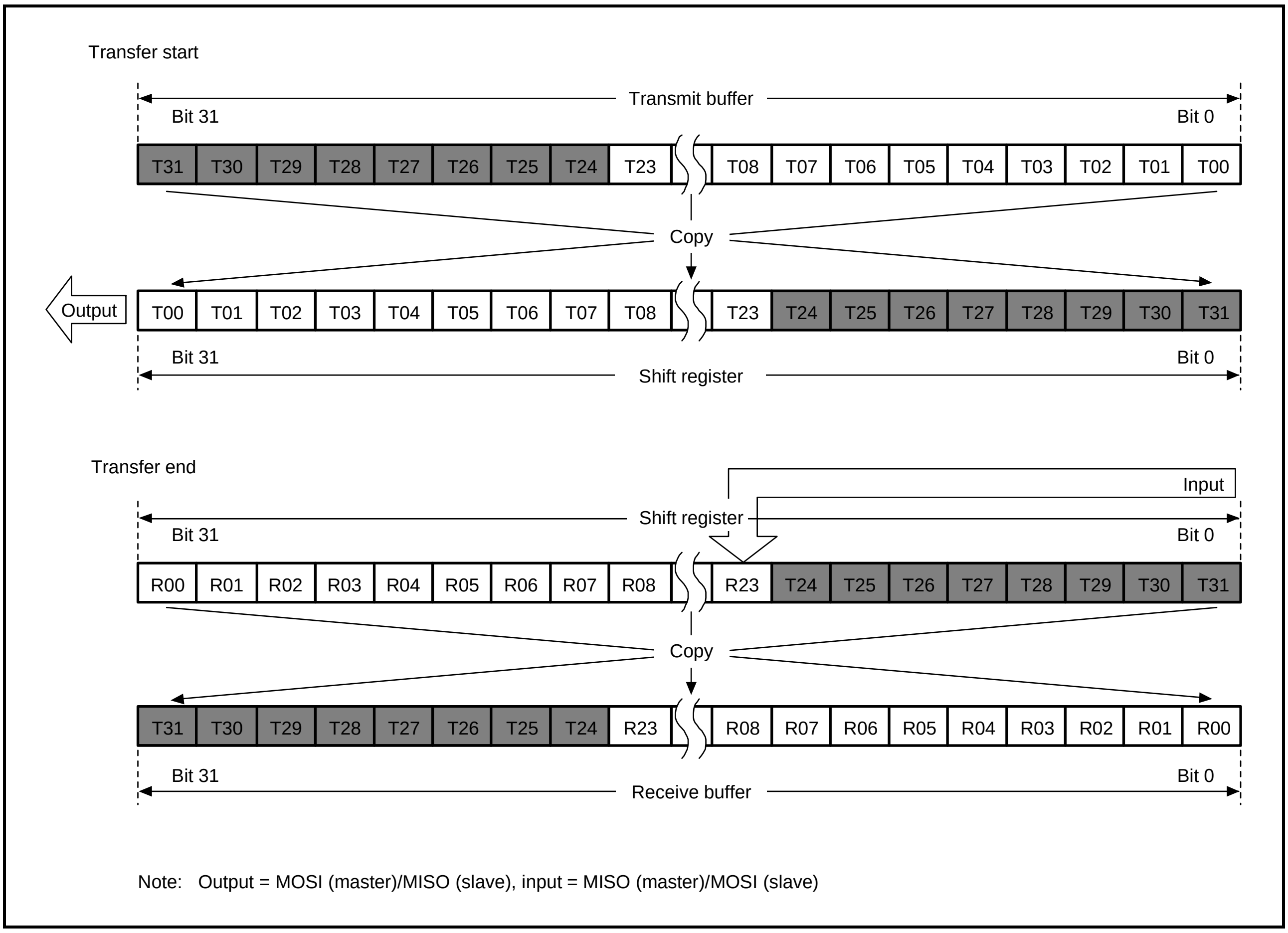


Figure 27.17 LSB First Transfer (24-Bit Data, Parity Disabled)

27.3.4.2 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 27.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

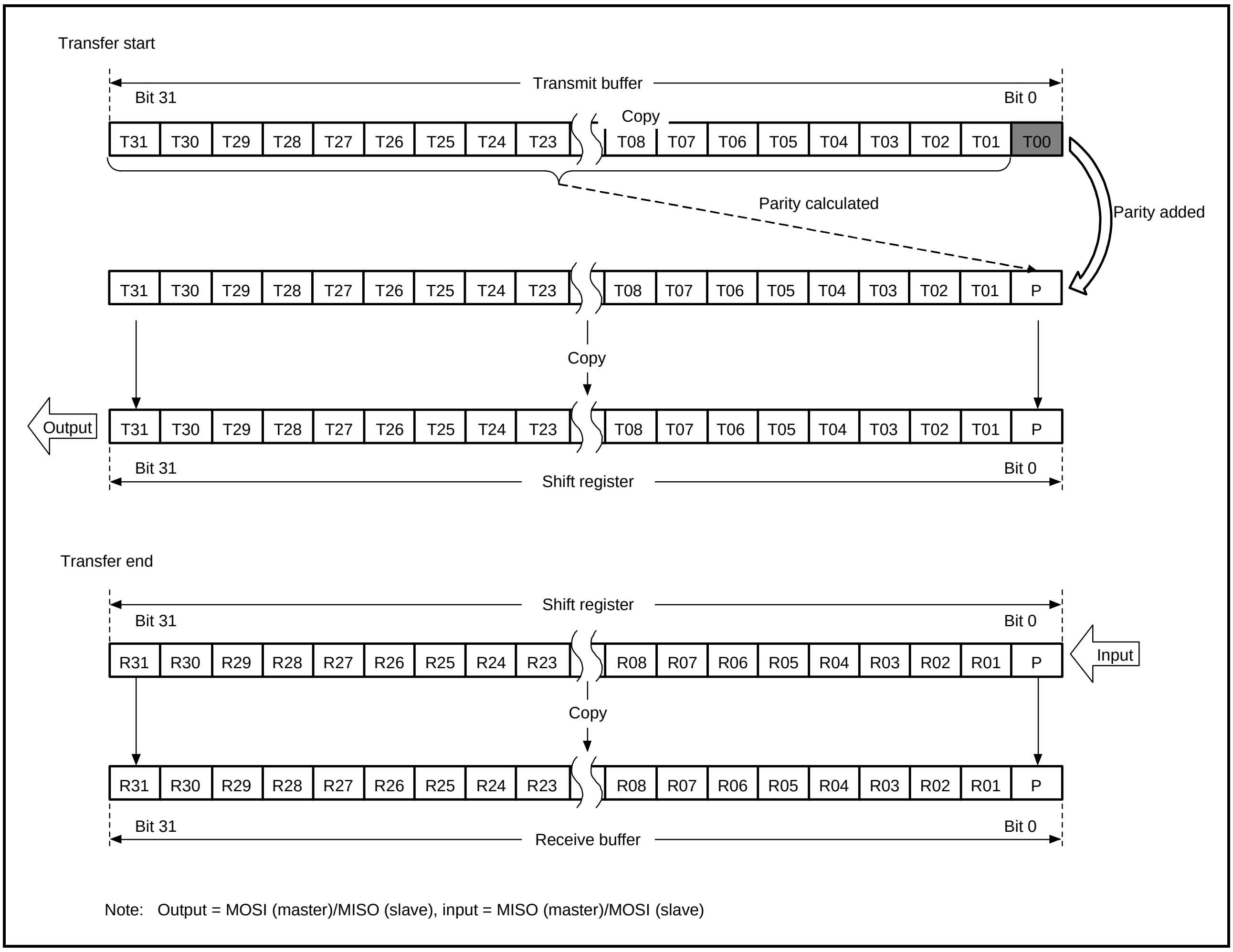


Figure 27.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 27.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

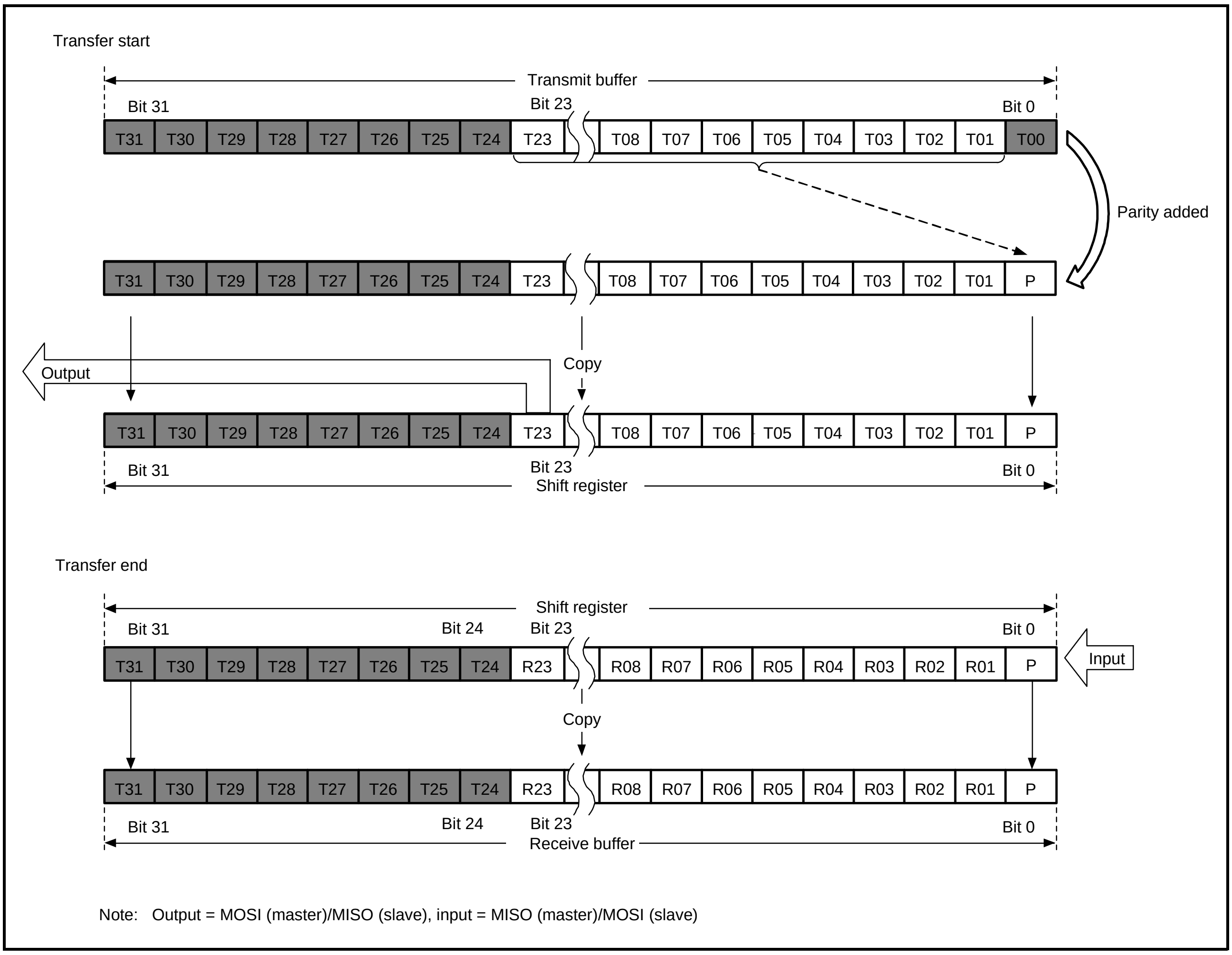


Figure 27.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 27.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

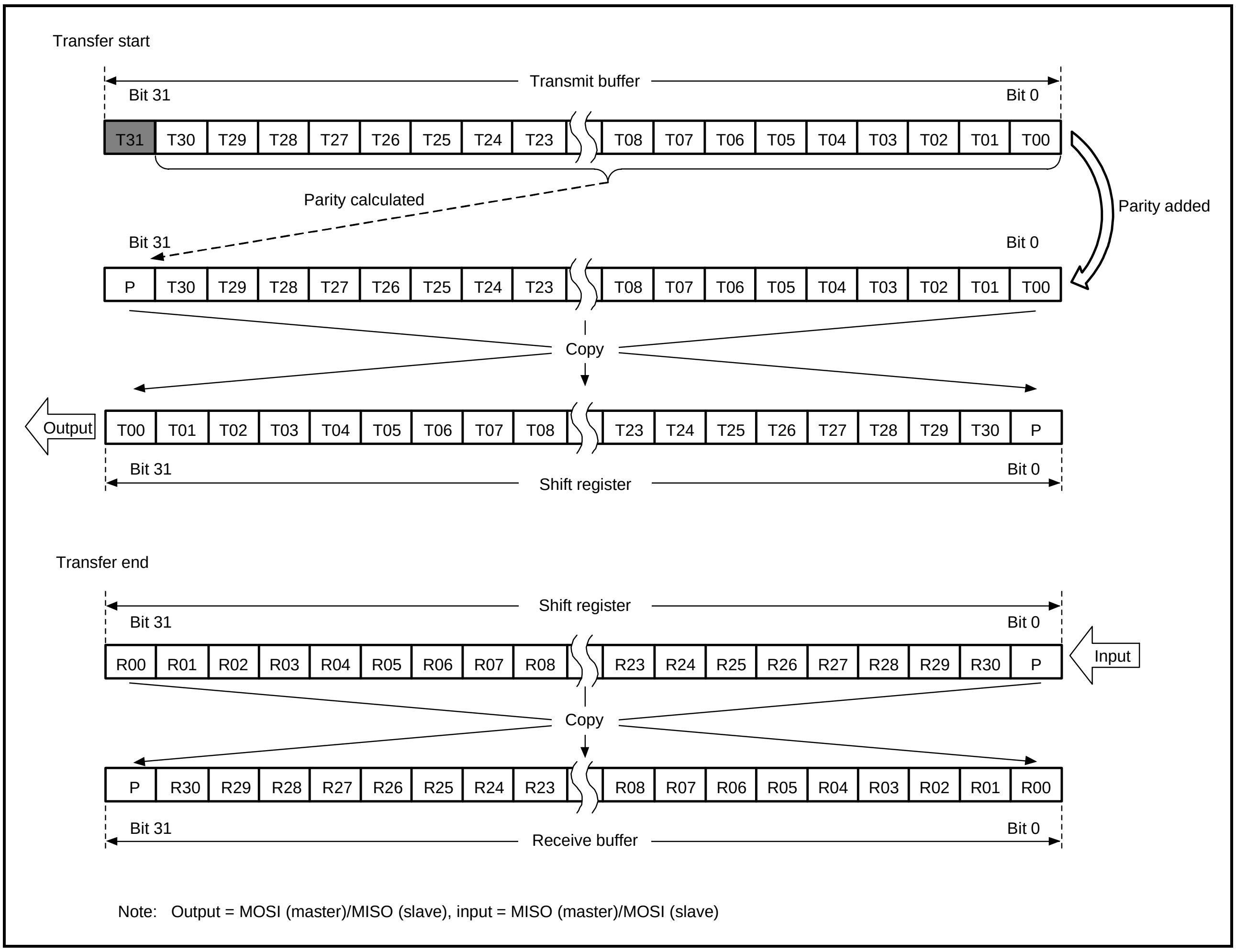


Figure 27.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 27.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

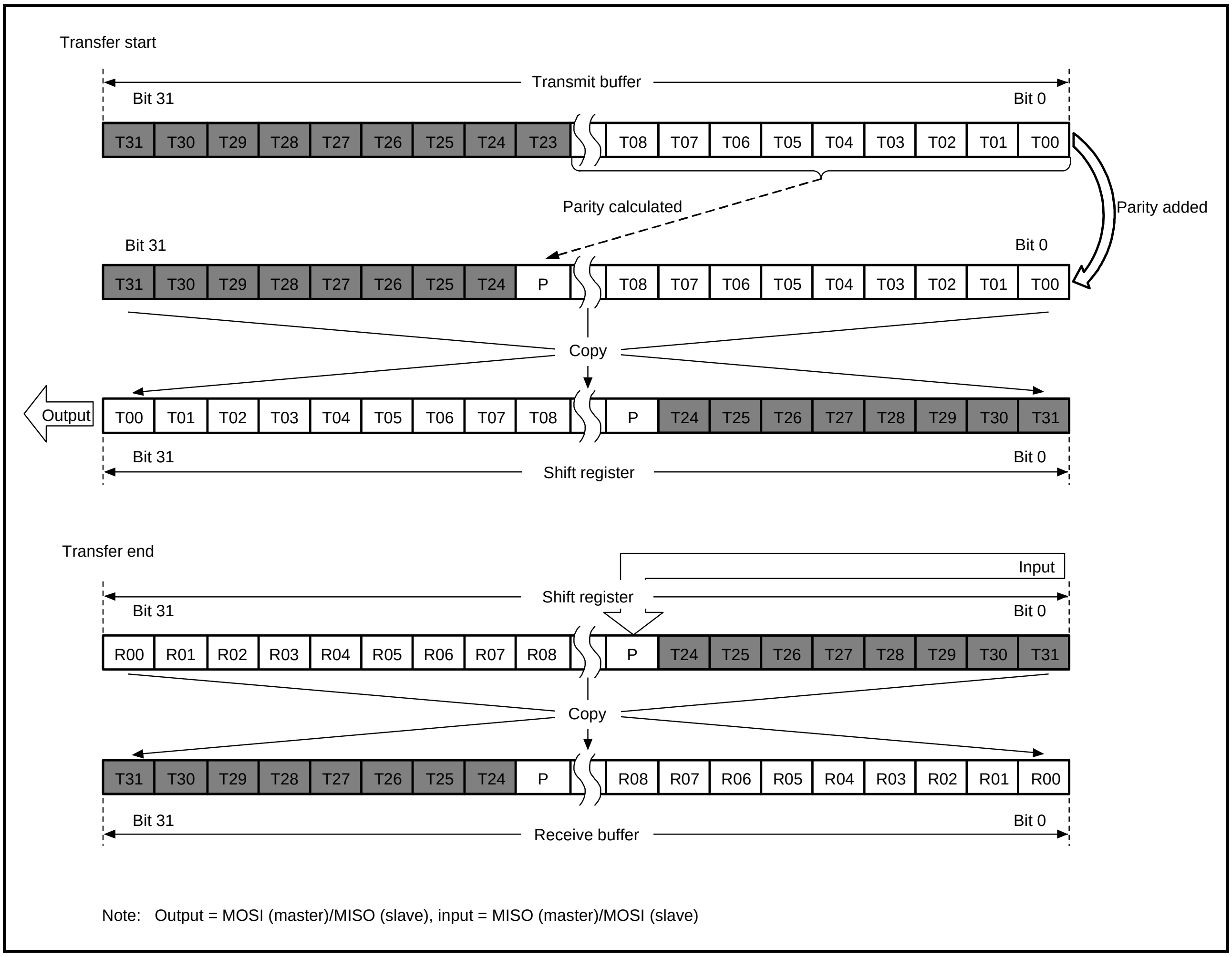


Figure 27.21 LSB First Transfer (24-Bit Data, Parity Enabled)

27.3.5 Transfer Format

27.3.5.1 CPHA = 0

Figure 27.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 27.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 27.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 27.3.10.1, Master Mode Operation.

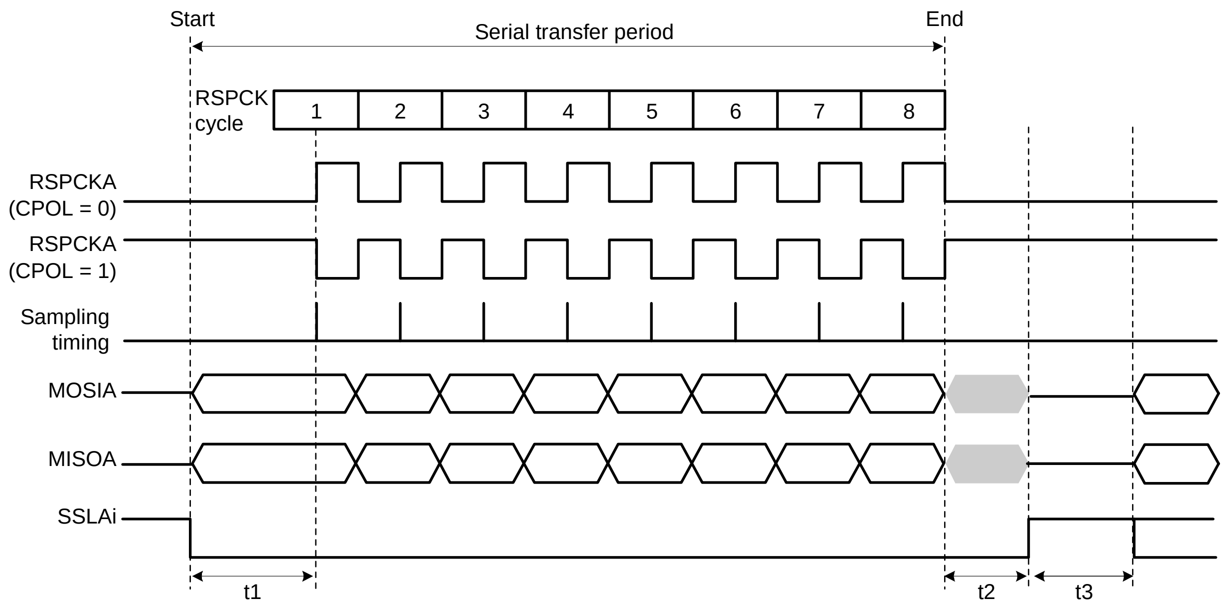


Figure 27.22 RSPI Transfer Format (CPHA = 0)

27.3.5.2 CPHA = 1

Figure 27.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 27.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 27.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 27.3.10.1, Master Mode Operation.

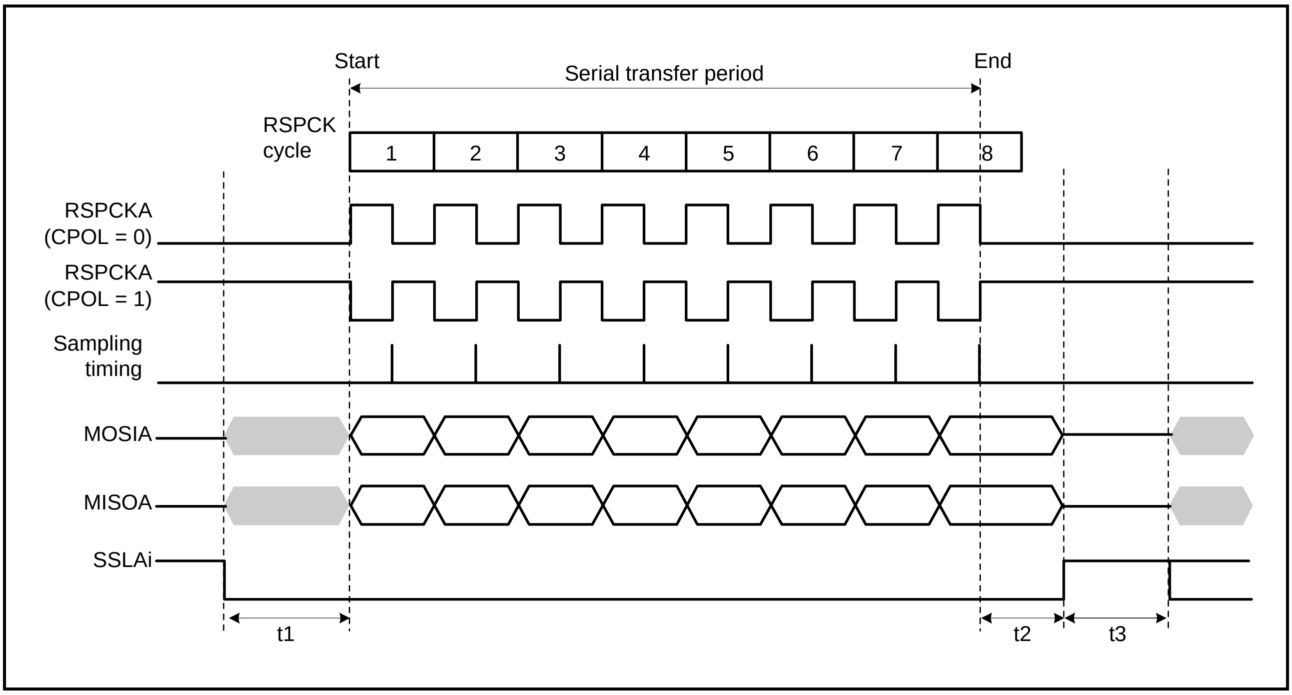


Figure 27.23 RSPI Transfer Format (CPHA = 1)

27.3.6 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 27.24 and Figure 27.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

27.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 27.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 27.24, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

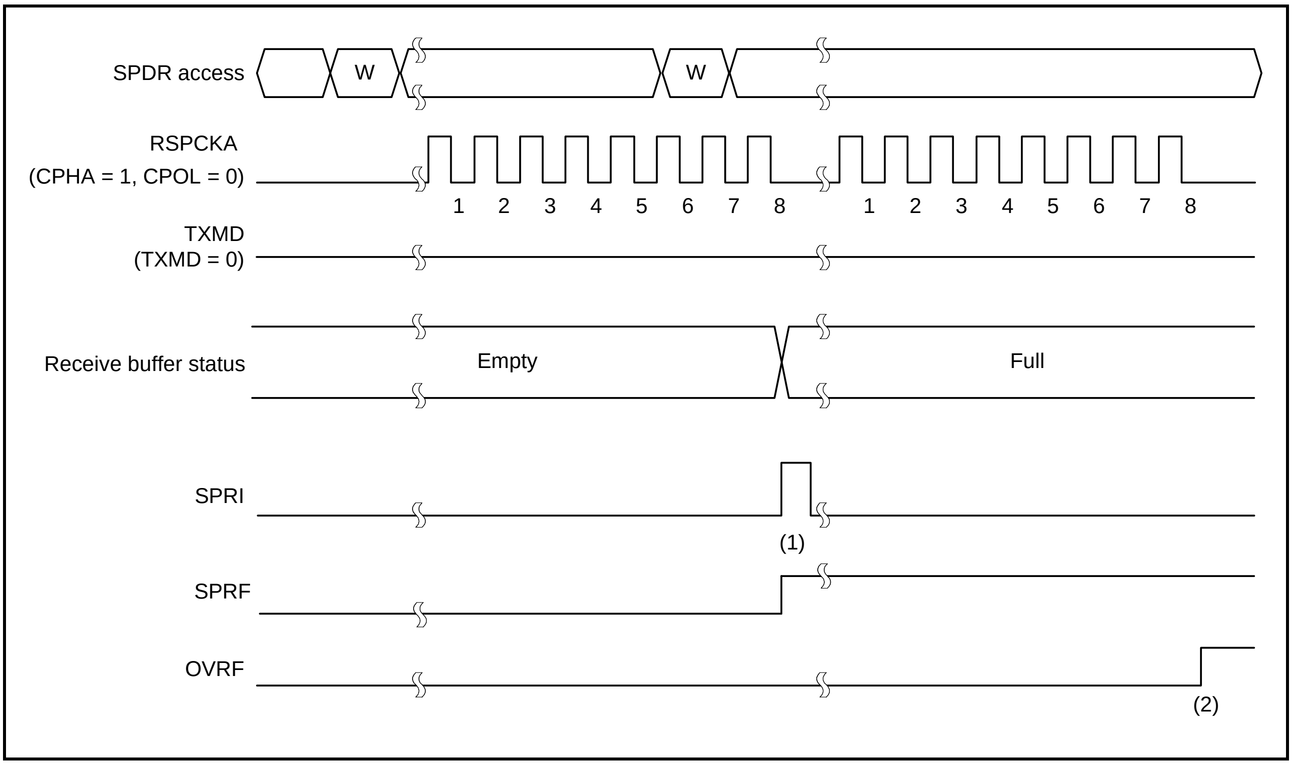


Figure 27.24 Operation Example of SPCR.TXMD = 0

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of 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.

When full-duplex synchronous serial communications (SPCR.TXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

27.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 27.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 27.25, 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).

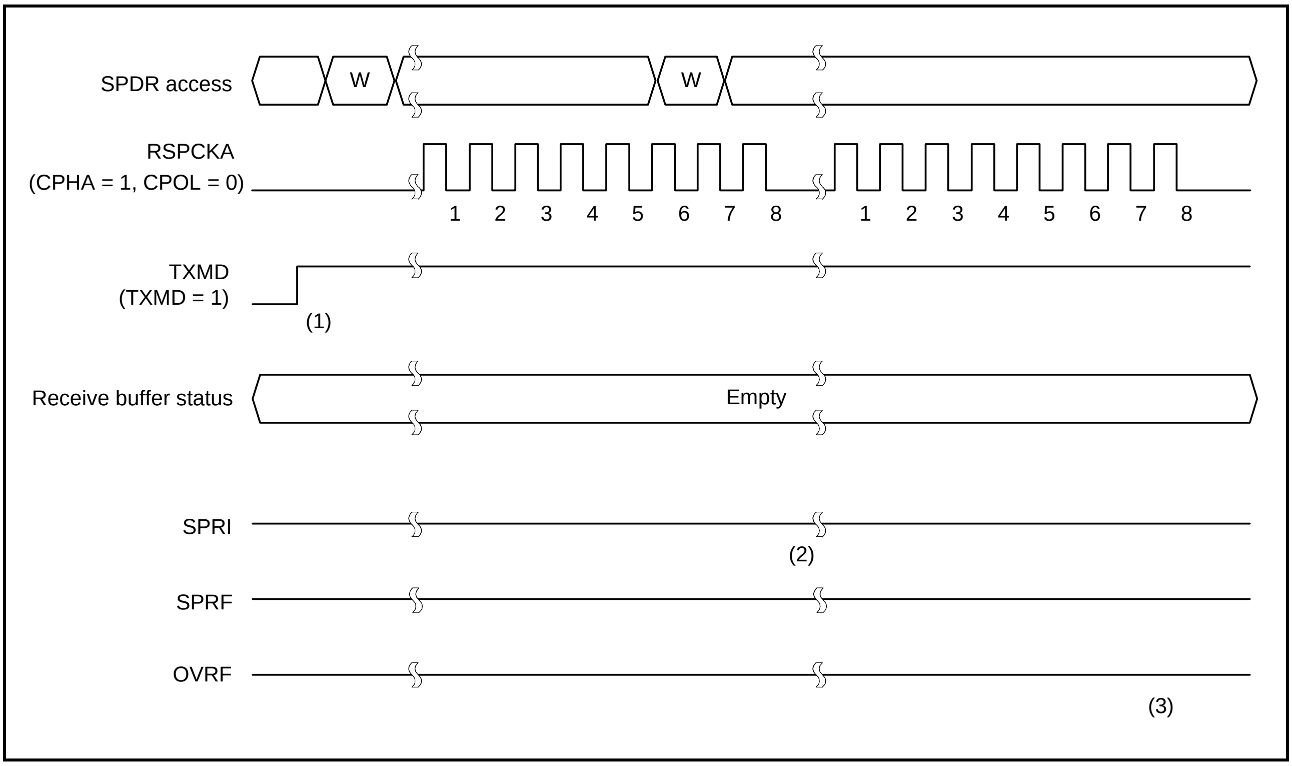


Figure 27.25 Operation Example of SPCR.TXMD = 1

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 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 SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of 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 data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

27.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 27.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 27.26 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 27.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

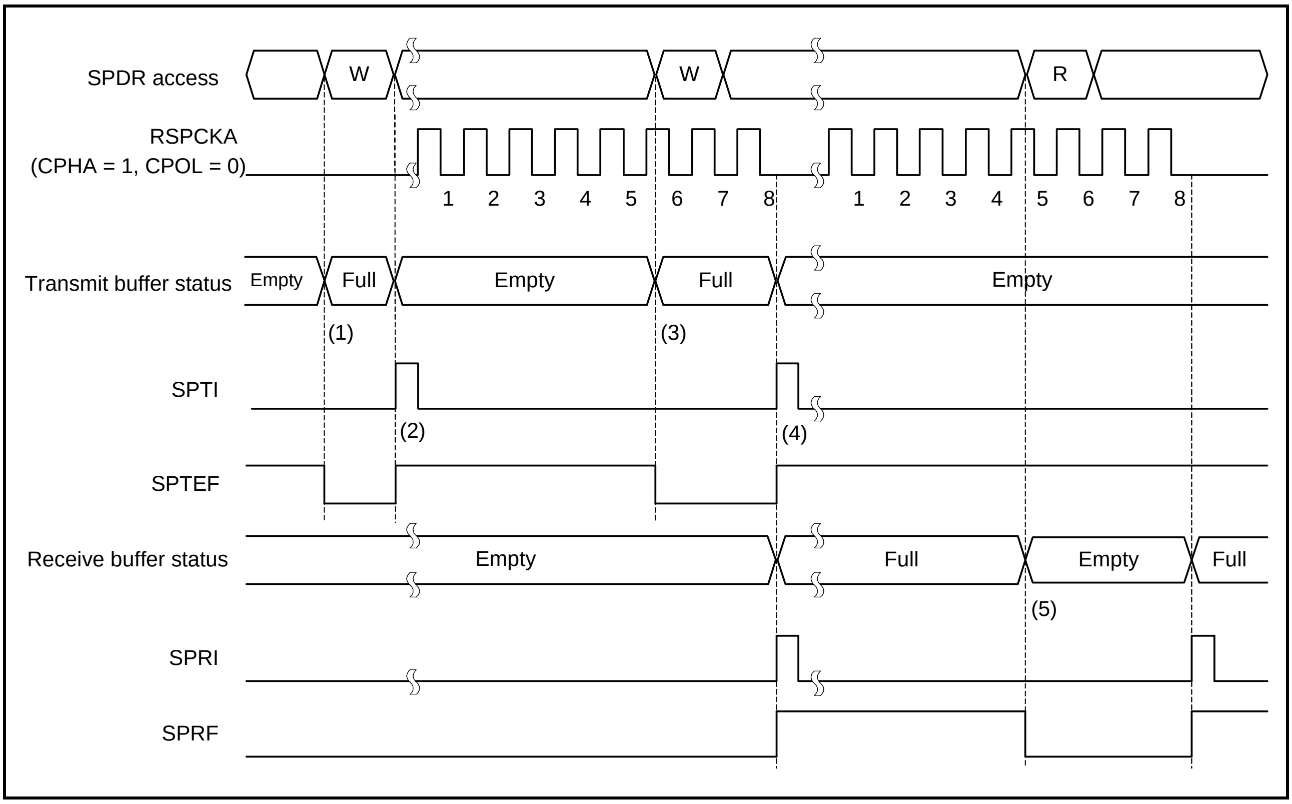


Figure 27.26 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 27.3.10, SPI Operation, and section 27.3.11, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to SPDR while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 27.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 14, Interrupt Controller (ICUb), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

27.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 27.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 27.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	SPDR is read when the receive buffer is empty.	Data received previously is output to the bus.	None
3	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is transmitted.	None
4	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
6	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 27.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when an RSPI receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

Similarly, the RSPI does not detect an error on operation 3. 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 3 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).

An overrun error shown in 4 is described in section 27.3.8.1, Overrun Error. A parity error shown in 5 is described in section 27.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 27.3.8.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 27.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

27.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.
Figure 27.27 shows an example of operations of the SPRF and OVRF flags. The SPSR and SPDR accesses shown in Figure 27.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 27.27, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

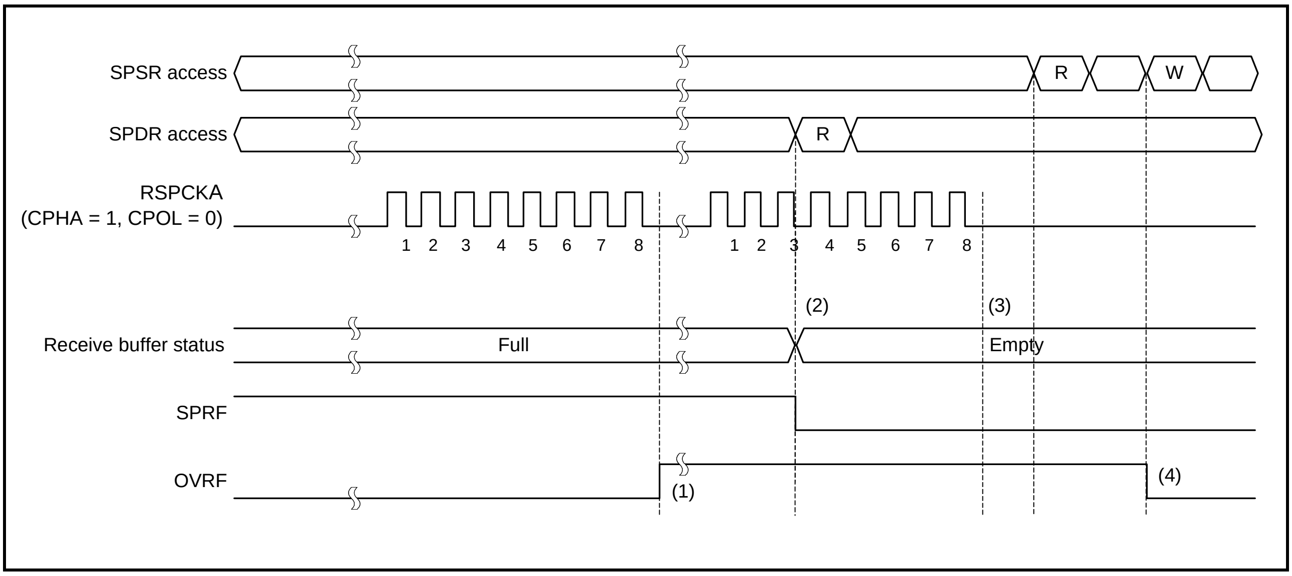


Figure 27.27 Operation Example of SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates with the receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 27.28 and Figure 27.29 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

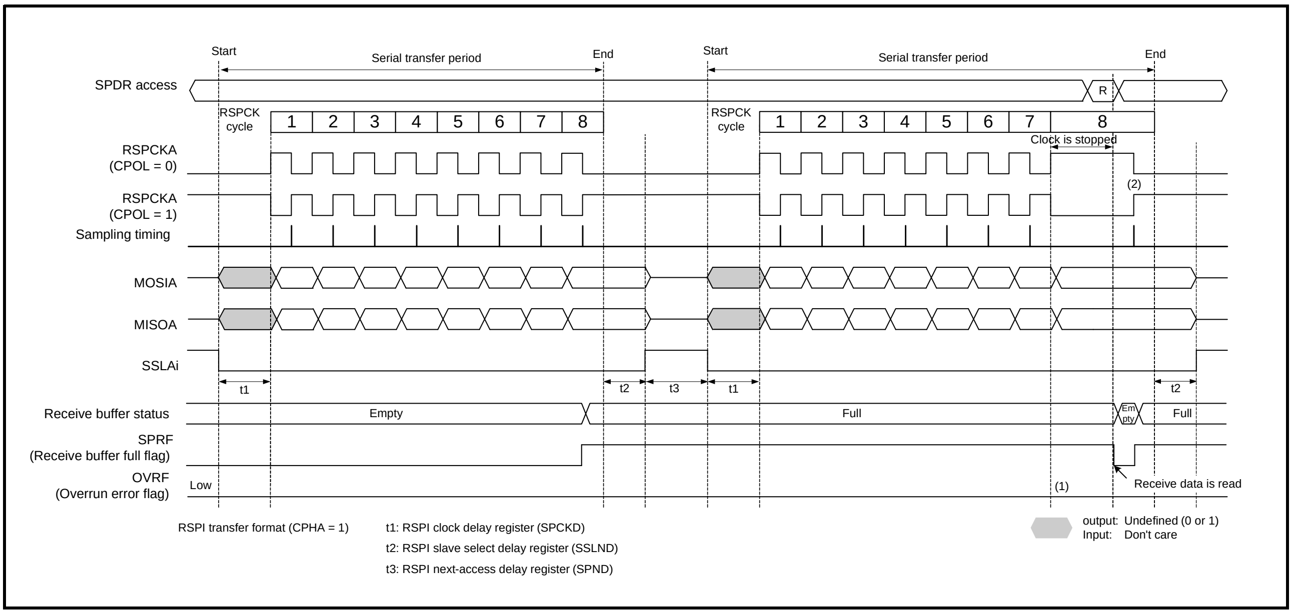


Figure 27.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

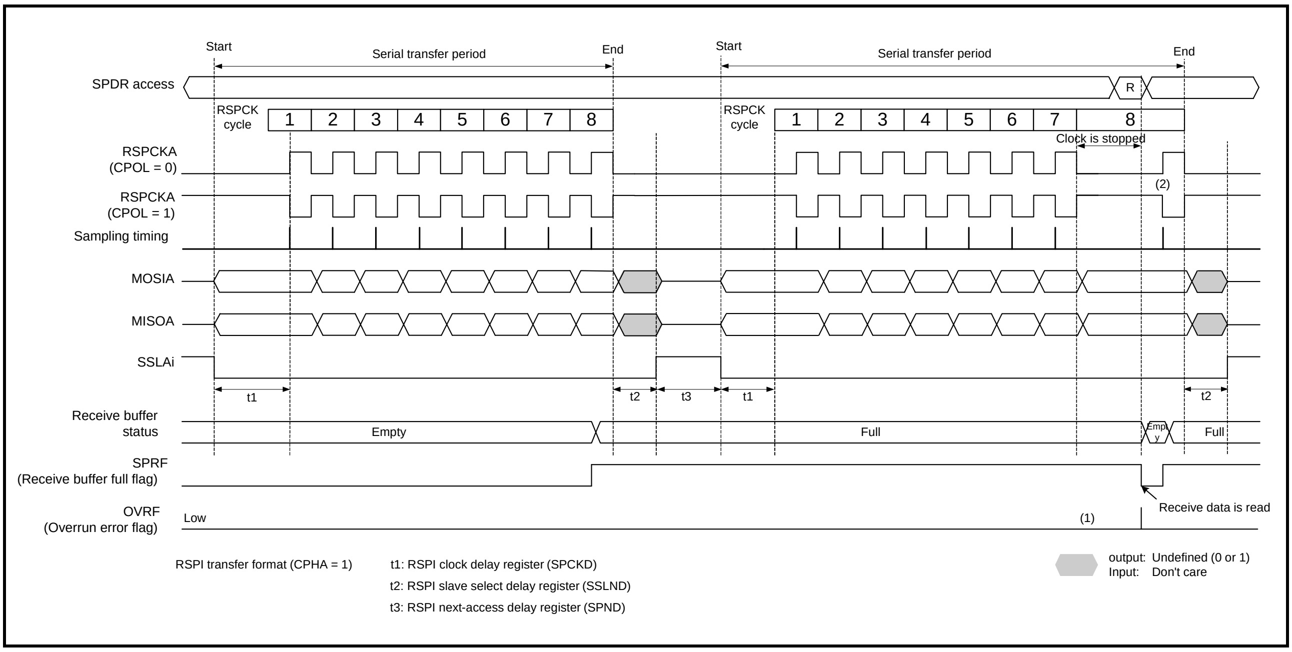


Figure 27.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

27.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1.
Figure 27.30 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 27.30 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 27.30, 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).

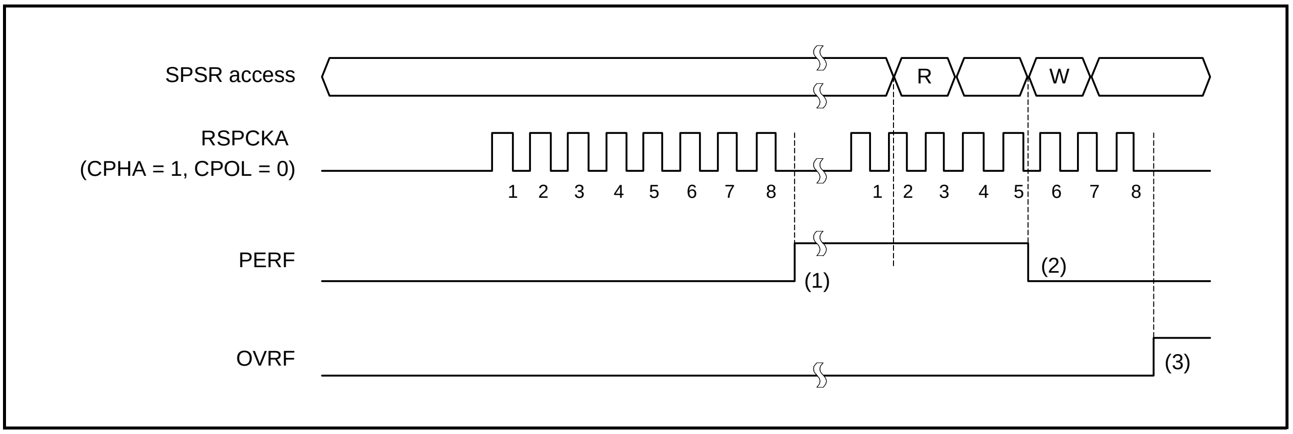


Figure 27.30 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an RSPI error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

27.3.8.3 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 SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 27.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

27.3.9 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

27.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, and SPSR.PERF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

27.3.9.2 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 27.3.9.1, Initialization by Clearing the SPE Bit.

27.3.10 SPI Operation

27.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 27.3.8, 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 (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.

For details on the RSPI transfer format, refer to section 27.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.

For details on the RSPI transfer format, refer to section 27.3.5, 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.

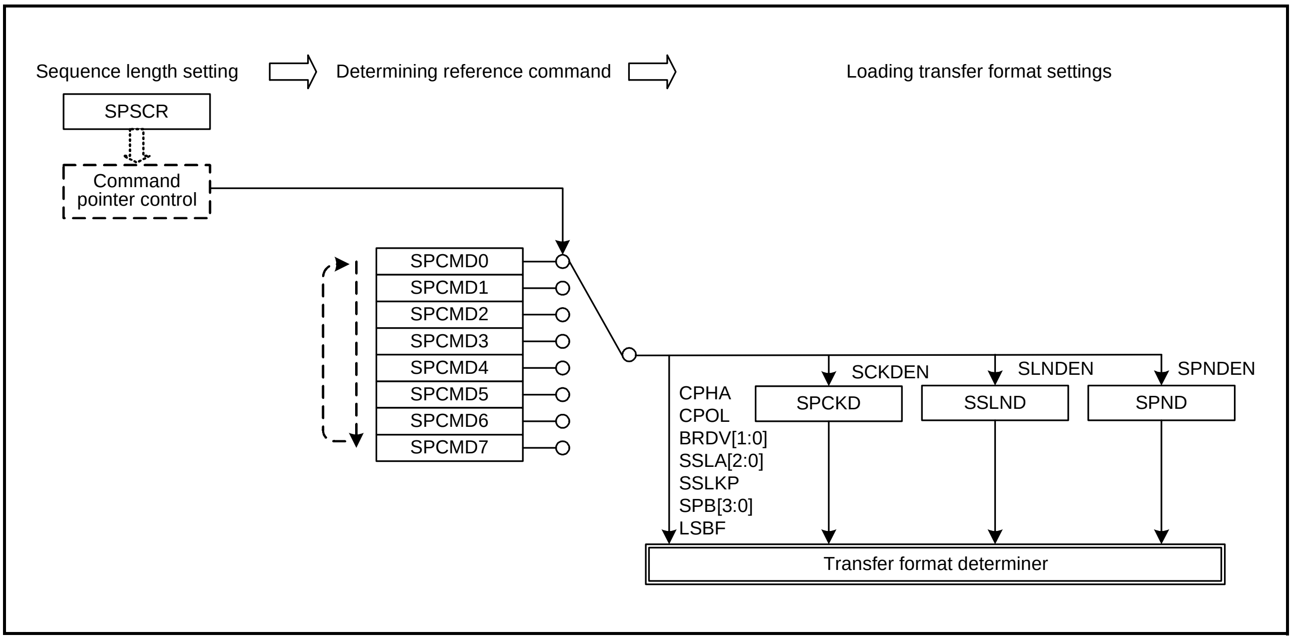


Figure 27.31 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

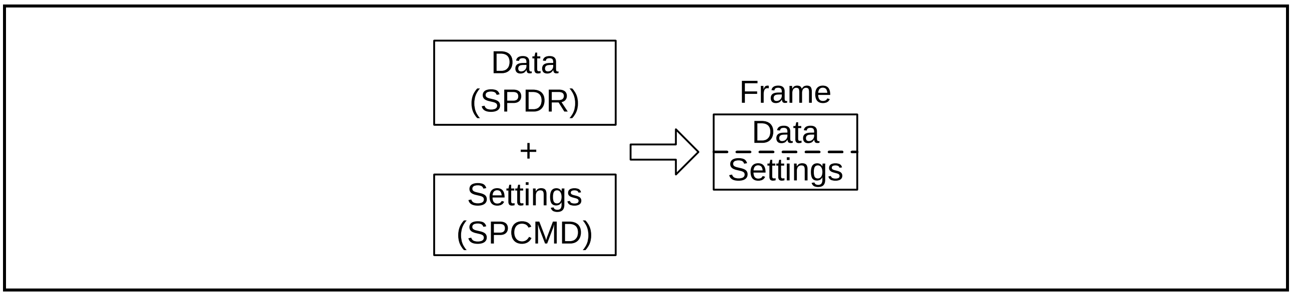


Figure 27.32 Concept of a Frame

Figure 27.33 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 27.4.

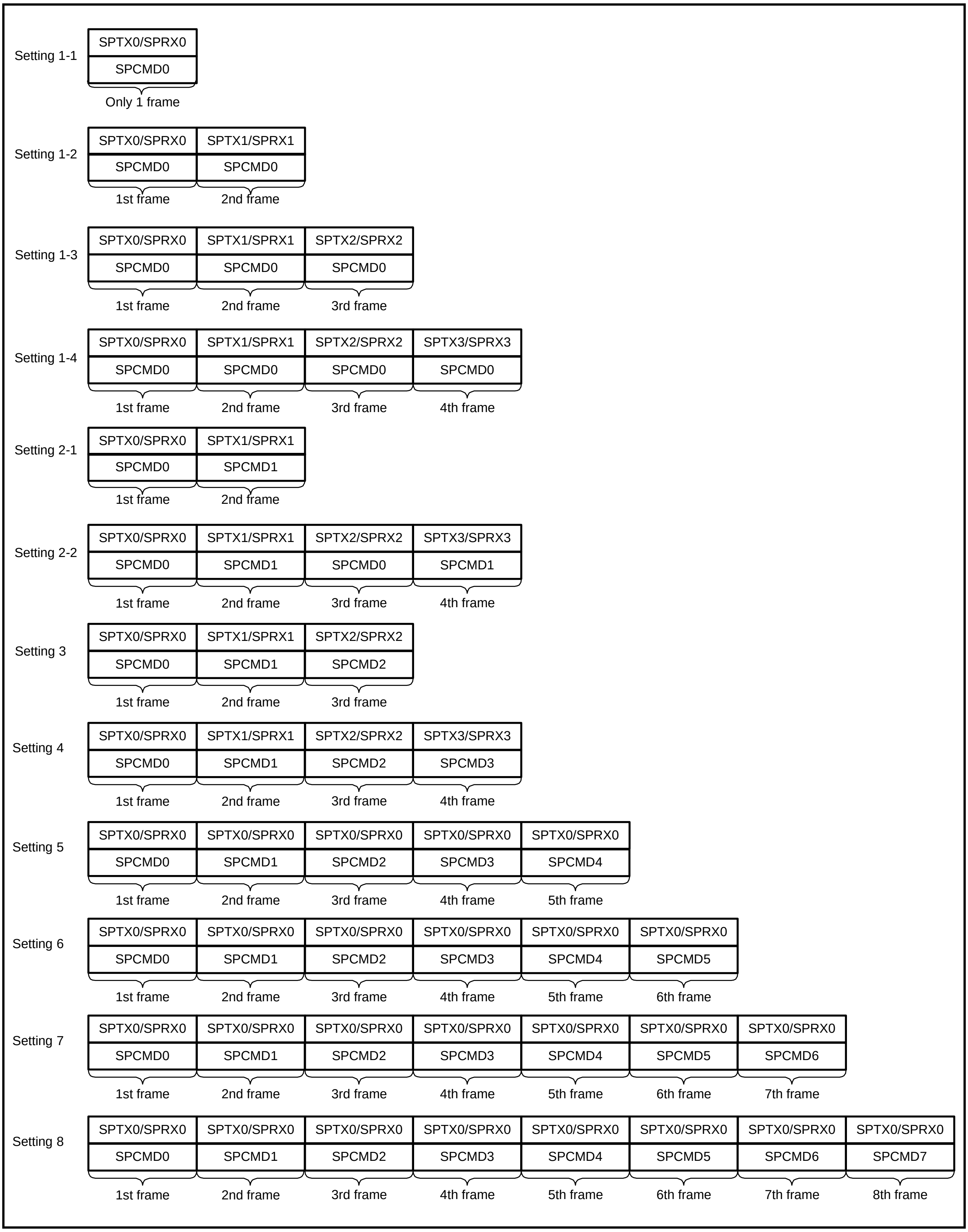


Figure 27.33 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 27.34 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 27.34. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

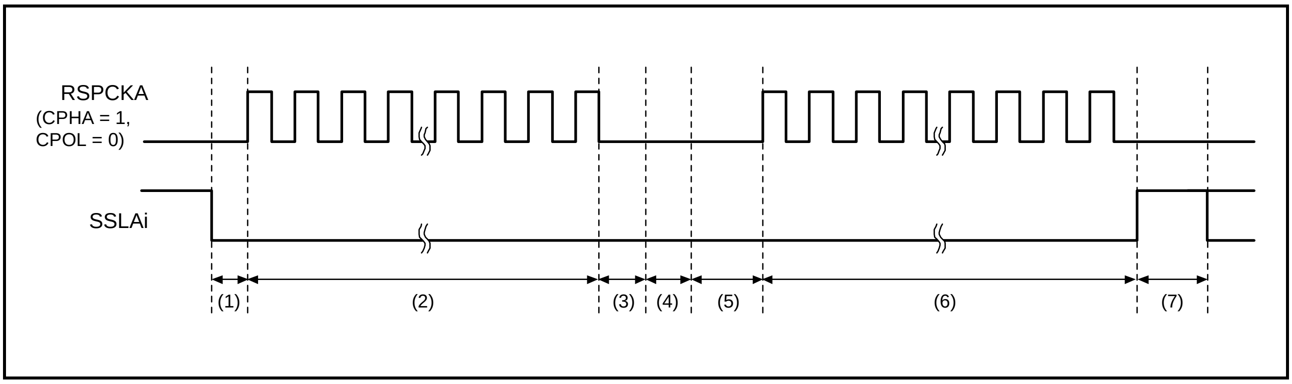


Figure 27.34 Example of Burst Transfer Operation Using SSLKP Bit

(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 27.34) 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.

(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 27.9. For a definition of RSPCK delay, refer to section 27.3.5, Transfer Format.

Table 27.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 27.10. For a definition of SSL negation delay, refer to section 27.3.5, Transfer Format.

Table 27.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 27.11. For a definition of next-access delay, refer to section 27.3.5, Transfer Format.

Table 27.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 27.35 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 and I/O ports, refer to the descriptions given in the individual blocks.

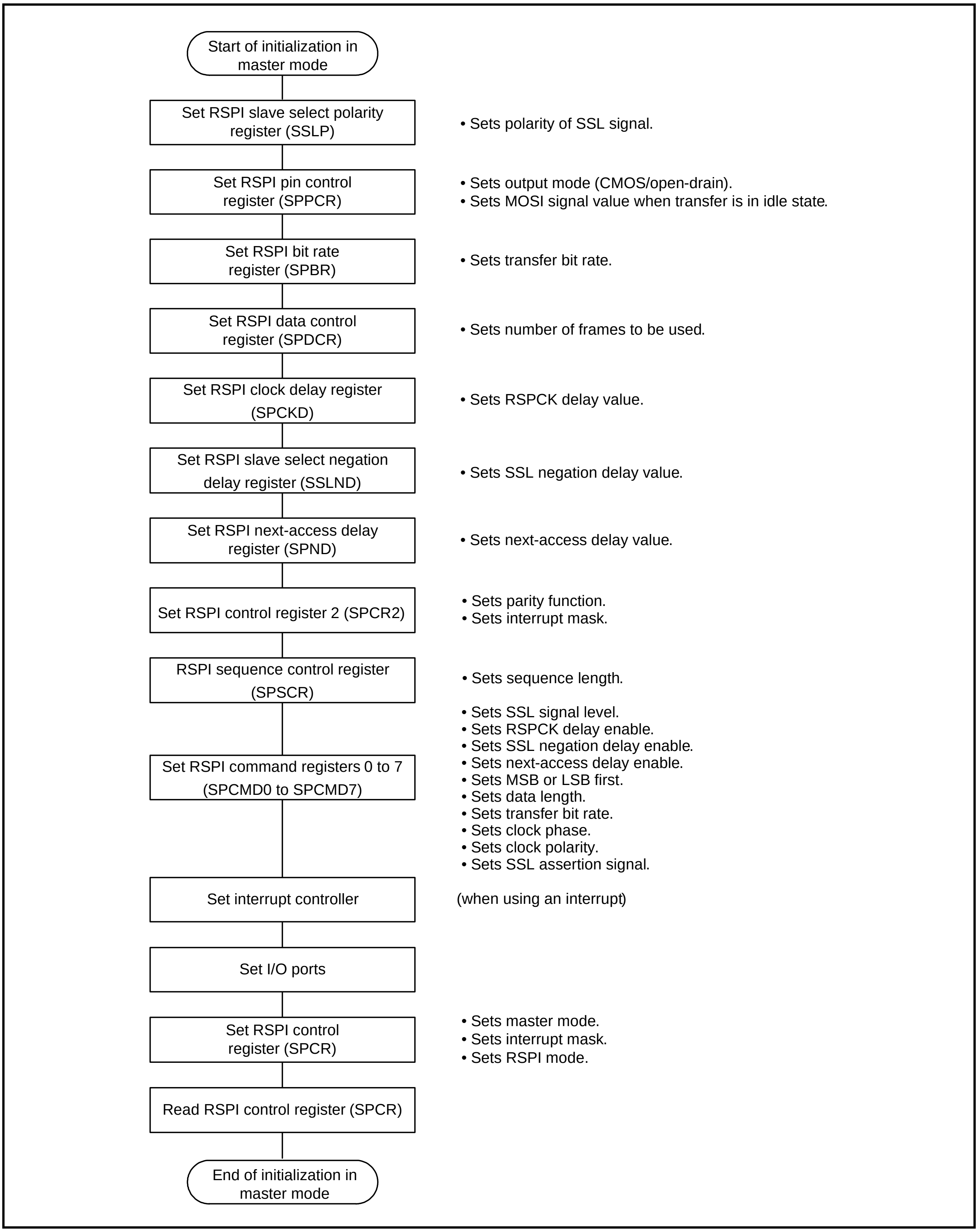


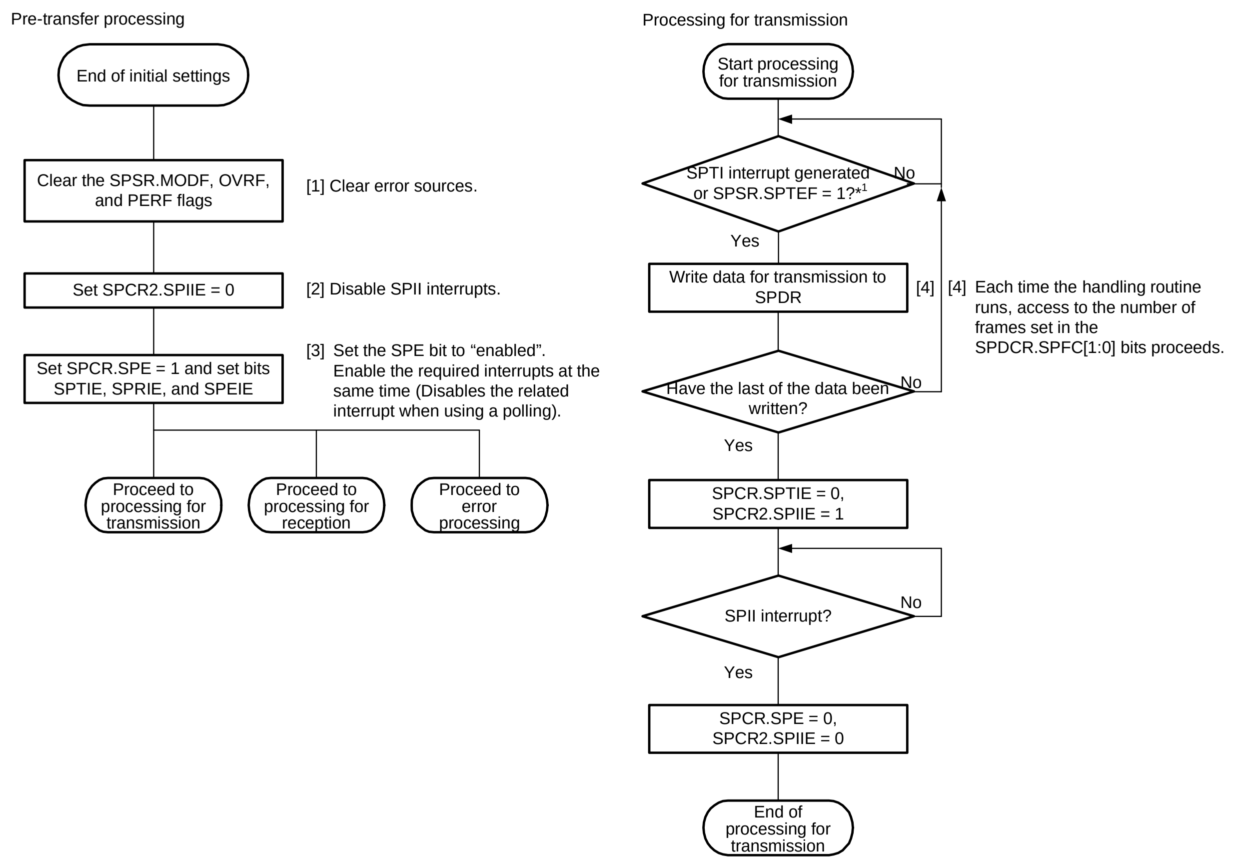
Figure 27.35 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 27.36 to Figure 27.38 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.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Figure 27.36 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

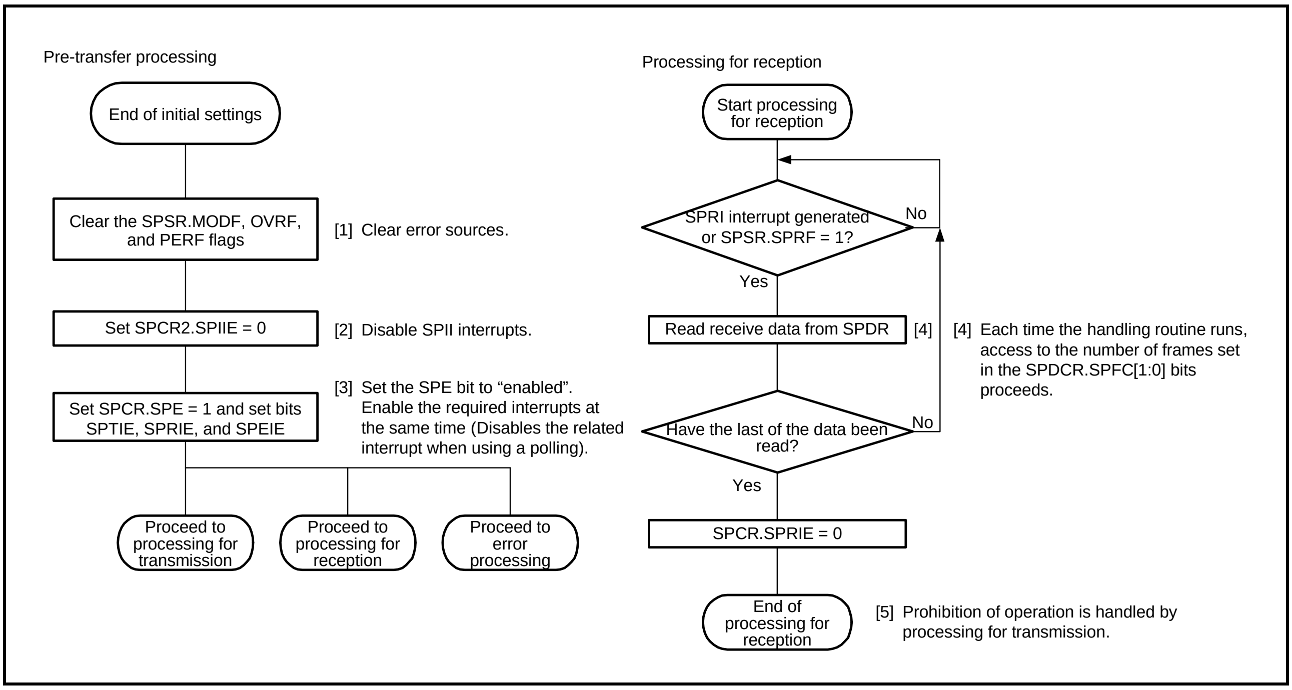


Figure 27.37 Flowchart in Master Mode (Reception)

(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 interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

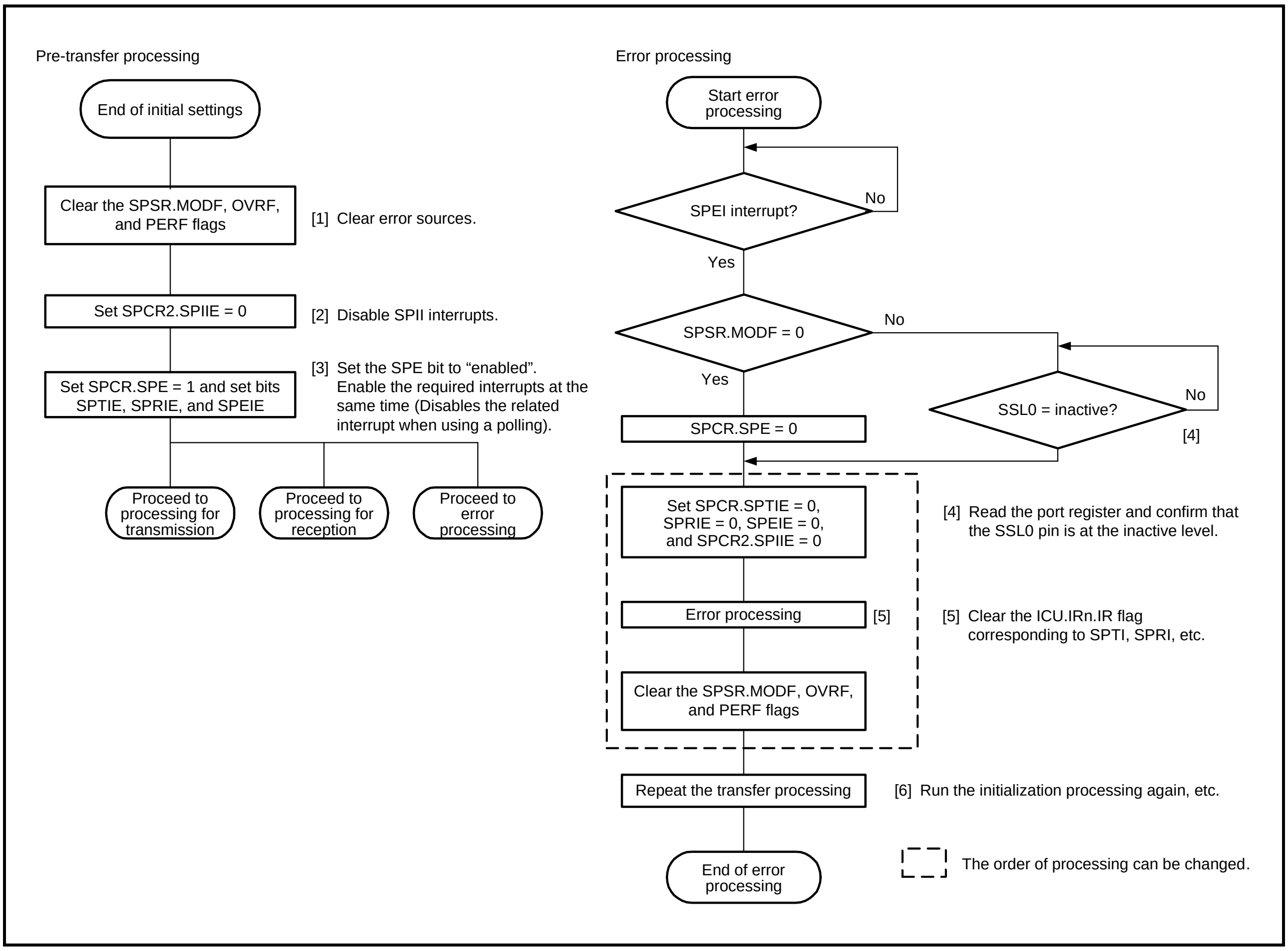


Figure 27.38 Flowchart for Master Mode (Error Processing)

27.3.10.2 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 the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, refer to section 27.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 27.3.8, 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, refer to section 27.3.5, 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 27.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 27.39 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 and I/O ports, refer to the descriptions given in the individual blocks.

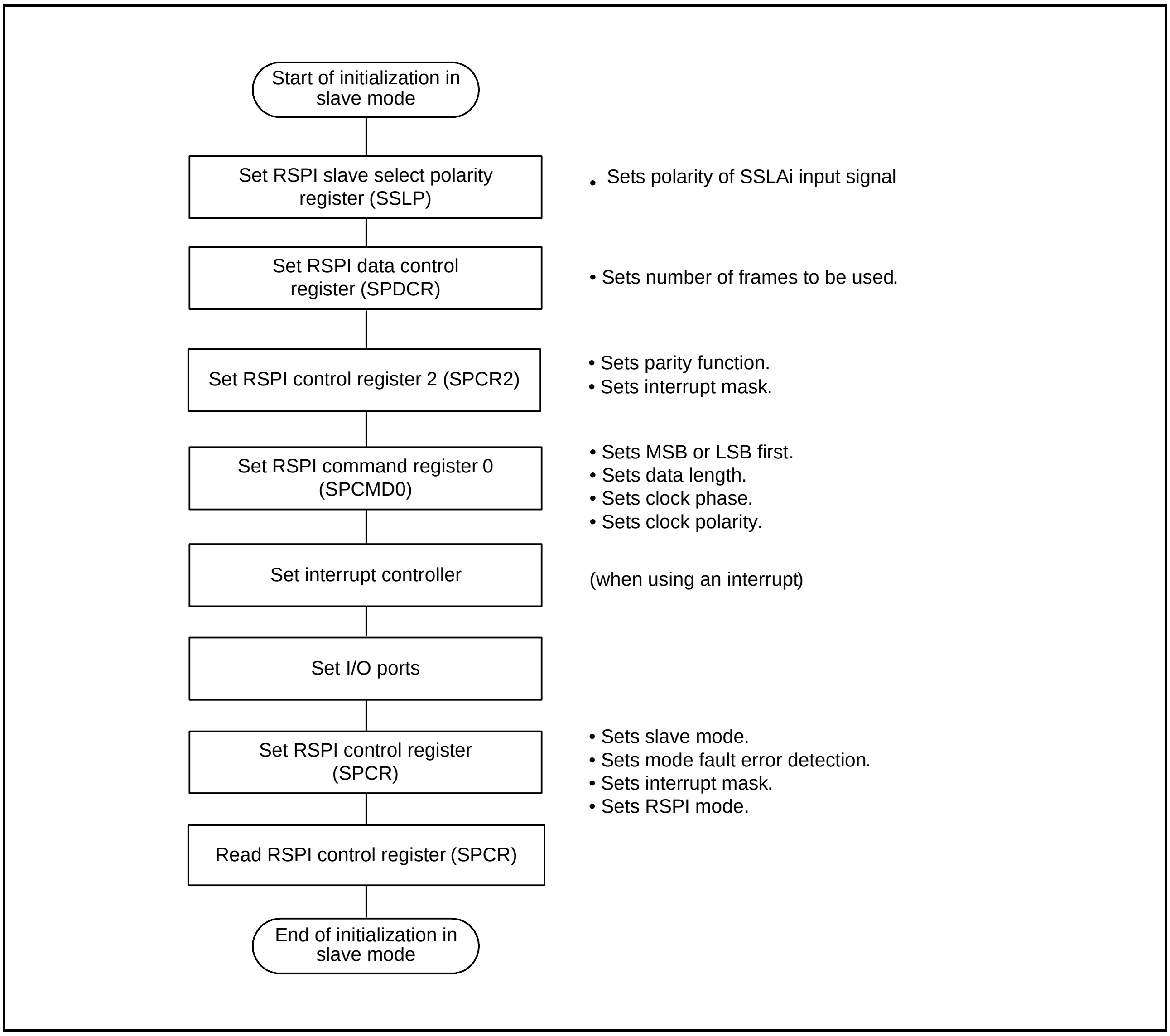


Figure 27.39 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 27.40 to Figure 27.42 show examples of the flow of software processing.

(a) Transmit Processing Flow

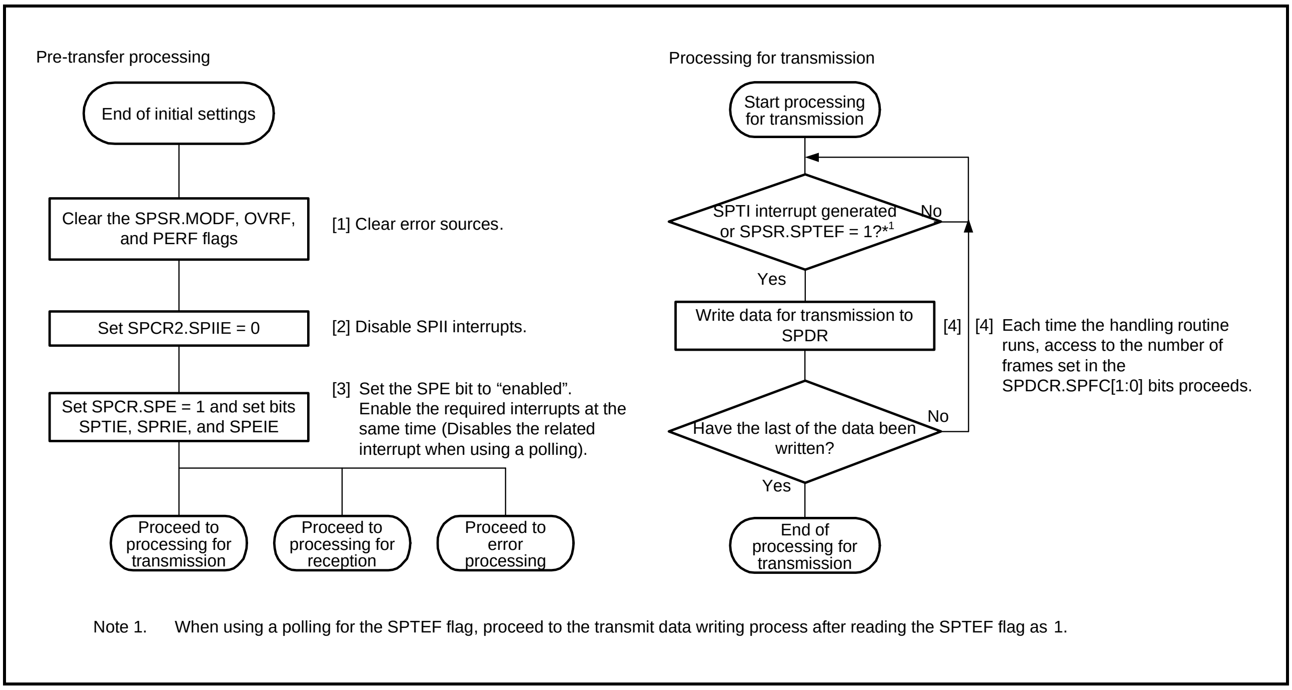


Figure 27.40 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

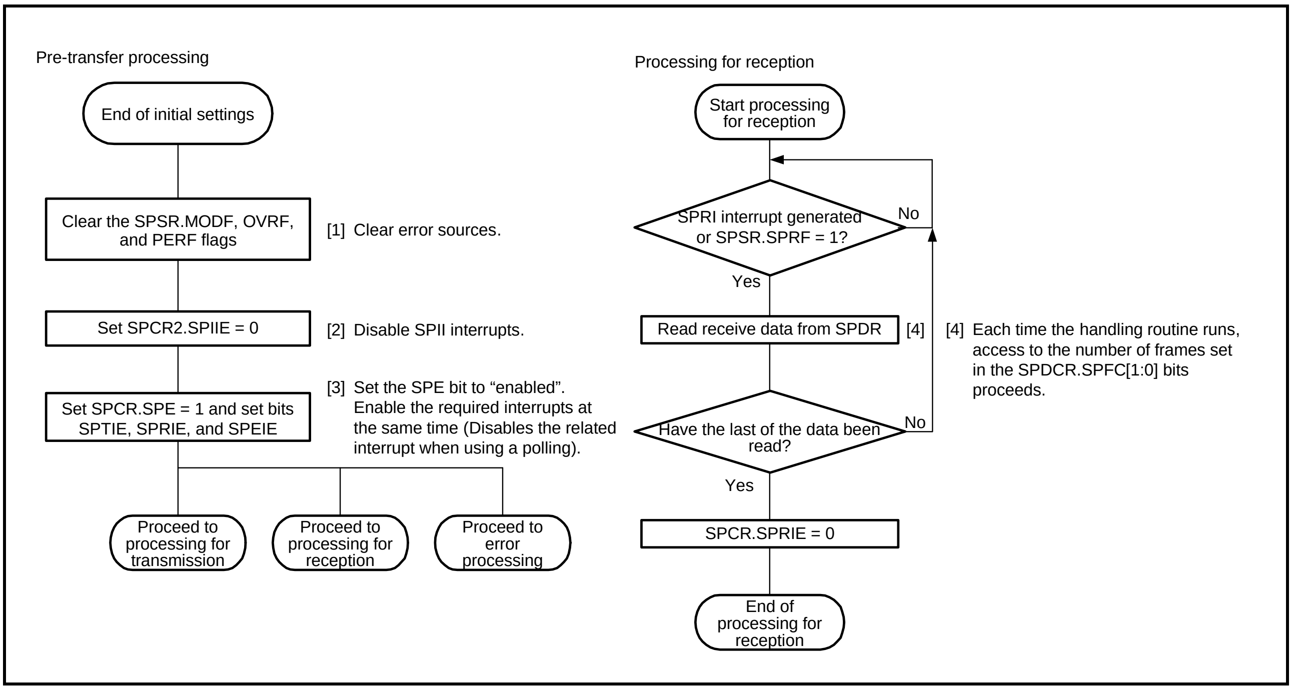


Figure 27.41 Flowchart in Slave Mode (Reception)

(c) Flow of Error Processing

In slave operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

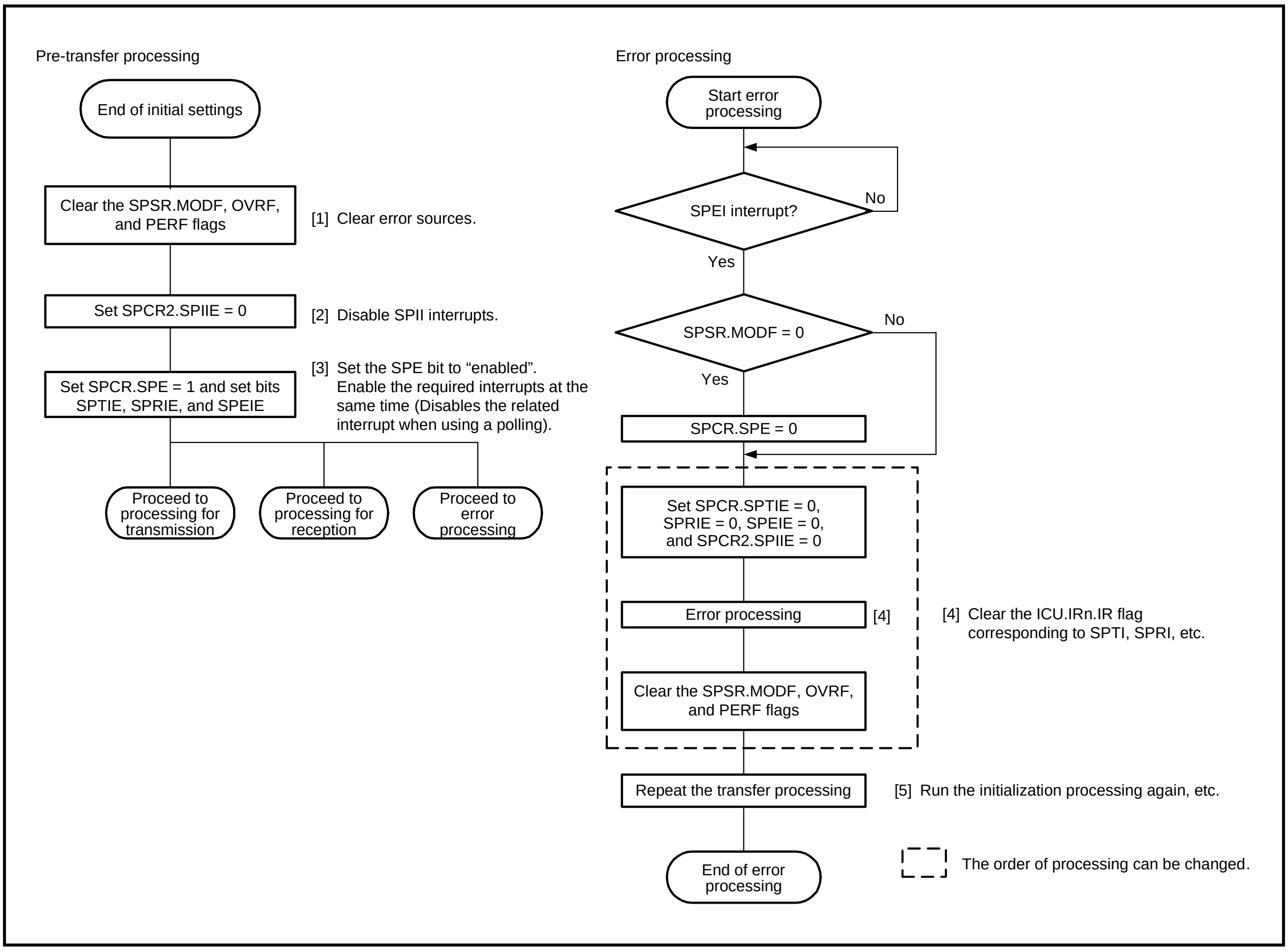


Figure 27.42 Flowchart for Slave Mode (Error Processing)

27.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave 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 should not be performed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

27.3.11.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 27.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 27.3.5, 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 SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

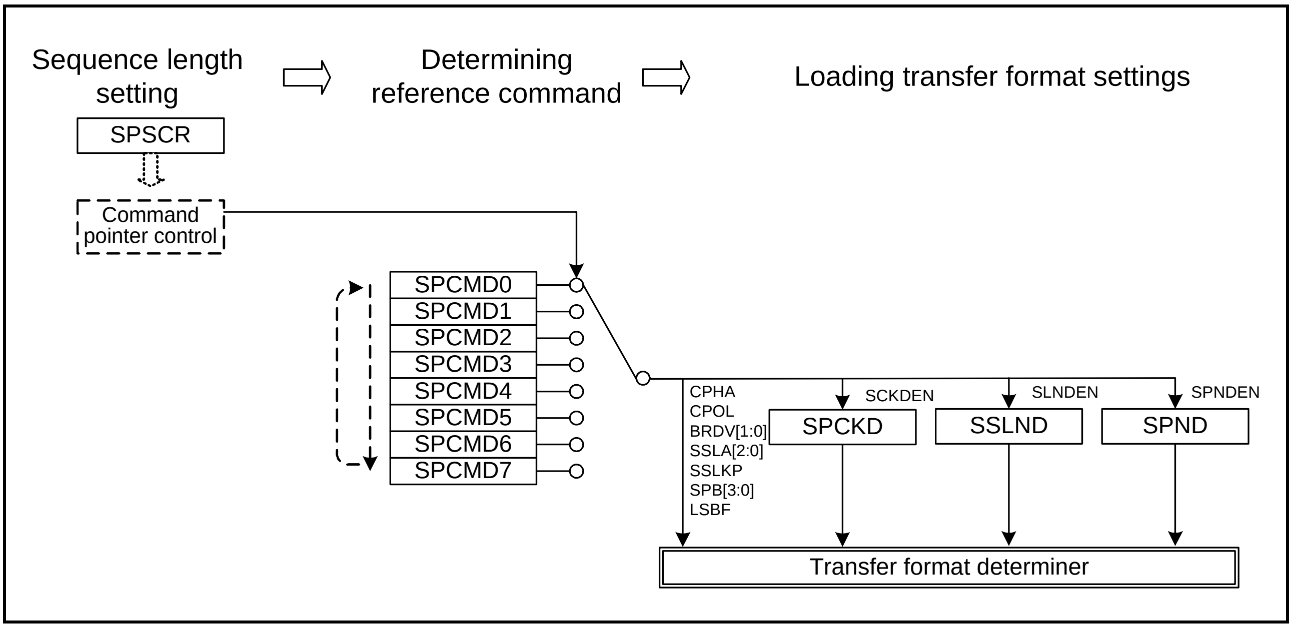


Figure 27.43 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

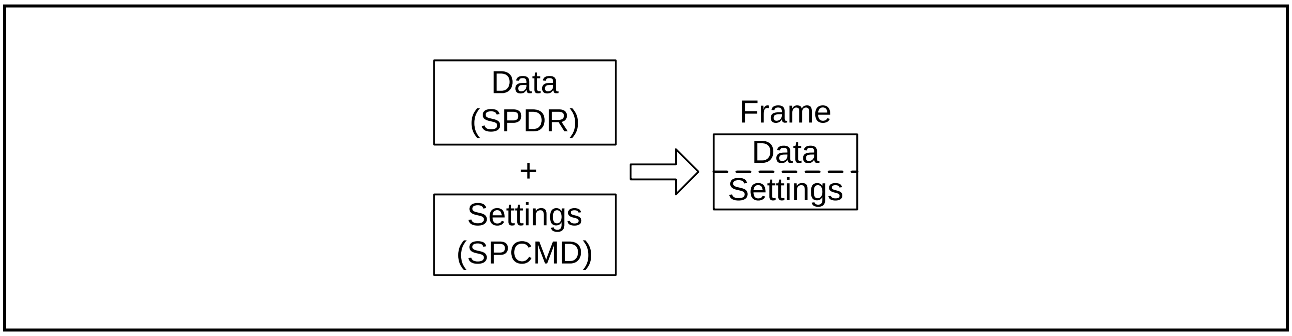


Figure 27.44 Concept of a Frame

Figure 27.45 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 27.4.

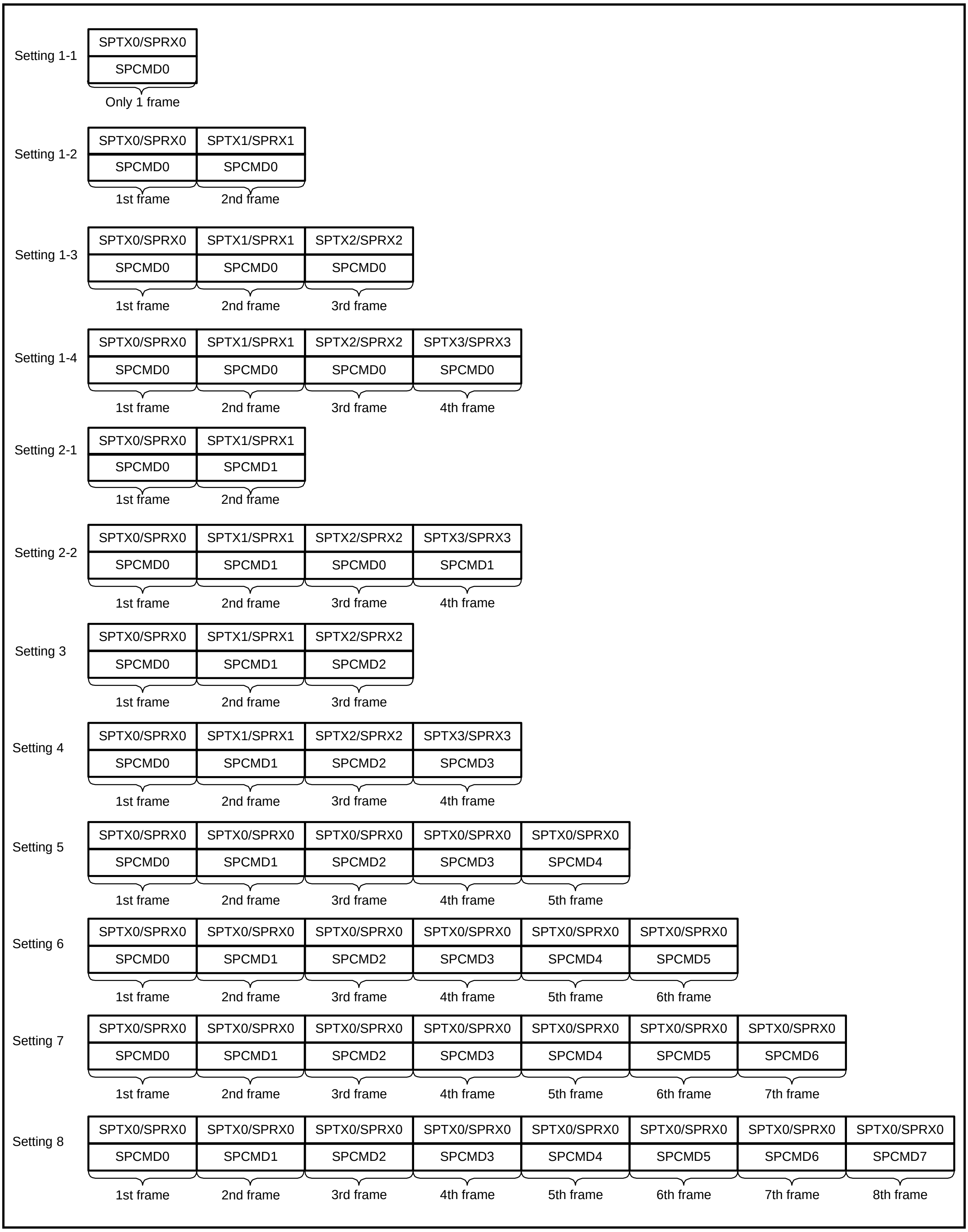


Figure 27.45 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 27.46 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 and I/O ports, refer to the descriptions given in the individual blocks.

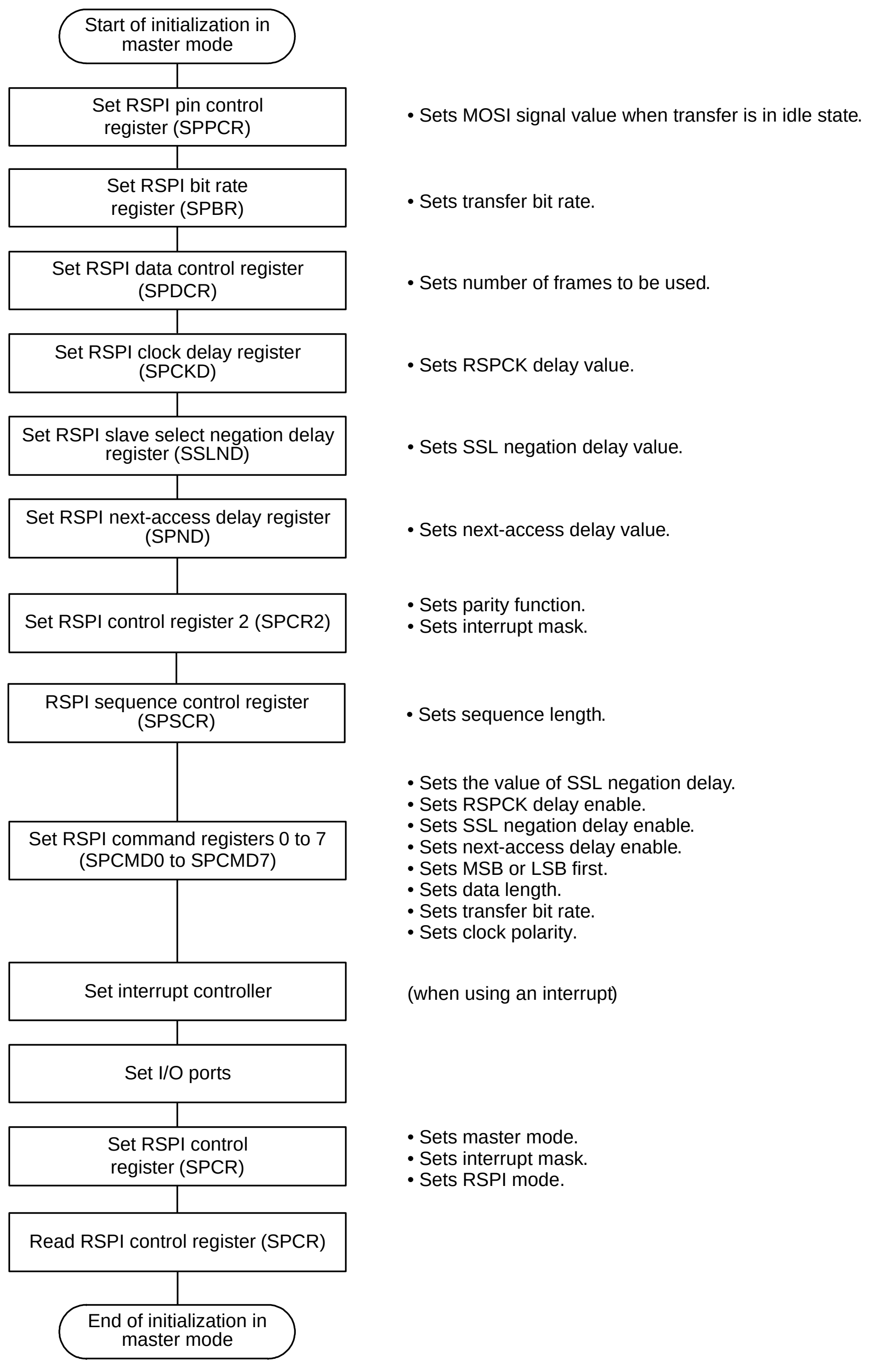


Figure 27.46 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 27.3.10.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

27.3.11.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 27.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing.
When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to "empty" regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 27.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 27.47 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 and I/O ports, refer to the descriptions given in the individual blocks.

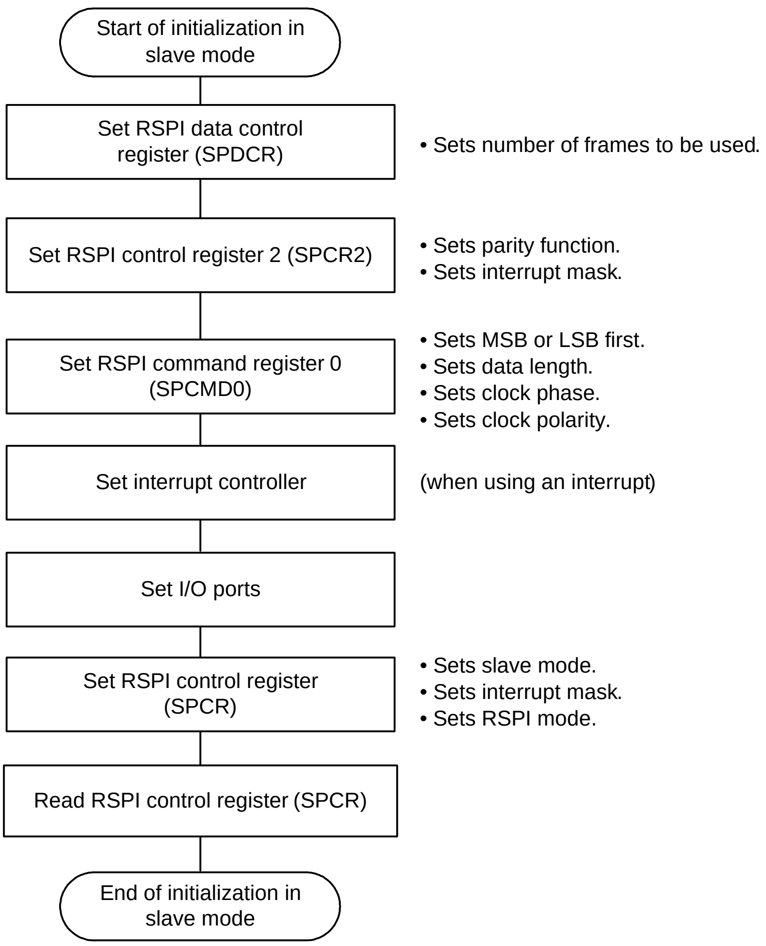


Figure 27.47 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 27.3.10.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

27.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the 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 27.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 27.48 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 27.12 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	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

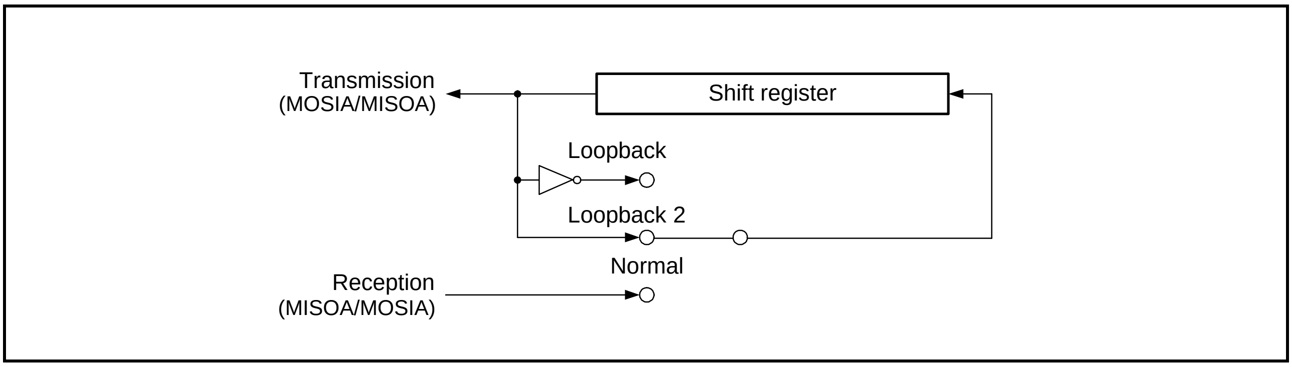


Figure 27.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

27.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 27.49.

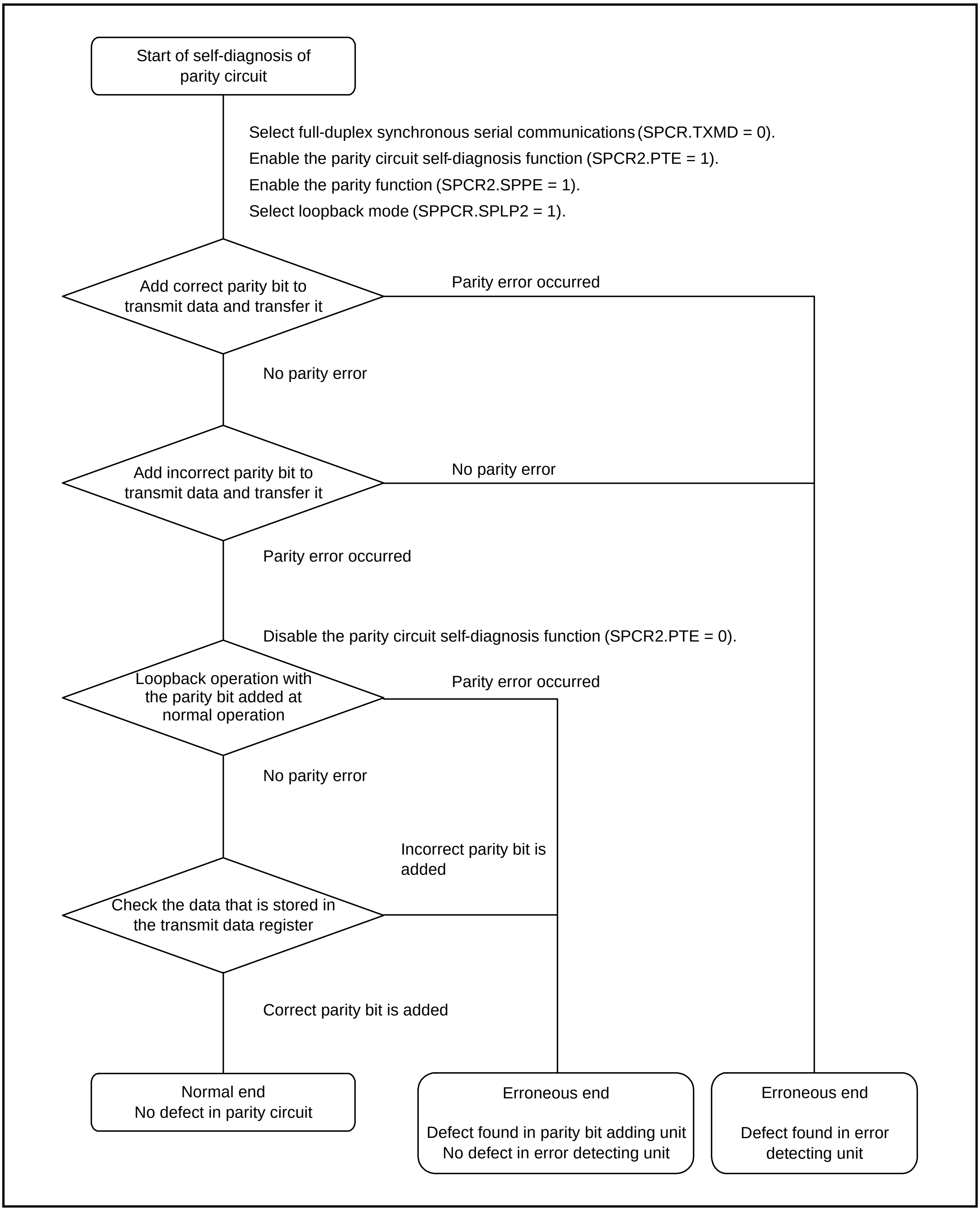


Figure 27.49 Flowchart for Self-Diagnosis of Parity Circuit

27.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC 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 27.13. An interrupt is generated on satisfaction of an interrupt condition in Table 27.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC to perform data transmission/reception, the DTC 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, refer to 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 27.13 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1.	Possible
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

27.4 Usage Notes

27.4.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

27.4.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

27.4.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

27.4.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

28. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

28.1 Overview

Table 28.1 lists the specifications of the CRC calculator, and Figure 28.1 shows a block diagram of the CRC calculator.

Table 28.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	Operation executed on 8 bits in parallel
CRC generating polynomial	One of three generating polynomials selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have functionality to divide data for calculation into CRC calculation units. Write data in 8-bit units.

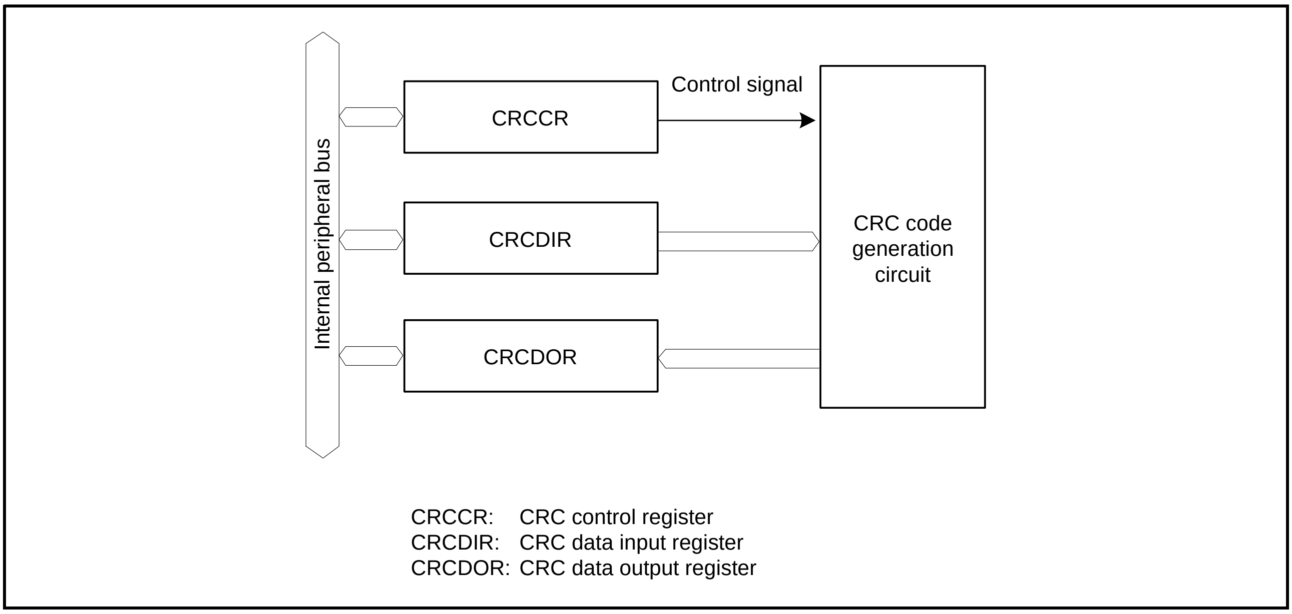


Figure 28.1 CRC Block Diagram

28.2 Register Descriptions

28.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

Set this bit to select the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on transmitting and receiving CRC code, refer to section 28.3, Operation.

28.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

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

CRCDIR is a readable/writable register. Write data for CRC calculation to this register.

28.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

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

CRCDOR is a readable/writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

28.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.

The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of CRCDOR.

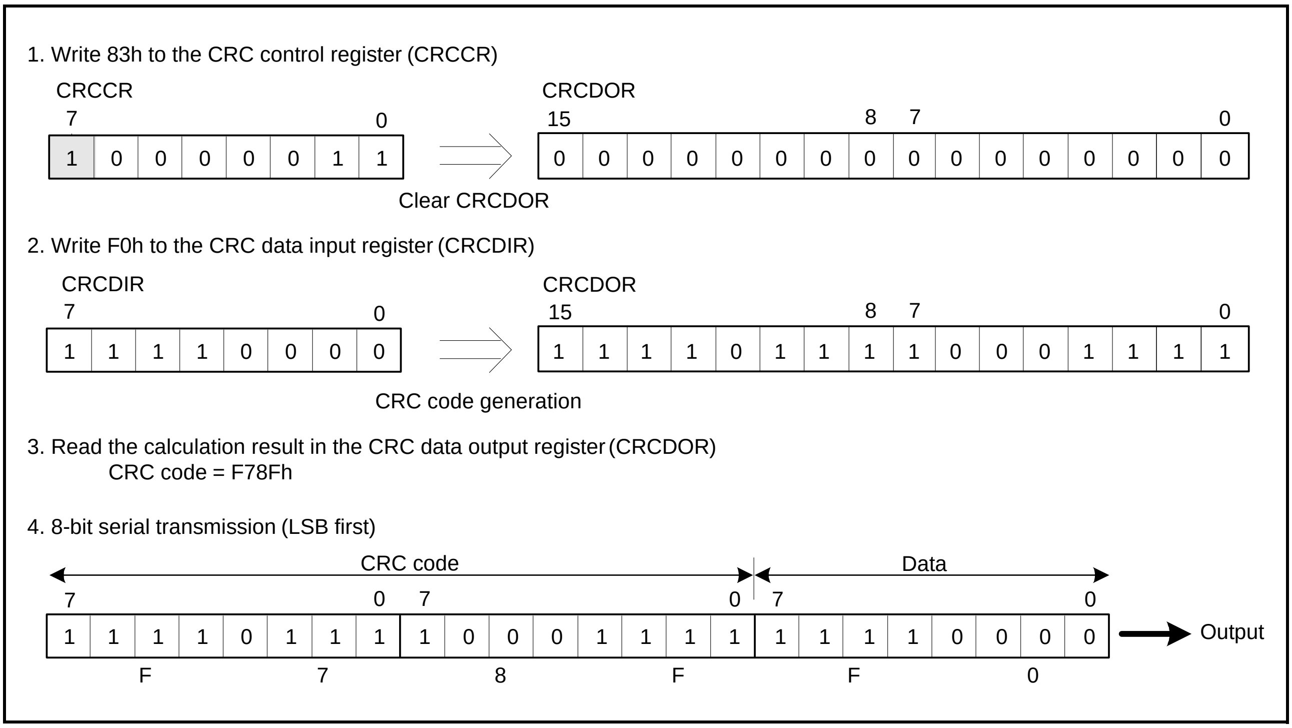


Figure 28.2 LSB First Data Transmission

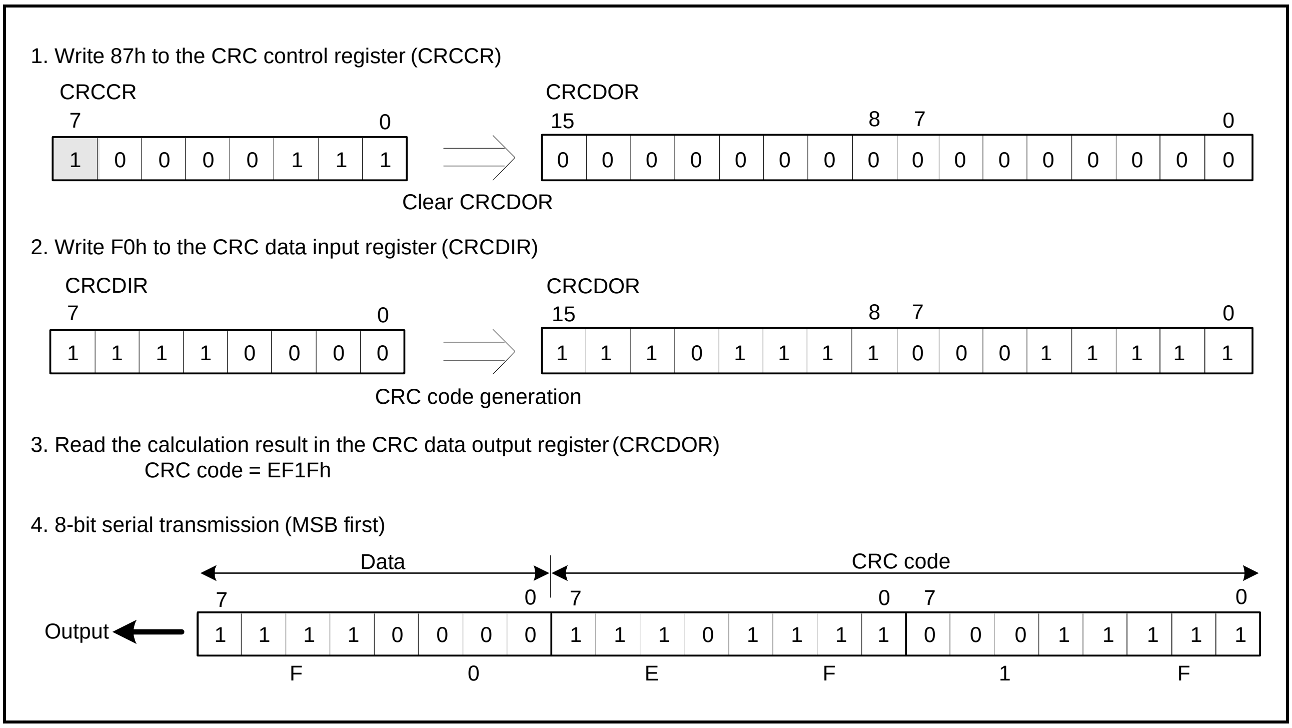


Figure 28.3 MSB First Data Transmission

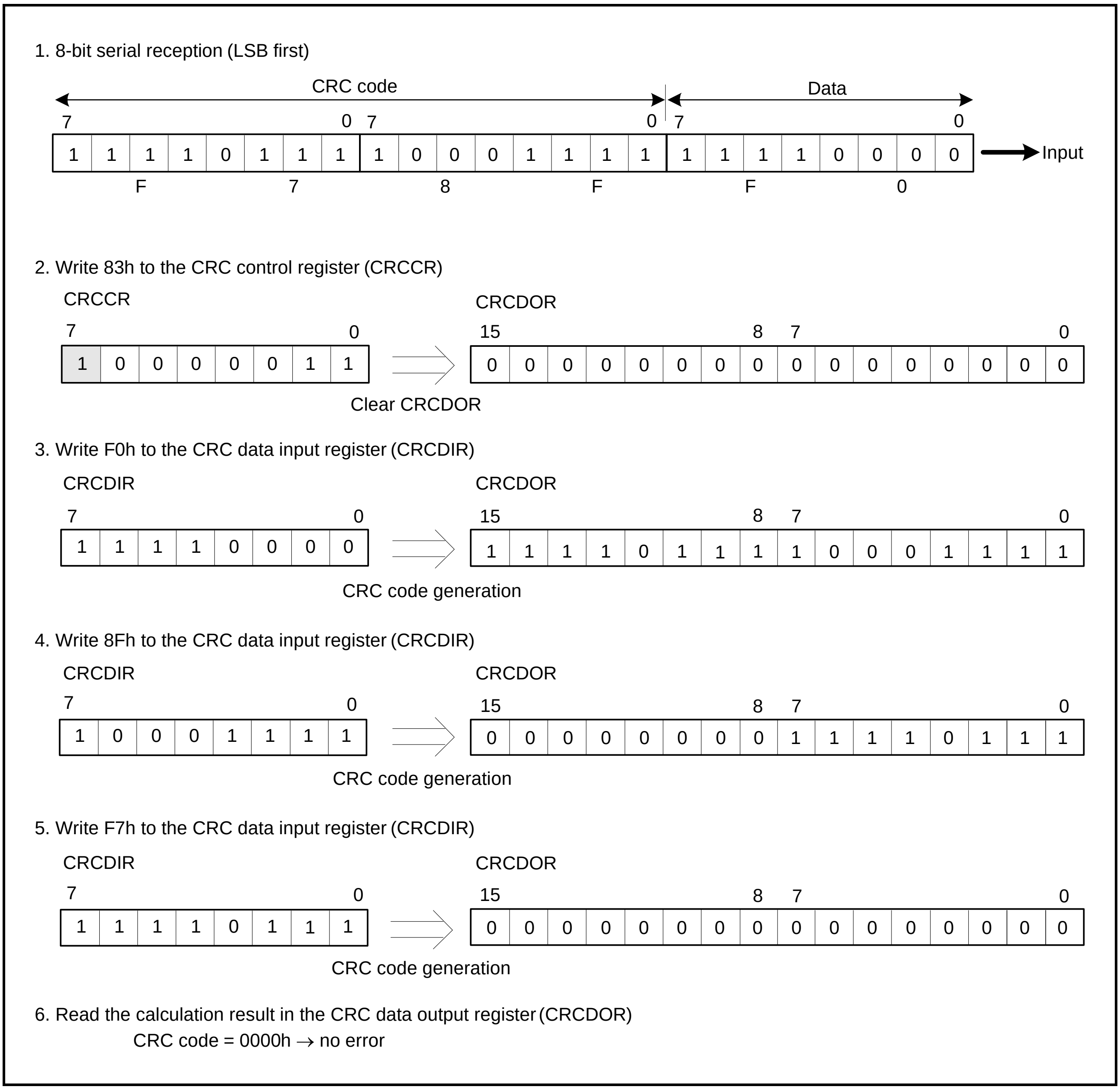


Figure 28.4 LSB First Data Reception

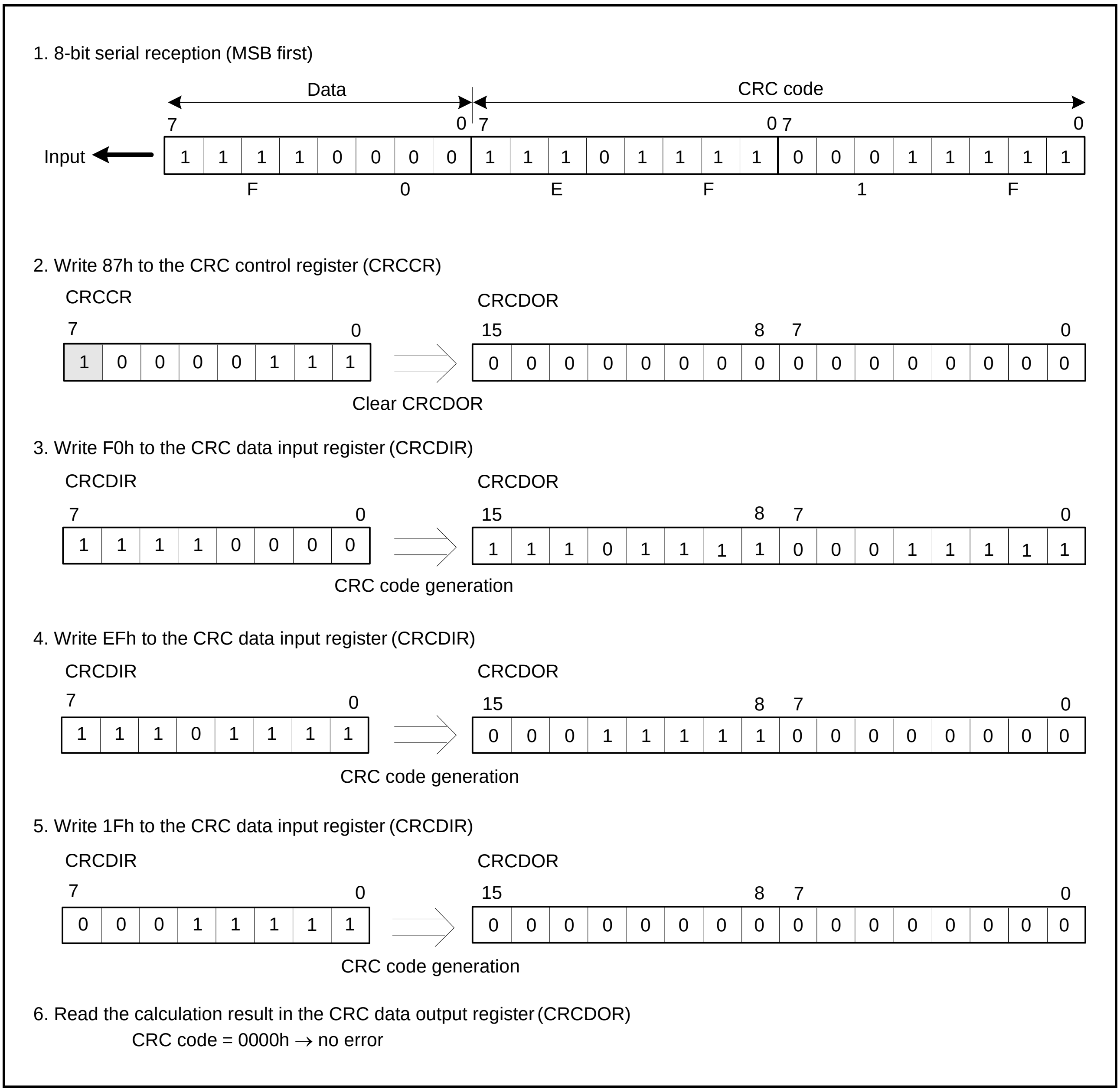


Figure 28.5 MSB First Data Reception

28.4 Usage Notes

28.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

28.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

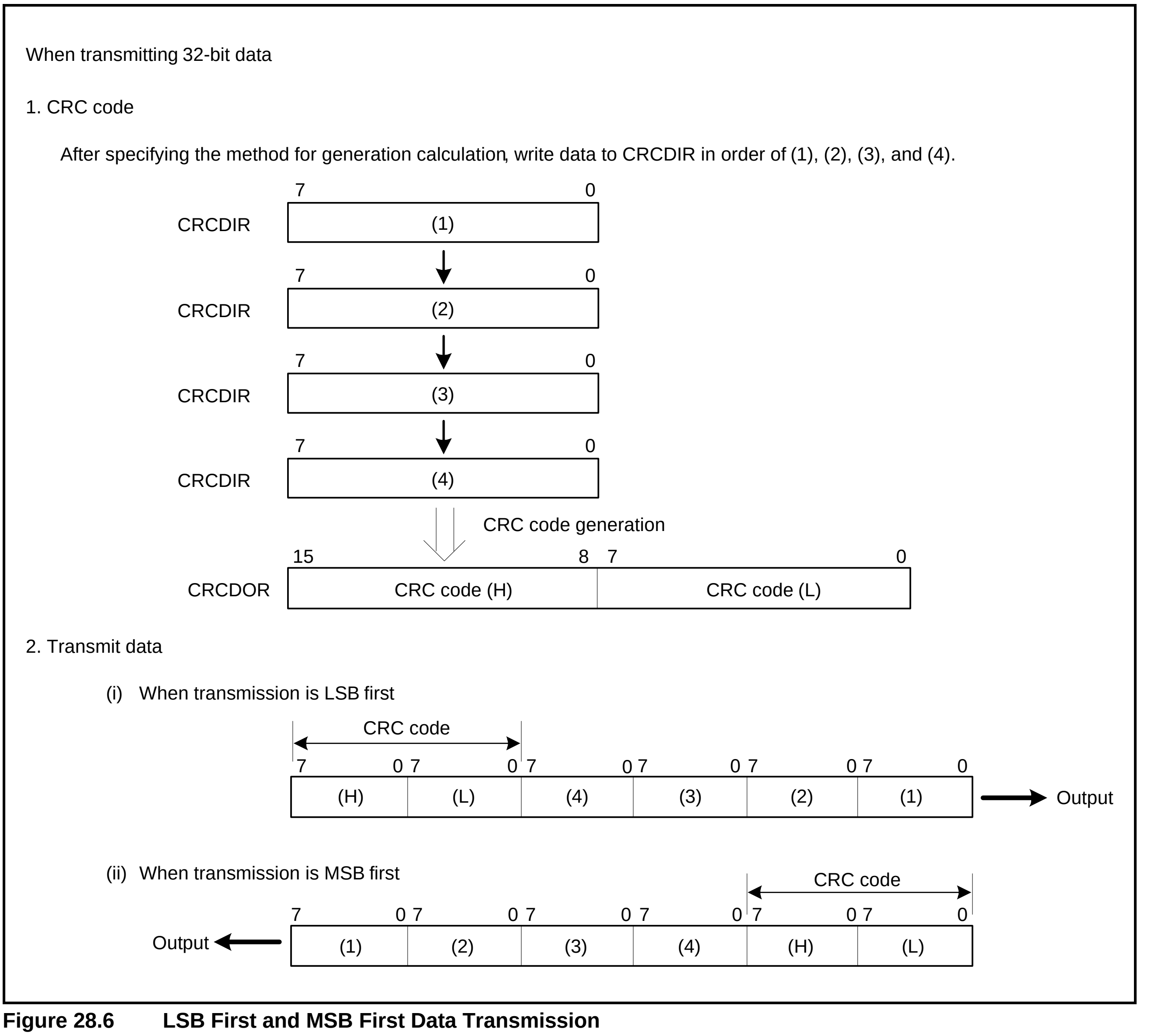


Figure 28.6 LSB First and MSB First Data Transmission

29. 12-Bit A/D Converter (S12ADE)

In this section, “PCLK” is used to refer to PCLKB.

29.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. Up to 10 channel analog inputs and internal reference voltage are selectable for conversion.
The 12-bit A/D converter converts a maximum of 10 selected channels of analog inputs and internal reference voltage, which have been selected, 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 10 arbitrarily selected channels are converted only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 10 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 10 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 conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. When group-A priority control is selected along with operation as described above, if a request to start scanning for group A is received during A/D conversion for group B, the conversion operation for group B is discontinued and the conversion for group A starts, which is given priority.
In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).
Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.
The external pin input (VREFH0) or the analog reference voltage (AVCC0) is selectable as the reference voltage on the high-potential side. The external pin input (VREFL0) or the analog reference voltage (AVSS0) is selectable as the reference voltage on the low-potential side.
Table 29.1 lists the specifications of the 12-bit A/D converter and Table 29.2 lists the functions of the 12-bit A/D converter. Figure 29.1 shows a block diagram of the 12-bit A/D converter.

Table 29.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	One unit
Input channels	Up to 10 channels
Extended analog function	Internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1 μs per channel (when A/D conversion clock ADCLK = 40 MHz)
A/D conversion clock	Peripheral module clock PCLK*[1] and A/D conversion clock ADCLK*[1] can be set so that the frequency ratio should be one of the following. PCLK to ADCLK frequency ratio = 1:1, 1:2, 2:1, 4:1, 8:1 ADCLK is set using the clock generation circuit.
Data registers	• 10 registers for analog input, 1 for A/D-converted data duplication in double trigger mode, and 2 for A/D-converted data duplication during extended operation in double trigger mode • One register for internal reference voltage • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*[2] in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplication register that is prepared for each type of trigger.
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of up to 10 channels arbitrarily selected. 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 10 channels arbitrarily selected. • Group scan mode: Analog inputs of up to 10 channels arbitrarily selected, are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. • Group scan mode (when group A is given priority): If a group A trigger is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A. Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU) or 8-bit timer (TMR). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# pin.
Functions	• Channel-dedicated sample-and-hold function (three channels) • Variable sampling state count • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function of A/D data registers

Table 29.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Interrupt sources	• In the modes except double trigger mode and group scan mode, A/D scan end interrupt request (S12ADI) can be generated on completion of single scan. • In double trigger mode, A/D scan end interrupt request (S12ADI) can be generated on completion of double scan. • In group scan mode, an A/D scan end interrupt request (S12ADI) can be generated on completion of group A scan, whereas an A/D scan end interrupt request (GBADI) for group B can be generated on completion of group B scan. • When double trigger mode is selected in group scan mode, A/D scan end interrupt request (S12ADI) can be generated on completion of double scan of group A, whereas A/D scan end interrupt request (GBADI) specially for group B can be generated on completion of group B scan. • The S12ADI and GBADI interrupts can activate the data transfer controller (DTC).
Low power consumption function	• Module stop state can be set.[*3, *4]

Note 1. The peripheral module clock PCLK frequency is set according to the setting of the SCKCR.PCKB[3:0] bits and the A/D conversion clock ADCLK frequency is set according to the setting of the SCKCR.PCKD[3:0] bits.
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (addition zero to three times)
4-bit extension: 16-time conversion (addition 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 29.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation
Analog input channels			AN000 to AN007, AN016, AN017, internal reference voltage
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger	ADTRG0#	Enabled
	Synchronous trigger	Compare match/input capture from MTU0.TGRA	TRGA0N
		Compare match/input capture from MTU1.TGRA	TRGA1N
		Compare match/input capture from MTU2.TGRA	TRGA2N
		Compare match/input capture from MTU3.TGRA	TRGA3N
		Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N
		Compare match from MTU0.TGRE	TRG0N
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN
		TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0
		TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1
Channel-dedicated sample-and-hold function		Target channel	AN000 to AN002
Interrupt			S12ADI, GBADI interrupt
Setting of module stop function[*1]			MSTPCRA. MSTPA17 bit

Note 1. See section 11, Low Power Consumption for details.

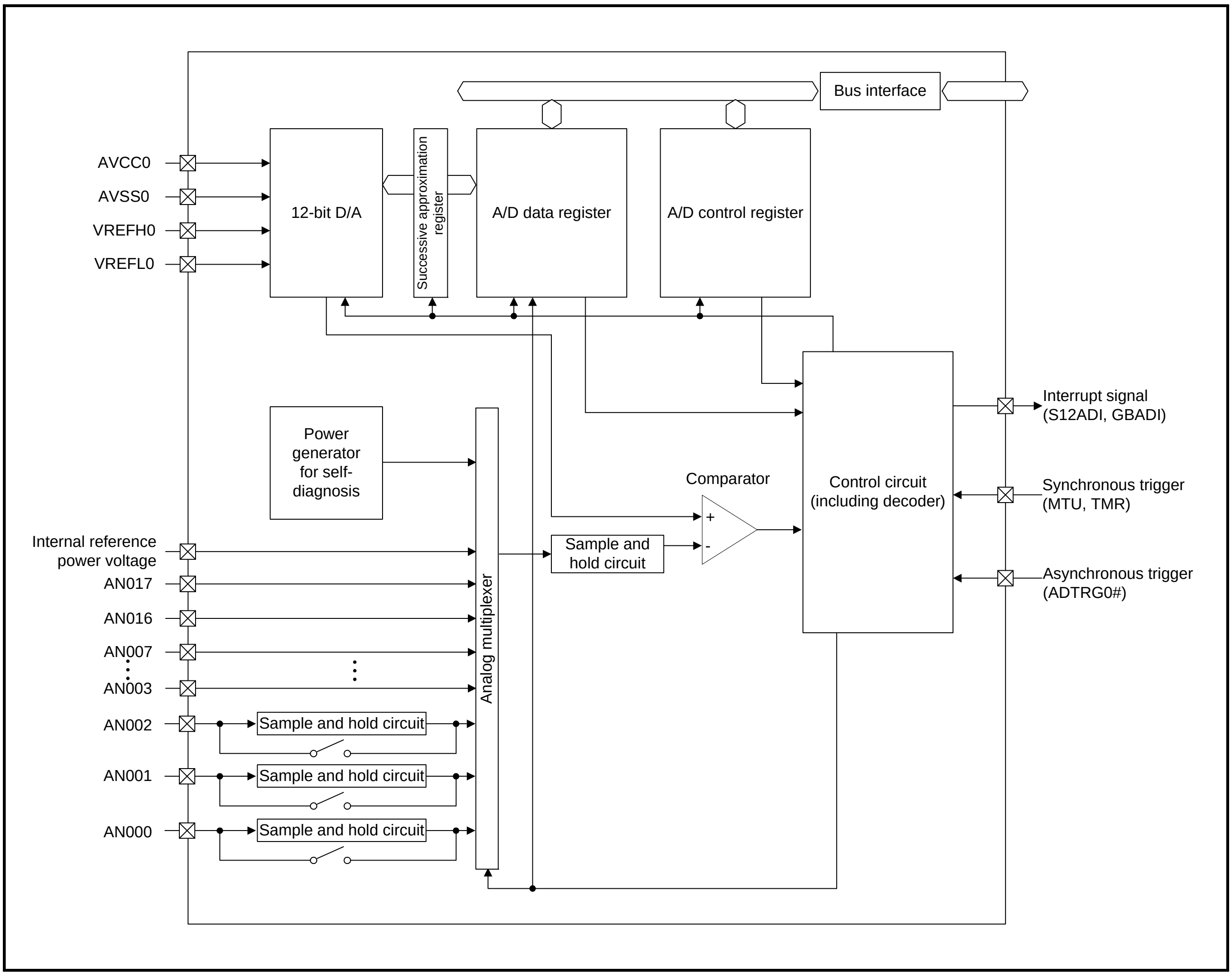


Figure 29.1 Block Diagram of 12-Bit A/D Converter

Table 29.3 lists the input pins of the 12-bit A/D converter.

Table 29.3 Pin Configuration of 12-Bit A/D Converter

Pin Name	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Reference power supply pin
VREFL0	Input	Reference power supply ground pin
AN000 to AN007, AN016 and AN017	Input	Analog input pins 0 to 7, analog input pins 16 and 17
ADTRG0#	Input	External trigger input pin for starting A/D conversion
ADST0	Output	ADST bit state output pin

29.2 Register Descriptions

29.2.1 A/D Data Registers y (ADDRy)
A/D Data Duplication Register (ADDBLDR)
A/D Data Duplication Register A (ADDBLDRA)
A/D Data Duplication Register B (ADDBLDRB)
A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDR7 0008 902Eh, S12AD.ADDR16 0008 9040h, S12AD.ADDR17 0008 9042h, S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h, S12AD.ADOCDR 0008 901Ch

	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

ADDRy (y = 0 to 7, 16, 17) are 16-bit read-only registers which store the A/D conversion results.
ADDBLDR is a 16-bit read-only register used in double trigger mode. ADDBLDR stores the results of A/D conversion when the conversion is started by the second trigger.
ADDBLDRA and ADDBLDRB are 16-bit read-only registers that store the A/D conversion results in response to the respective triggers during extended operation in double trigger mode.
ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when twice or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)

The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

29.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh

	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

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 29.2.11, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

Table 29.4 Self-Diagnosis Status Description

Bits 15 and 14 for flush-right format setting Bits 1 and 0 for flush-left format setting	**Self-diagnosis status**
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of reference power supply × 1/2 has been executed.
11b	Self-diagnosis using the voltage of reference power supply has been executed.

Note: For details of self-diagnosis, see section 29.2.11, A/D Control Extended Register (ADCER).

29.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 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]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: 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 synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI interrupt generation upon scan completion. 1: Enables S12ADI 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)
After a high-level signal is input to the external pin (ADTRG0#), write 1 to both the TRGE and EXTRG bits in ADCSR and change the signals of ADTRG0# to low. Thus 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 clock cycles of PCLK.

ADCSR sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 29.5 shows selection of the channel for double triggered operation.
When double trigger mode is selected, the channels selected by the ADANSA0 and ADANSA1 registers are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.
When double trigger mode is used, do not select A/D conversion for the self-diagnosis function and internal reference voltage (analog input of multiple channels can be selected for A/D conversion for group B in group scan mode). 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.
To enter A/D-converted value addition/average mode while double trigger mode is set, the channel selected by the

DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

Table 29.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000b	AN000	10000b	AN016
00001b	AN001	10001b	AN017
00010b	AN002		
00011b	AN003		
00100b	AN004		
00101b	AN005		
00110b	AN006		
00111b	AN007		

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)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.

When double trigger mode is selected, the channels specified in the ADANS0 and ADANS1 registers are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger.

The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.

In continuous scan mode, double trigger mode should not be selected.

The DBLE bit should be set after the ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion.

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 (S12ADI) in scans except for group B scan in group scan mode.

With double trigger mode deselected, the S12ADI interrupt is generated after the first scan is completed if the ADIE bit is set to 1.

With double trigger mode selected, the S12ADI interrupt is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 10 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.

In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 10 channels selected with the ADANSA0 and ADANSA1 registers 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 from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.

In group scan mode, A/D conversion is performed for the analog inputs (group A) of a 10 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, 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 10 channels selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.

When selecting group scan mode, different channels and triggers should be selected for group A and group B.

When selecting the internal reference voltage, select single scan mode, and deselect all the channels selected with the ADANSA0 and ADANSA1 registers before performing A/D conversion. When A/D conversion of the selected internal reference voltage is completed, A/D conversion is stopped.

The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.

[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] 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[5:0] bits being set to 000000b.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of group B is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels or the internal reference voltage is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B started by a resumption trigger is completed.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRP bit is set to 1 and the scanning of group B by a trigger is completed.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

29.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): S12AD.ADANSA0 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA0 07	ANSA0 06	ANSA0 05	ANSA0 04	ANSA0 03	ANSA0 02	ANSA0 01	ANSA0 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSA0 selects analog input channels for A/D conversion among AN000 to AN007. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSA0n bit selects analog input channels for A/D conversion among AN000 to AN007. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA007 bit corresponds to AN007.

When performing A/D conversion of the internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA0n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA0n bit should be set while the ADCSR.ADST bit is 0.

29.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD.ADANSA1 0008 9006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSA101	ANSA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA100	A/D Conversion Channel Select	0: AN016 and AN017 are not subjected to conversion. 1: AN016 and AN017 are subjected to conversion.	R/W
b1	ANSA101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSA1 selects analog input channels for A/D conversion among AN016 and AN017. In group scan mode, group A channels are to be selected.

ANSA1n Bit (n = 00, 01) (A/D Conversion Channel Select)

The ANSA1n bit (n = 00, 01) select analog input channels for A/D conversion among AN016 and AN017. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN016 and the ANSA101 bit corresponds to AN017.

When performing A/D conversion of the internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA1n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA1n bit should be set while the ADCSR.ADST bit is 0.

29.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): S12AD.ADANSB0 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSB0 selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSB0n bit selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB007 bit corresponds to AN007.

When performing A/D conversion of the internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB0n bit should be set while the ADCSR.ADST bit is 0.

29.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD.ADANSB1 0008 9016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSB101	ANSB100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB100	A/D Conversion Channel Select	0: AN016 and AN017 are not subjected to conversion. 1: AN016 and AN017 are subjected to conversion.	R/W
b1	ANSB101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSB1 selects analog input channels for A/D conversion among AN016 and AN017 in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

ANSB1n Bit (n = 00, 01) (A/D Conversion Channel Select)

The ANSB1n bit selects analog input channels for A/D conversion among AN016 and AN017 in group B when group scan mode is selected. The ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB100 bit corresponds to AN016 and the ANSB101 bit corresponds to AN017.

When performing A/D conversion of the internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB1n bit should be set while the ADCSR.ADST bit is 0.

29.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0)

Address(es): S12AD.ADADS0 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN007 is not selected. 1: A/D-converted value addition/average mode for AN000 to AN007 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADADS0 selects the channels 0 to 7 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA0.ANSA0n bit (n = 00 to 07) or ADCSR.DBLANS[4:0] bits and ADANSB0.ANSB0n bit (n = 00 to 07) is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS0n bit should be set while the ADCSR.ADST bit is 0.

29.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1)

Address(es): S12AD.ADADS1 0008 900Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADS101	ADS100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS100	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN016 and AN017 is not selected. 1: A/D-converted value addition/average mode for AN016 and AN017 is selected.	R/W
b1	ADS101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADADS1 selects the channels 16 and 17 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS1n Bit (n = 00, 01) (A/D-Converted Value Addition/Average Channel Select)

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA1.ANSA1n bit (n = 00, 01) or ADCSR.DBLANS[4:0] bits and ADANSB1.ANSB1n bit (n = 00, 01) is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS1n bit should be set while the ADCSR.ADST bit is 0.

Figure 29.2 shows a scanning operation sequence in which both the ADS002 and ADS006 bits are set to 1.
It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011b), and the channels AN000 to AN007 are selected (ADANSA0.ANSA0n = FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively four times (addition three times), and the added (integrated) value is stored in A/D data register 2. After that the AN003 conversion is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is stored in A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

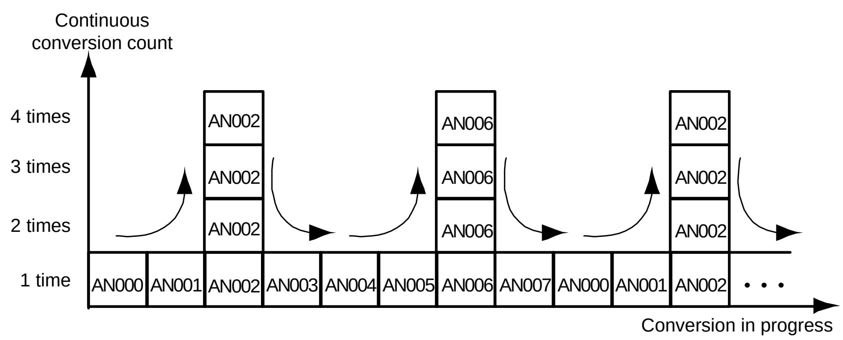


Figure 29.2 Scan Conversion Sequence with ADADC.ADC[2:0] = 011b, ADS002 = 1, and ADS006 = 1

29.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADC[2:0]	Addition Count Select	b2 b0 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Settings other than above are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

ADADC sets the addition count for A/D conversion of the channel and internal reference voltage for which A/D-converted value addition/average mode is selected, and selects either addition or average mode.

ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of internal reference voltage.

When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).

The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits) and internal reference voltage.

When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.

The AVEE bit should be set while the ADCSR.ADST bit is 0.

29.2.11 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 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	A/D Data Register 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]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited in self-diagnosis voltage fixed mode 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the voltage of reference power supply × 1/2 for self-diagnosis. 1 1: Uses the voltage of reference power supply 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: Flush-right is selected for the A/D data register format. 1: Flush-left is selected for the A/D data register format.	R/W

ADCER sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all "0") of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, or ADOCDR after any of these registers have been read by the CPU and DTC. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation

starts at the fixed voltage value.
The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A and B.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, or ADDBLDRB.
The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, see section section 29.2.1, A/D Data Registers y (ADDRy) A/D Data Duplication Register (ADDBLDR) A/D Data Duplication Register A (ADDBLDRA) A/D Data Duplication Register B (ADDBLDRB) A/D Internal Reference Voltage Data Register (ADOCDR), and section 29.2.2, A/D Self-Diagnosis Data Register (ADRD).

29.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSB[5:0]	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, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	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, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSTRGR selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.
When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh.
Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.
When the trigger from the module operated in 40 MHz (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 29.3.5, Analog Input Sampling Time and Scan Conversion Time for details.
Table 29.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, software trigger and asynchronous trigger cannot be used.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. When the trigger from the module operated in 40 MHz (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 29.3.5, Analog Input Sampling Time and Scan Conversion Time for details.

Table 29.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 29.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselection state			1	1	1	1	1	1
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA0 and TMR0.TCNT0 (unit0.ch0)	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA0 and TMR2.TCNT0 (unit1.ch0)	0	1	1	1	1	0
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0

Table 29.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			1	1	1	1	1	1
External pin	ADTRG0#	Input pin for the trigger	0	0	0	0	0	0
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA0 and TMR0.TCNT0 (unit0.ch0)	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA0 and TMR2.TCNT0 (unit1.ch0)	0	1	1	1	1	0
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0

29.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD.ADEXICR 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCSA	—	—	—	—	—	—	—	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/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OCSA	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

ADEXICR specifies the settings of A/D conversion of the internal reference voltage.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage in single scan mode. When A/D conversion of the internal reference voltage is to be performed, set all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE bit should be set to all 0 in single scan mode.
The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the internal reference voltage, so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

29.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, O)

Address(es): S12AD.ADSSTRL 0008 90DDh, S12AD.ADSSTRO 0008 90DFh,
S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h,
S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h,
S12AD.ADSSTR6 0008 90E6h, S12AD.ADSSTR7 0008 90E7h

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

The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 40 MHz, one state is 25 ns. The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLK to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.

Table 29.8 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 29.3.5, Analog Input Sampling Time and Scan Conversion Time.

Table 29.8 Relationship between A/D Sampling State Register and Relevant Channels

Register Name	Channels
ADSSTR0	AN000
ADSSTR1	AN001
ADSSTR2	AN002
ADSSTR3	AN003
ADSSTR4	AN004
ADSSTR5	AN005
ADSSTR6	AN006
ADSSTR7	AN007
ADSSTRL	AN016, AN017
ADSSTRO	Internal reference voltage*1

Note 1. When performing A/D conversion of the internal reference voltage, the sampling time should be 5 μs or longer.

29.2.15 A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address(es): S12AD.ADSHCR 0008 9066h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting	Set the sampling time (4 to 255 states).	R/W
b10 to b8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. 0: Bypass the channel-dedicated sample-and-hold circuits. 1: Use the channel-dedicated sample-and-hold circuits.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSHCR sets the parameters related to channel-dedicated sample-and-hold circuits.

SSTSH[7:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

These bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 40 MHz, one state is 25 ns. The initial value is 26 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The SSTSH[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and is 255 or less. Also, the sampling time should be 0.4 µs or longer.
For example, when ADCLK is set to 40 MHz, the lower-limit of the sampling state setting value is 16 states.

SHANS[2:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

These bits select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. The SHANS[0] bit selects AN000, SHANS[1] bit selects AN001, and SHANS[2] bit selects AN002. The SHANS[2:0] bits should be set while the ADCSR.ADST bit is 0.
If any channel from among AN000 to AN002 is selected for group B while operation is in group scan mode under group-A priority control, make the setting to bypass the channel-dedicated sample-and-hold circuit.

29.2.16 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 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]	A/D Disconnection Detection Assist Setting	b4 ADNDIS[4]: Discharge/precharge selected 0: Discharge 1: Precharge b3 to b0 ADNDIS[3:0]: Discharge/precharge period	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 (A/D Disconnection Detection Assist Setting)

These bits selects either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge. The ADNDIS[4:0] bits should be set when the ADCSR.ADST bit is 0.
When ADNDIS[3:0] are set to any value other than 0000b and the disconnection detection assist function is enabled, the channel-dedicated disconnection detection assist function is also enabled. Be sure to secure the wait time for the sample and hod circuit when using the channel-dedicated disconnection detection assist function.
When the ADEXICR.OCSA bit is set to 1 to perform A/D conversion of the internal reference voltage, ADNDIS[4:0] are automatically fixed to 0Fh, and discharging is executed prior to A/D conversion (auto-discharging). An auto-discharge period of 15 ADCLK cycles is inserted before sampling each time the internal reference voltage is A/D-converted.

29.2.17 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group-A Priority Control Setting *1	0: Operation is without group-A priority control 1: Operation is with group-A priority control	R/W
b1	GBRSCN	Group B Restart Setting *2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for group B is not restarted after having been discontinued due to group-A priority control. 1: Scanning for group B is restarted after having been discontinued due to group-A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	GBRP	Group B Single Scan Continuous Start *3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan for group B is not continuously activated. 1: Single scan for group B is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.

Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLK to A/D conversion clock ADCLK should be set to 1:1.

Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

ADGSPCR is used to make settings for priority control of A/D conversion for group A in group scan mode.

PGS Bit (Group-A Priority Control Setting)

This bit sets the priority of operation on group A. Set this bit to 1 when giving priority to operation on group A.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
When setting the PGS bit to 0, clearing should be performed by software according to section 29.7.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 29.3.4.3, Operation under Group-A Priority Control.

GBRSCN Bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.
If a scan operation on group B has been stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

GBRP Bit (Group B Single Scan Continuous Start)

This bit is set when a single scan operation is to be performed continuously on group B.
Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B has been stopped due to an operation on group A that takes priority, single scan on group B is automatically restarted on completion of the A/D conversion on group A.
Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the

GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

29.2.18 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT)

Address(es): S12AD.ADHVREFCNT 0008 908Ah

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

Bit	Symbol	Bit Name	Description	R/W
b0	HVSEL	High-Potential Reference Voltage Select	0: AVCC0 is selected as the high-potential reference voltage. 1: VREFH0 is selected as the high-potential reference voltage.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	LVSEL	Low-Potential Reference Voltage Select	0: AVSS0 is selected as the low-potential reference voltage. 1: VREFL0 is selected as the low-potential reference voltage.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

HVSEL Bits (High-Potential Reference Voltage Select)

These bits are used to set the high-potential reference voltage. AVCC0 or VREFH0 is selectable as the high-potential reference voltage.

LVSEL Bit (Low-Potential Reference Voltage Select)

This bit is used to set the low-potential reference voltage. AVSS0 or VREFL0 is selectable as the low-potential reference voltage.

29.3 Operation

29.3.1 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 ADCSR.ADST bit 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 synchronous trigger.

In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, 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 ADANSA0, ADANSA1, ADANSB0, and ADANSB1 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.

Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

Extended double trigger mode indicates a state when 0Bh (TRG4AN or TRG4BN) is selected by the TRSA[5:0] bits in the A/D conversion start trigger select register (ADSTRGR) in double trigger mode.

In extended double trigger mode, in addition to normal operations in double trigger mode, A/D conversion data is stored in A/D data duplication register A (ADDBLDRA) or A/D data duplication register B (ADDBLDRB) depending on the trigger type. If two types of triggers (TRG4AN or TRG4BN) have occurred simultaneously in this mode, A/D conversion data is not sorted by the trigger sources and is stored in data duplication register B (ADDBLDRB).

Note that if a new trigger is input during A/D conversion caused by another trigger, the new trigger is ignored.

When any of AN000 to AN002 channels is set as a channel-dedicated sample-and-hold circuit by the ADSHCR.SHANS[2:0] bits, the target analog input is sampled and held before the first A/D conversion of each scan.

The ADST0 output is used to output the ADCSR.ADST bit state.

29.3.2 Single Scan Mode

29.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, 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 S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI 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.

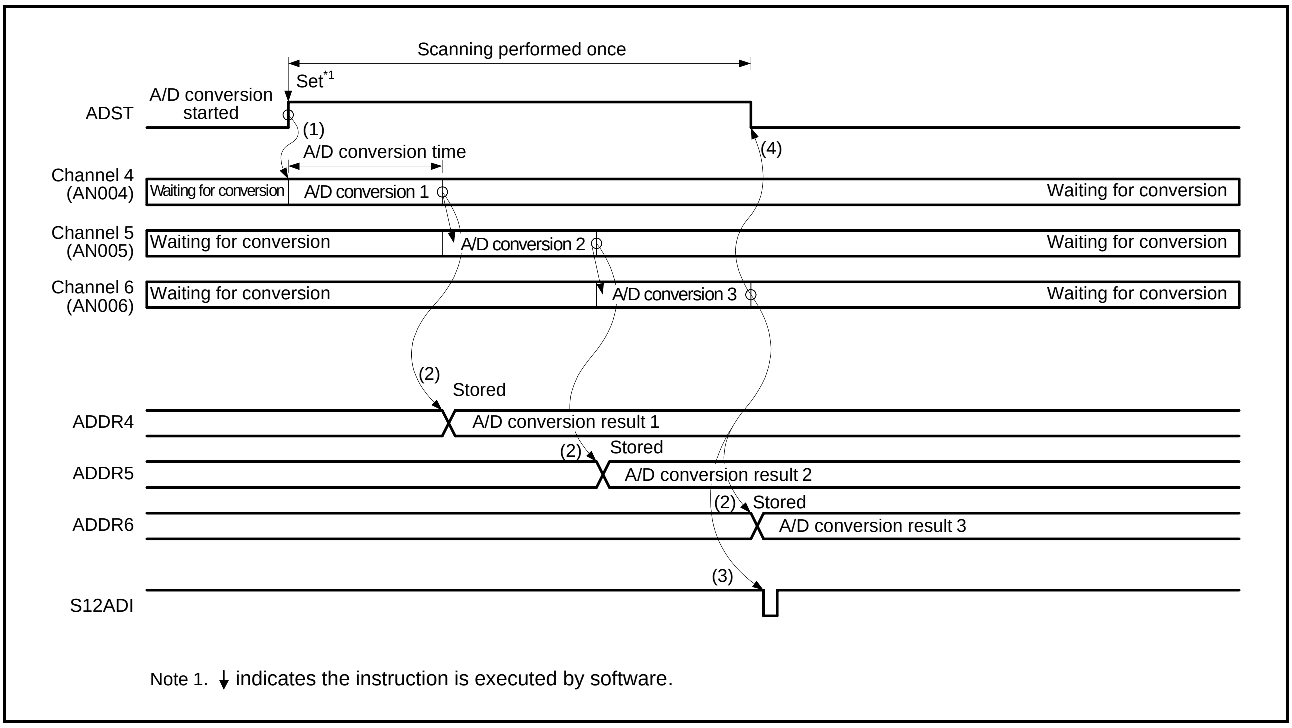


Figure 29.3 Example of Operation in Single Scan Mode (Basic Operation: AN004, AN005, AN006 Selected)

29.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.
In selected channel scanning, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI 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 waiting state.

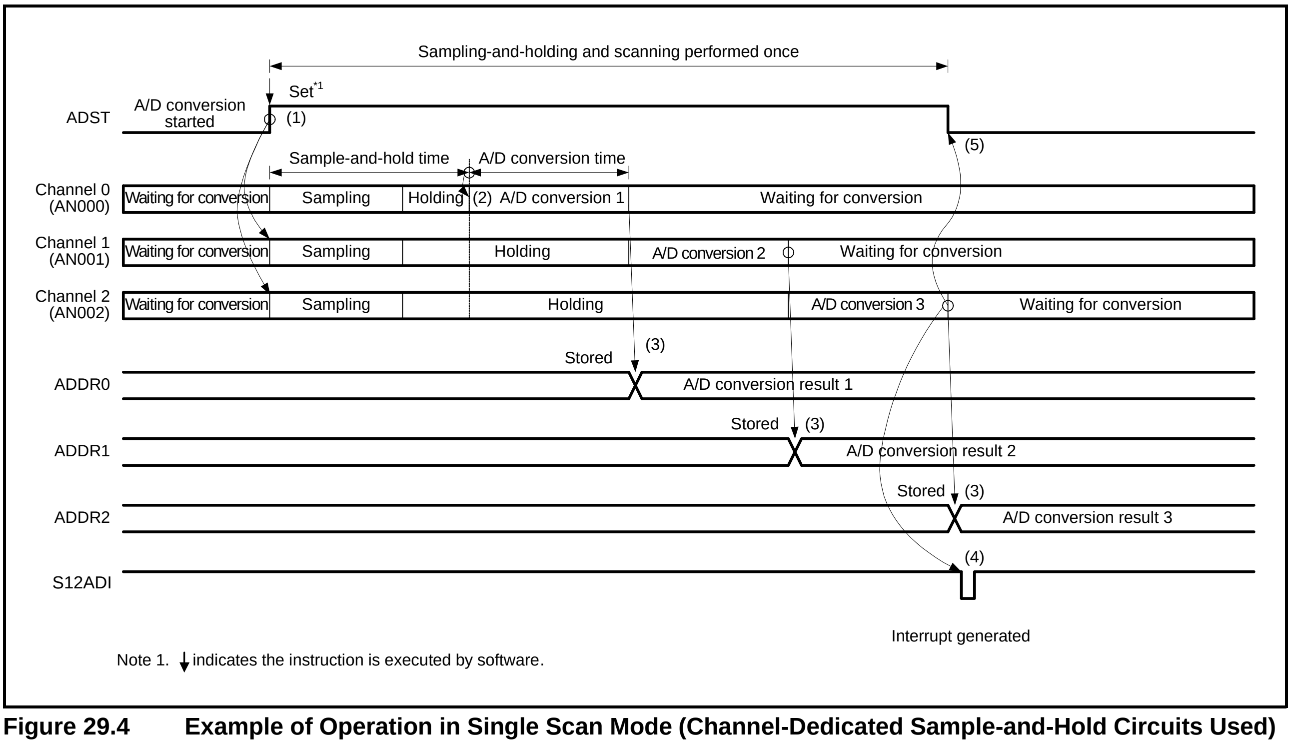


Figure 29.4 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)

29.3.2.3 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage VREFH0 supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, 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 S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI 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.

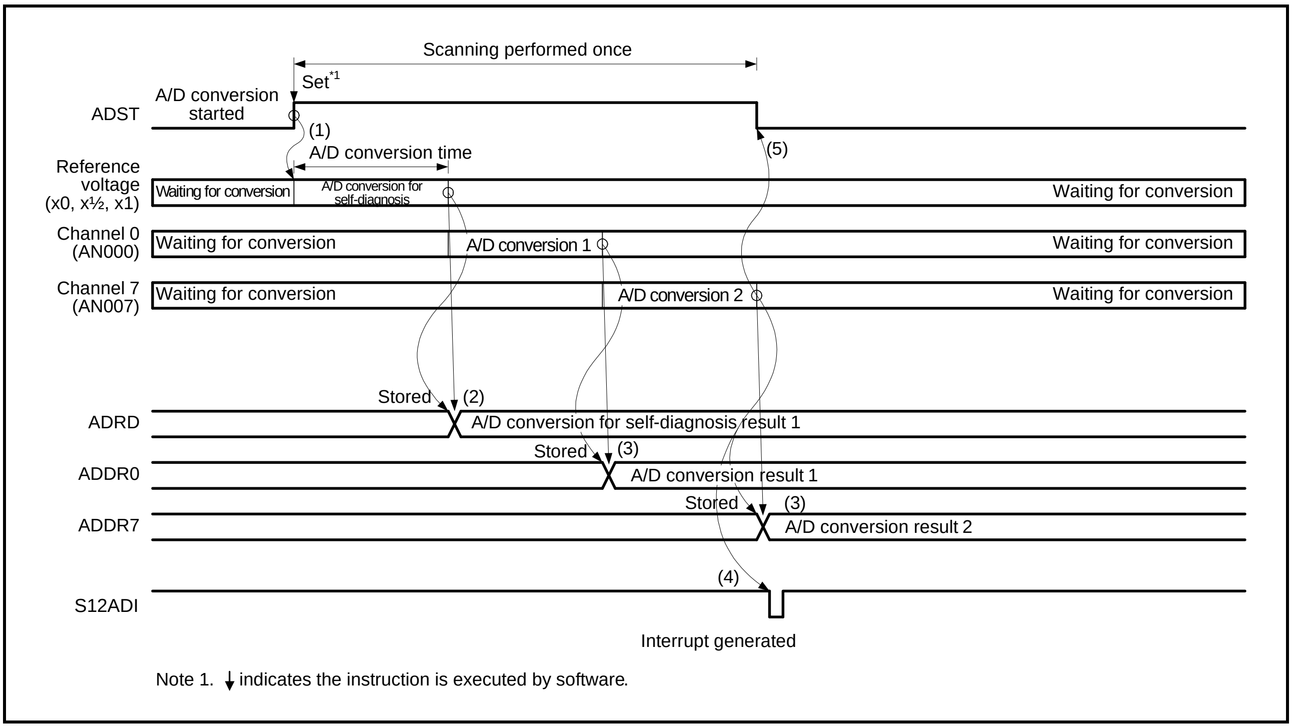


Figure 29.5 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN007 Selected + Self-Diagnosis)

29.3.2.4 A/D Conversion of Internal Reference Voltage

A/D conversion of the internal reference voltage is performed in single scan mode as below.
All channels should be deselected (by setting the ADANSA0 and ADANSA01 register bits to all 0 and the ADCSR.DBLE bit to 0).

(1) Set the sampling time to 5 µs or longer.
(2) After switching to A/D conversion of the internal reference voltage, start A/D conversion by setting the ADST bit to 1.
(3) When A/D conversion is completed, the conversion result is stored into the A/D internal reference voltage data register (ADOCDR). If the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled), an S12ADI interrupt request is generated.
(4) The ADST bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

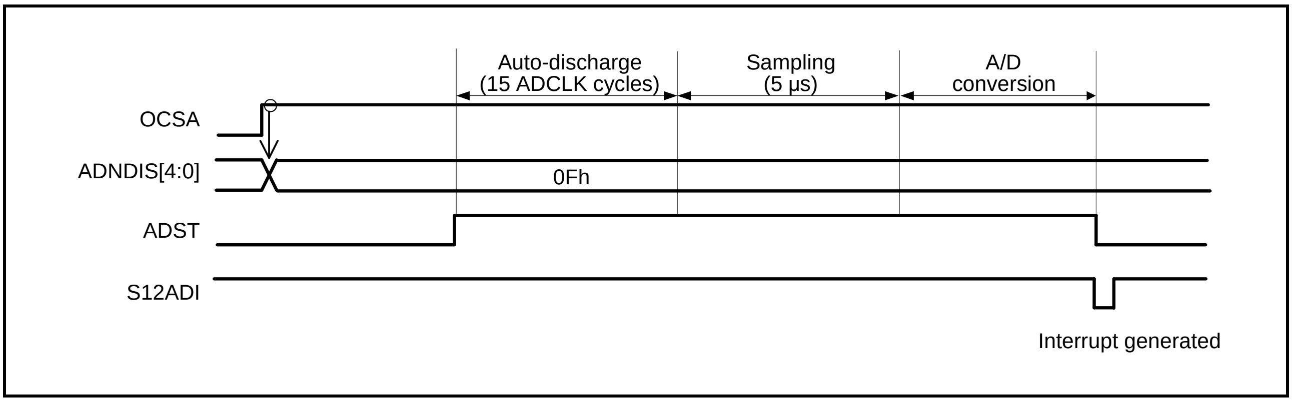


Figure 29.6 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)

29.3.2.5 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

Self-diagnosis should be deselected, and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[5:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.

(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (S12ADI interrupt upon scanning completion enabled).

(4) When the ADCSR.ADST bit 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 ADCSR.DBLANS[4:0] bits.

(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.

(6) If the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled), an S12ADI 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.

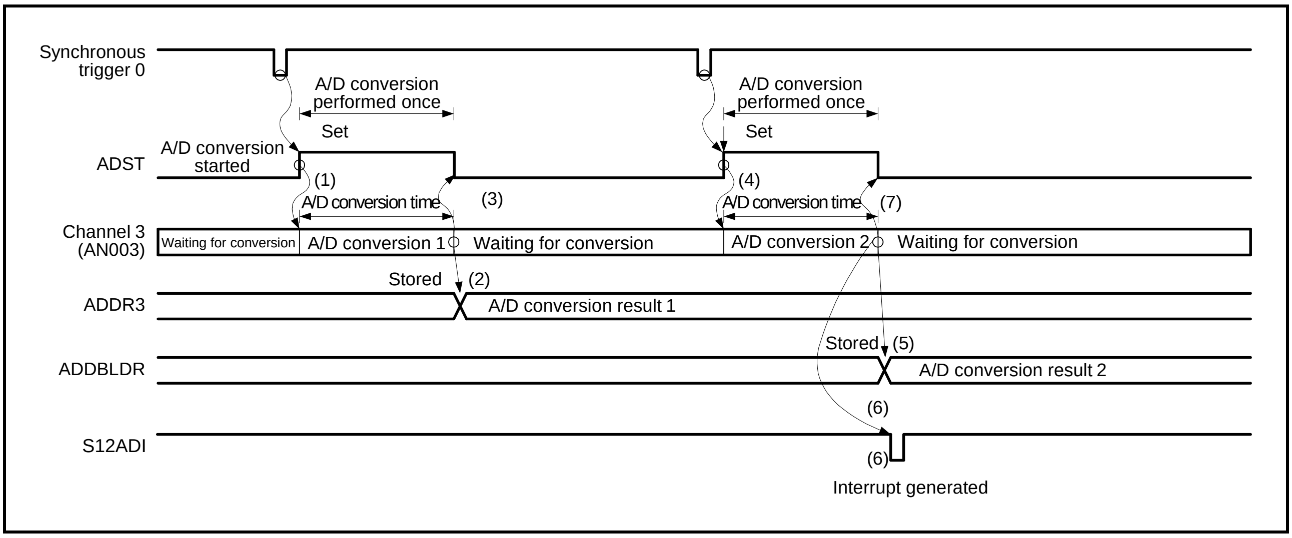


Figure 29.7 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

29.3.2.6 A/D Conversion in Extended Double Trigger Mode

When double trigger mode is selected in single scan mode, and 0Bh (TRG4AN or TRG4B) is selected by the TRSA[5:0] bits in the A/D conversion start trigger select register (ADSTRGR), single scan operation are performed twice as below. Self-diagnosis should be deselected, and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In extended double trigger mode, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4AN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) and A/D data-duplication register A (ADDBLDRA).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (S12ADI interrupt upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4BN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into A/D data-duplication register A (ADDBLDRA) and A/D data-duplication register B (ADDBLDRB).
(6) If the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled), an S12ADI 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.

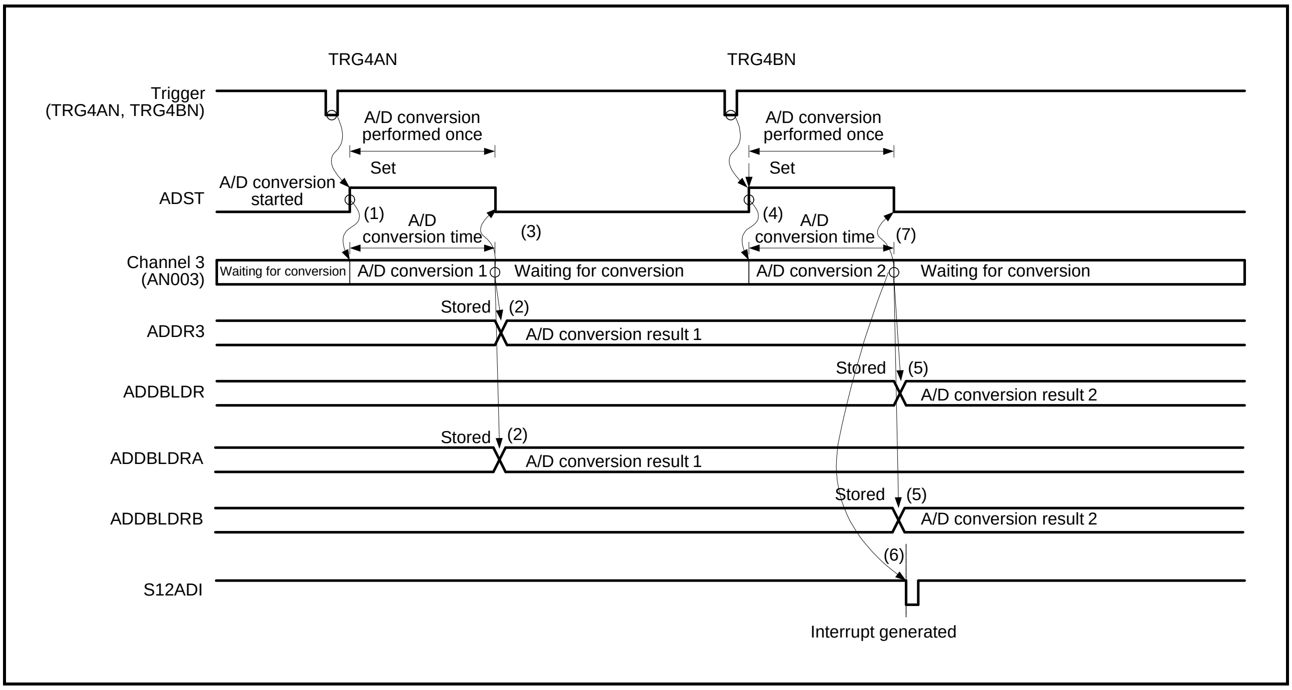


Figure 29.8 Example of Operation in Single Scan Mode (AN003 Duplicated; Extended Double Trigger Mode Operation by TRG4AN or TRG4BN)

29.3.3 Continuous Scan Mode

29.3.3.1 Basic Operation (Without Channel-Dedicated Sample and-Hold Circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.
In continuous scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, 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 S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

(4) The ADCSR.ADST bit 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 ADCSR.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.

(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 ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

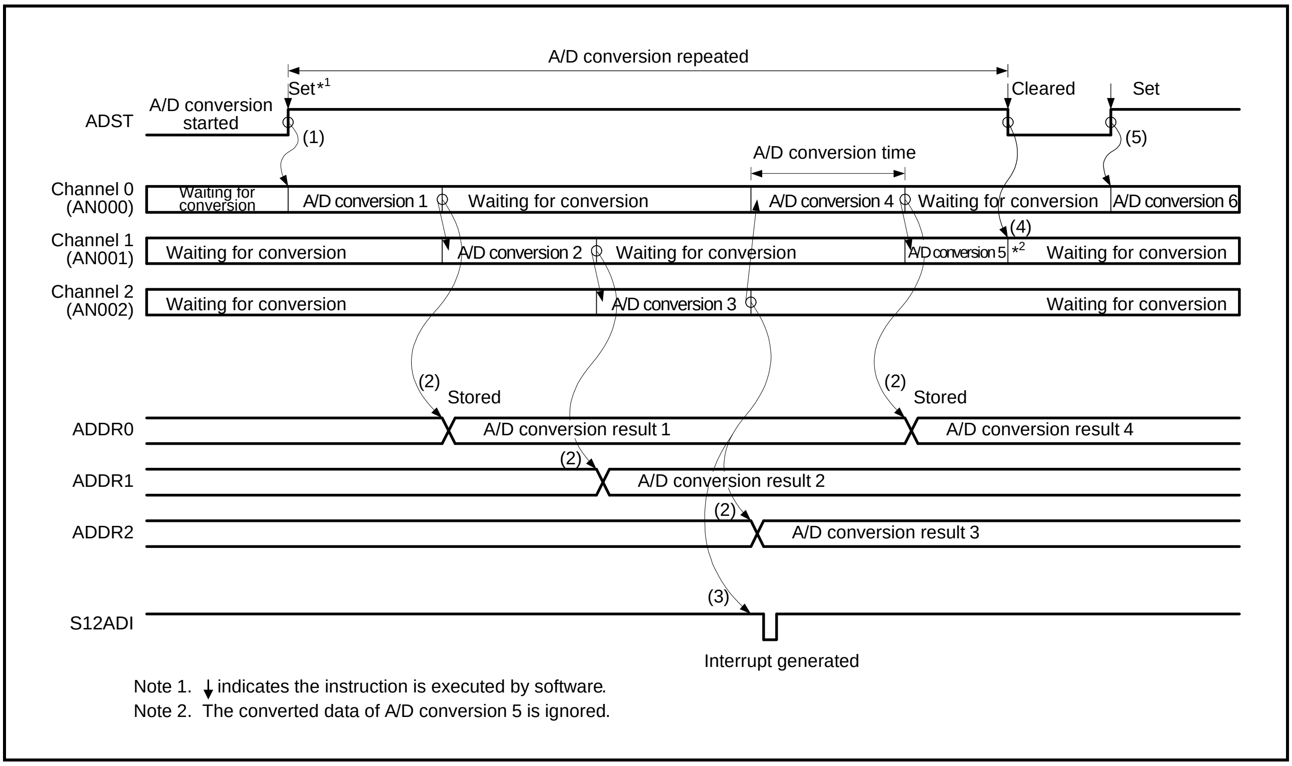


Figure 29.9 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

29.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, after which the analog inputs on all selected channels are A/D converted as below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:0] bits.
In continuous scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0, ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

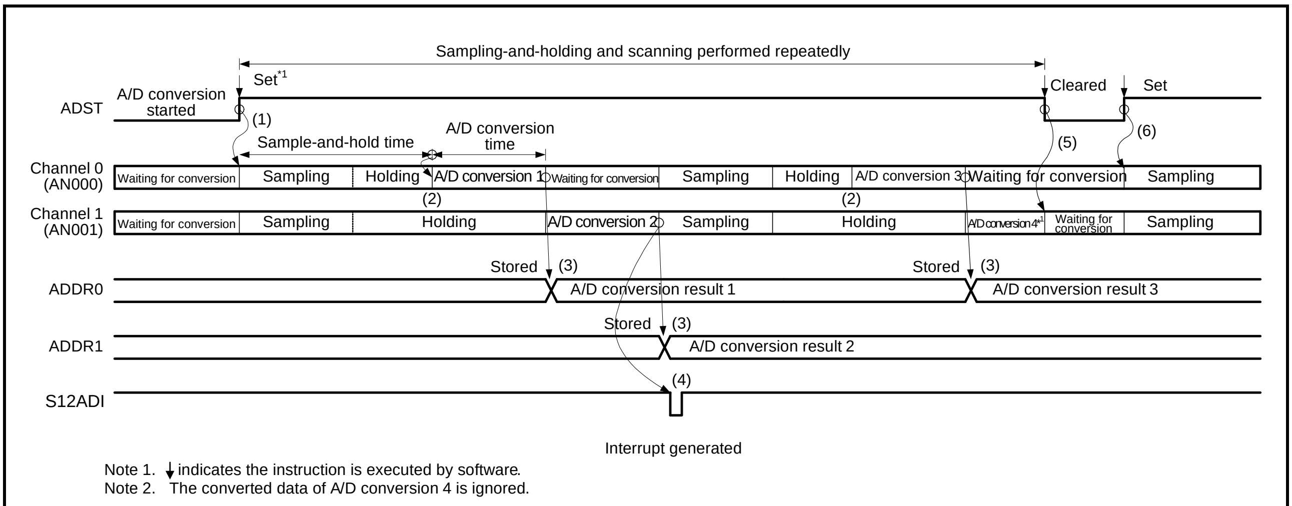


Figure 29.10 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)

29.3.3.3 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference 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, which sequence is repeated as below. In continuous scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger, or asynchronous trigger input, A/D conversion for self-diagnosis is started first.

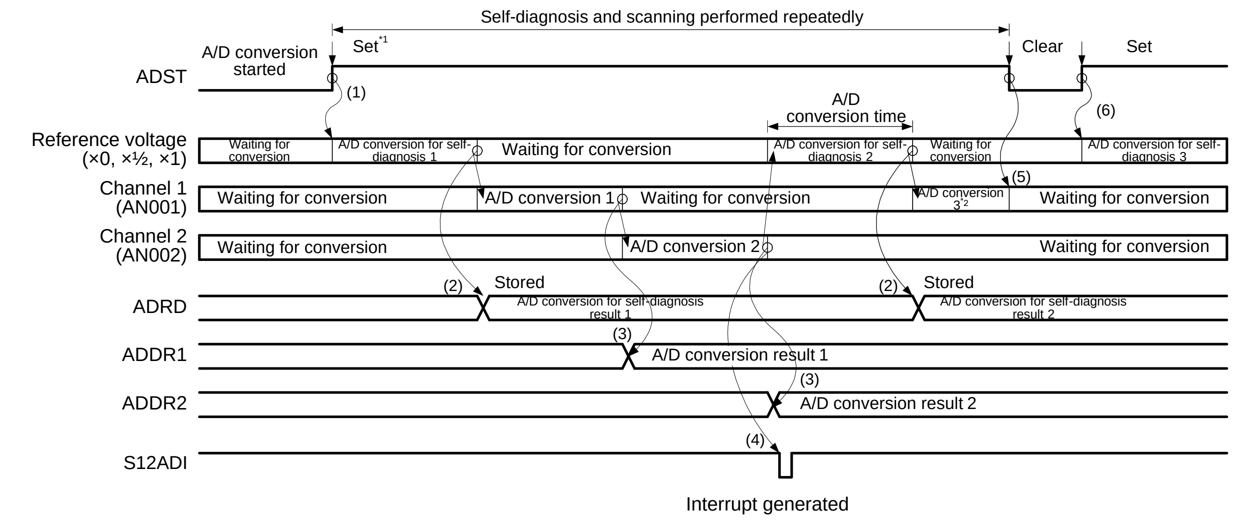
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

(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 S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI 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 ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

(5) The ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (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. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 29.11 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

29.3.4 Group Scan Mode

29.3.4.1 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 a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] 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 channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. The same channels cannot be selected for both groups.

In group scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

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. 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 S12ADI interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion 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 generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).

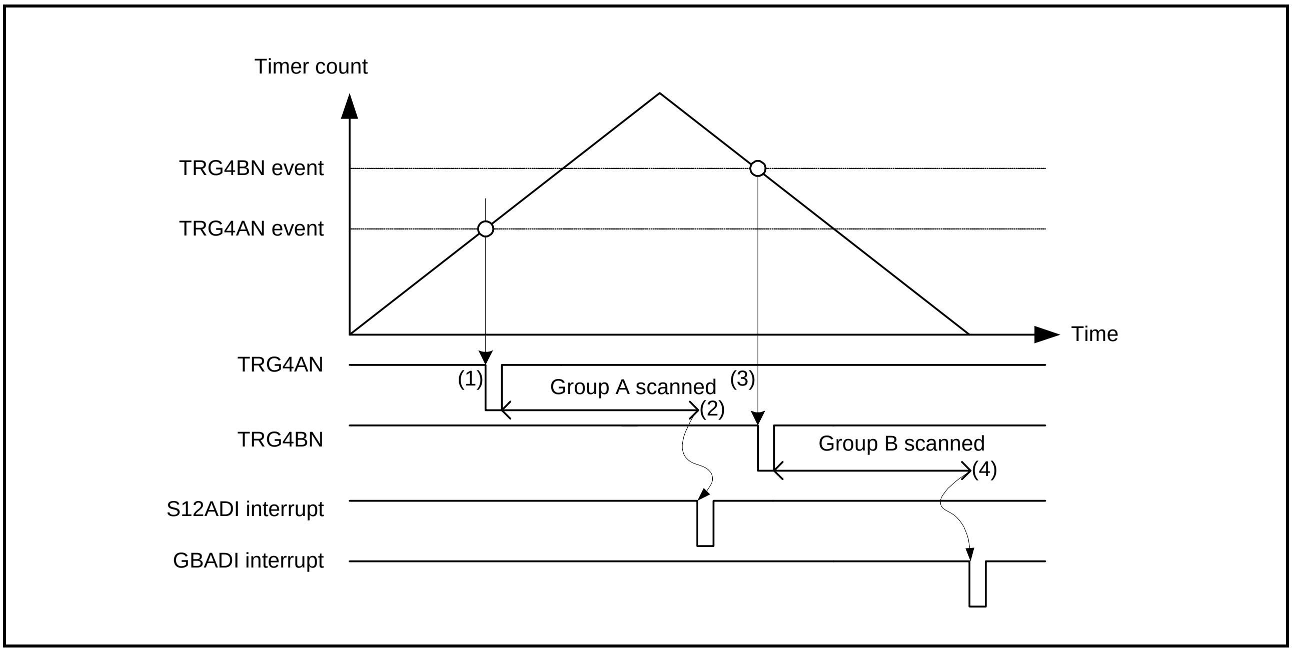


Figure 29.12 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

29.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger is performed once.

In group scan mode, the synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] 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 and asynchronous trigger should not be used. When a synchronous trigger (TRG4AN or TRG4BN) is selected (the ADSTRGR.TRSA[5:0] bits are set to 0Bh), operation is performed in extended double trigger mode.

The group A and group B channels to be A/D-converted are selected using the ADCSR.DBLANS[4:0] bits and the ADANSB0 and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.

In group scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following describes operation in group scan mode with double trigger mode using a synchronous trigger from the MTU. The TRG4ABN 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 generated if the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion 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 the corresponding A/D data register (ADDRy); an S12ADI 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 S12ADI interrupt is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).

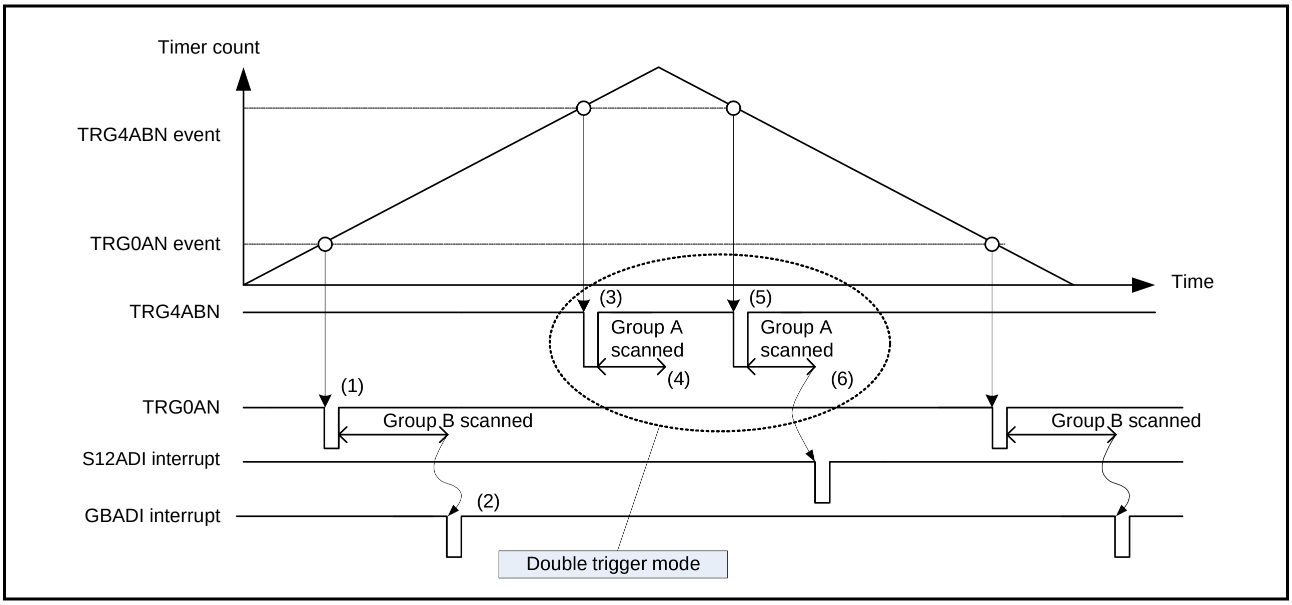


Figure 29.13 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

29.3.4.3 Operation under Group-A Priority Control

Setting the PGS bit in the A/D group scan priority control register (ADGSPCR) to 1 in group scan mode makes operation proceed under group-A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 29.14. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.
In operation in basic group scan mode, input of the trigger for the other group during operation for A/D conversion in group A or group B is ignored. Under group-A priority control, if a group-A trigger is input during A/D conversion for group B, A/D conversion for group B is discontinued and A/D conversion for group A proceeds. If the setting of the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for group B from the head of the group after completion of the A/D conversion for group A. Table 29.9 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

Scan operations in group A or group B are the same in single scan mode. Furthermore, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for group B.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits and select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1. Furthermore, as targets for A/D conversion, select channels for group A using the ADANSA0 and ADANSA1 registers, and for group B, select channels different from those for group A using the ADANSB0 and ADANSB1 registers.

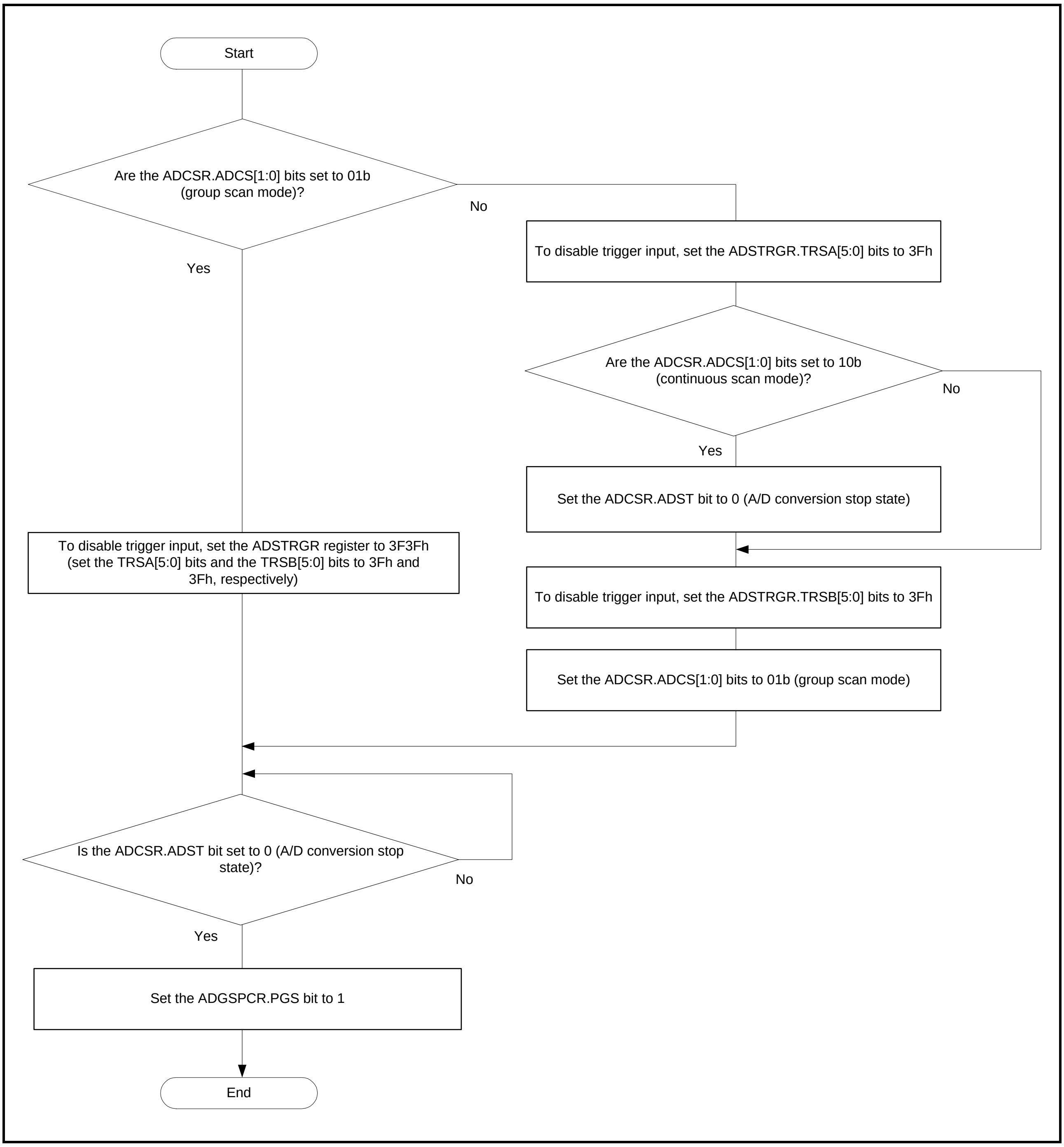


Figure 29.14 Flow of Setting the ADGSPCR.PGS Bit

Table 29.9 Control of A/D Conversion Operations According to the Settings of the ADGSPCR.GBRSCN Bit

A/D Conversion Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

The following describes the operations in group scan mode under group-A priority control (i.e. ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start), and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon group B scanning completion enabled).
(9) The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

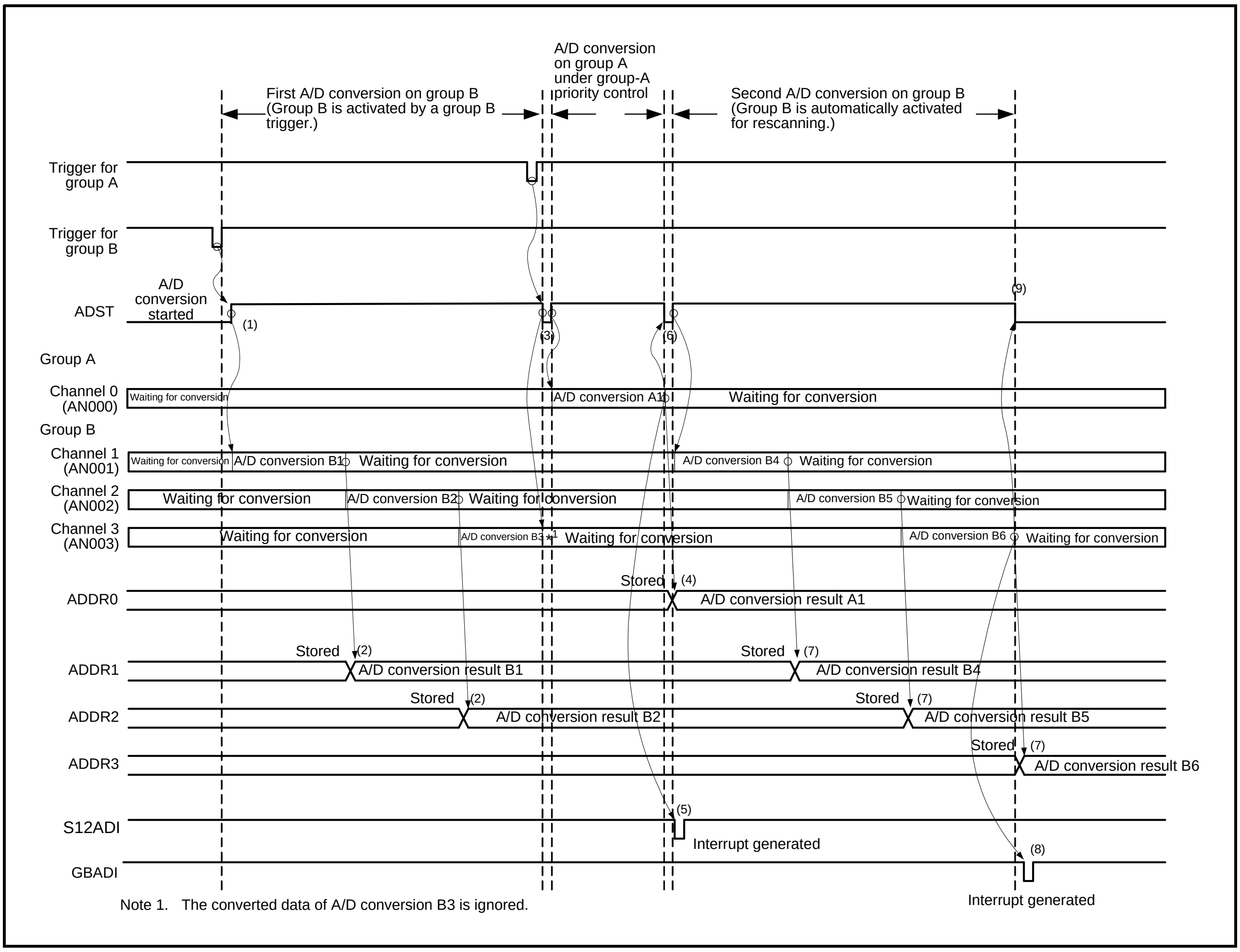


Figure 29.15 Example of Operations under Group-A Priority Control (1) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example when a group A trigger is input again during rescanning operation on group B. In this example, channel 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared to 0 (A/D conversion stop) on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued.
(4) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(6) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).

(7) On completion of A/D conversion on the group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(9) If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit is cleared to 0 (A/D conversion stop) and the ongoing A/D conversion on group B is stopped.

(10) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.

(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(12) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).

(13) On completion of A/D conversion on group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (rescanning operation enabled). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.

(14) If a group A trigger is input during A/D conversion on group B for rescanning, steps 9 to 13 are repeated. If a group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on group B and the 12-bit A/D converter enters a wait state.

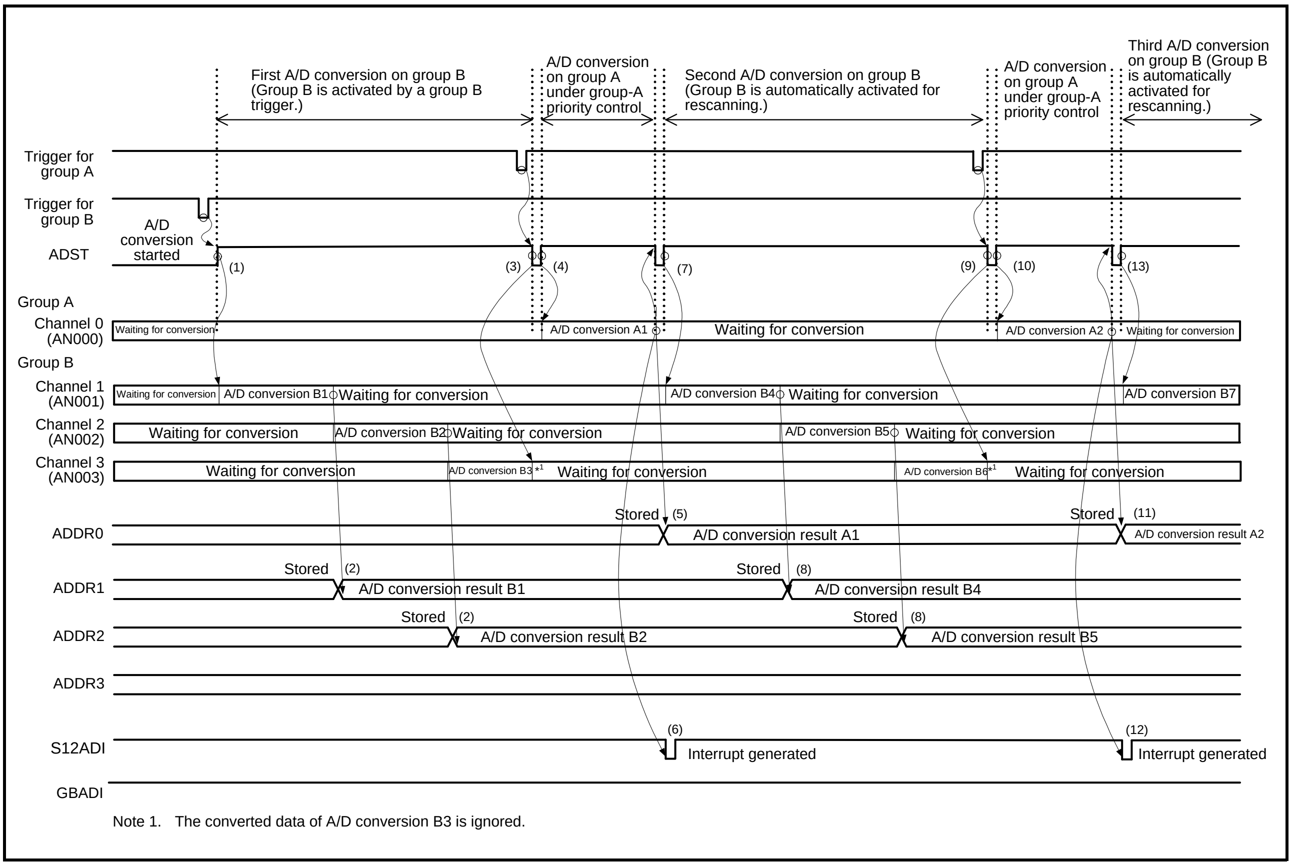


Figure 29.16 Example of Operations under Group-A Priority Control (2) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of a rescanning operation in which a group B trigger is input during A/D conversion on group A. In this example, channels 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group A sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group B trigger is input during A/D conversion on group A, A/D conversion on group B can be performed after the A/D conversion on group A is completed. (However, if group A triggers are input continuously, the scan operation on group B is canceled by group A and is not performed.)
(4) On completion of the A/D conversion on the group A, an S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(5) On completion of the A/D conversion on the group A, activation of group B for rescanning sets the ADCSR.ADST bit to 1 automatically.
After that, conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(7) On completion of the rescanning operation on the group B, a GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (GBADI interrupt upon scanning completion enabled).
(8) The ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

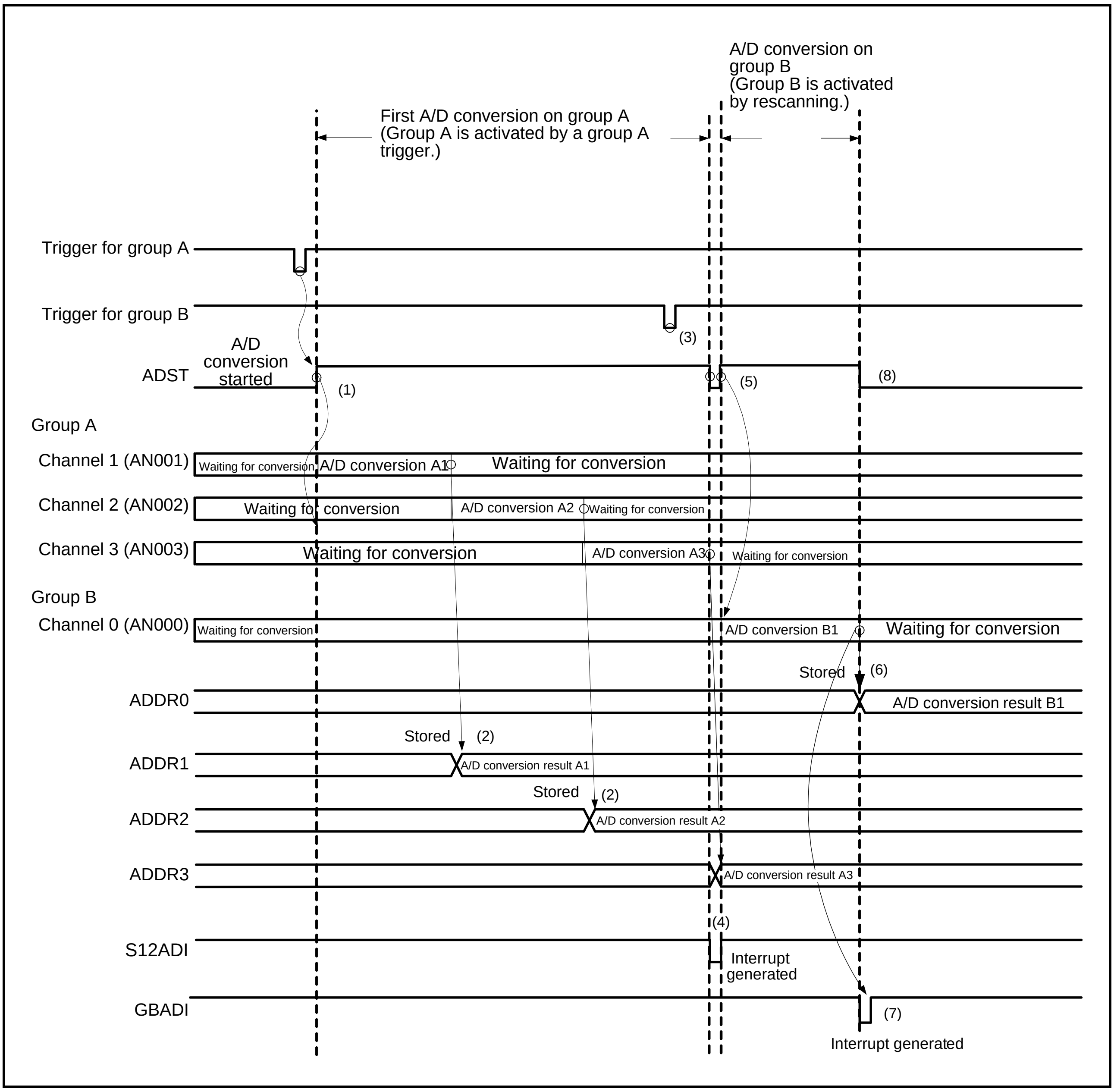


Figure 29.17 Example of Operations under Group-A Priority Control (3) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).

(6) The ADCSR.ADST bit retains the value 1 (A/D conversion start) during A/D conversion and is cleared on completion of conversion, after which the A/D converter enters a wait state.

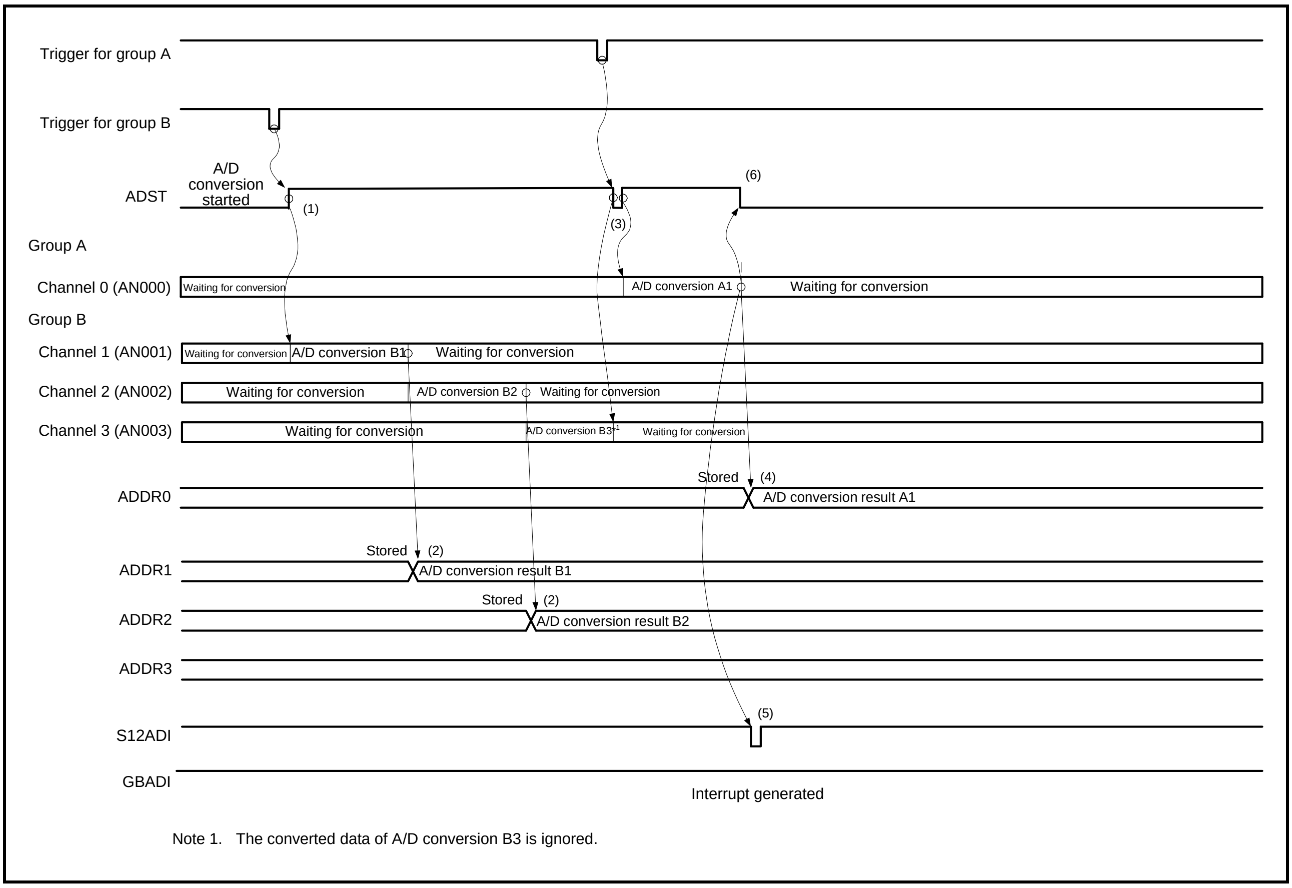


Figure 29.18 Example of Operation under Group-A Priority Control (4) (when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0)

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRP = 1).

(1) The ADCSR.ADST bit is set to 1 (A/D conversion start) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(6) After the ADST bit is automatically cleared, again, the ADCSR.ADST bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A GBADI interrupt request is generated if the setting of the ADCSR. GBADIE bit is 1.
(9) After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (A/D conversion start) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n. Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop A/D conversion when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 29.7.2, Notes on Stopping A/D Conversion.

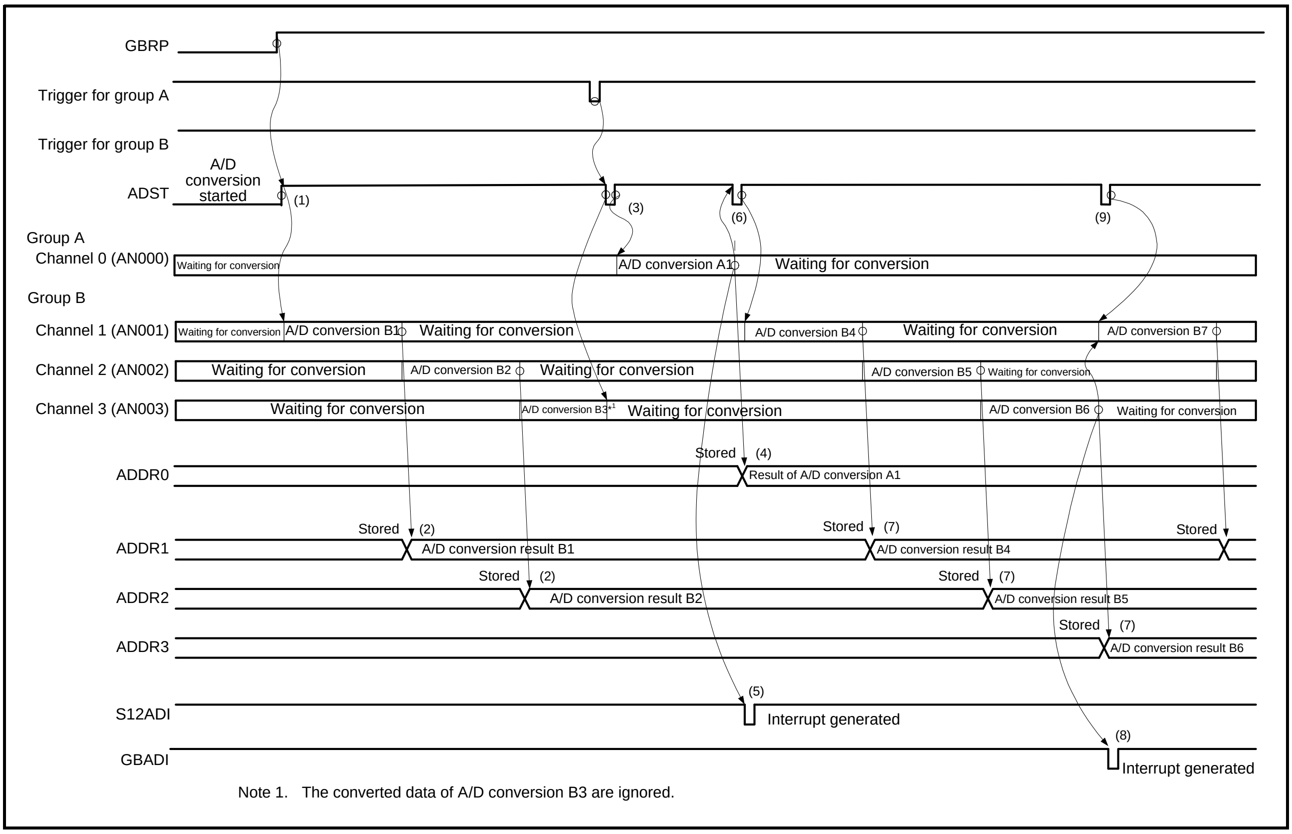


Figure 29.19 Example of Operation under Group-A Priority Control (5) (when ADGSPCR.GBRP = 1)

29.3.5 Analog Input Sampling Time and Scan Conversion Time

Scan conversion can be activated either by software, synchronous trigger, or asynchronous trigger input. After the start-of-scanning-delay time (t_D) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.

Figure 29.20 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software or a synchronous trigger. Figure 29.21 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH})*1, disconnection detection assistance processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states. Table 29.10 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*5} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}.

The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n) + t_{SHED}$.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.

Note 2. When disconnection detection assistance is not selected, $t_{DIS} = 0$. The auto-discharge period of 15 ADCLK states is inserted only when internal reference voltage is A/D-converted.

Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

Note 4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).

Note 5. $t_{CONV} \times n$ when the sampling time (t_{SPL}) of selected channels is the same, but it is the total of the sampling time of each channel and time for conversion by successive approximation (t_{SAM}).

Table 29.10 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK)

Item			Symbol		Type/Conditions				Unit
					Synchronous Trigger (MTU)	Synchronous Trigger (TMR)	Asynchronous Trigger	Software Trigger	
Scan start processing time*1, *2	A/D conversion on group A under group-A priority control.	Group B is to be stopped. (Group A is activated after group B is stopped due to an A/D conversion source of group A.)	t_D		1PCLKA + 4PCLKB + 6ADCLK	3PCLKB + 6ADCLK	—	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source of group A.)			1PCLKA + 3PCLKB + 4ADCLK	2PCLKB + 4ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			1PCLKA + 3PCLKB + 6ADCLK	2PCLKB + 6ADCLK	4PCLKB+ 6ADCLK	6ADCLK	
	Other than above				1PCLKA + 3PCLKB + 4ADCLK	2PCLKB + 4ADCLK	2PCLKB + 4ADCLK	4ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	The setting of ADSHCR.SSTSH[7:0] (initial value = 1Ah) × ADCLK				
	Wait time between sampling and A/D conversion			t_W	13 ADCLK				
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*3				
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Dh) × ADCLK				
	Time for conversion by successive approximation			t_{SAM}	32 ADCLK				
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.		t_{DED}		2 ADCLK				
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.		t_{DSD}		2 ADCLK				
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 7, L, O) (initial value = 0Dh) × ADCLK				
	Time for conversion by successive approximation			t_{SAM}	32 ADCLK				
Channel-dedicated sample-and-hold end processing time			t_{SHED}		3 ADCLK				
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK				

Note 1. For t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}, see Figure 29.20 and Figure 29.21.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The value is fixed to 0Fh (15 ADCLK) when the internal reference voltage is A/D-converted.

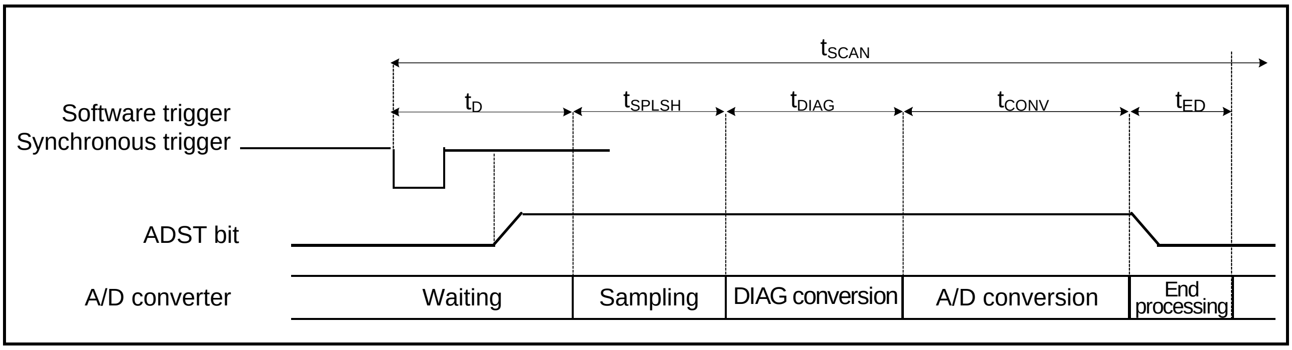


Figure 29.20 Scan Conversion Timing (Activated by Software or Synchronous Trigger)

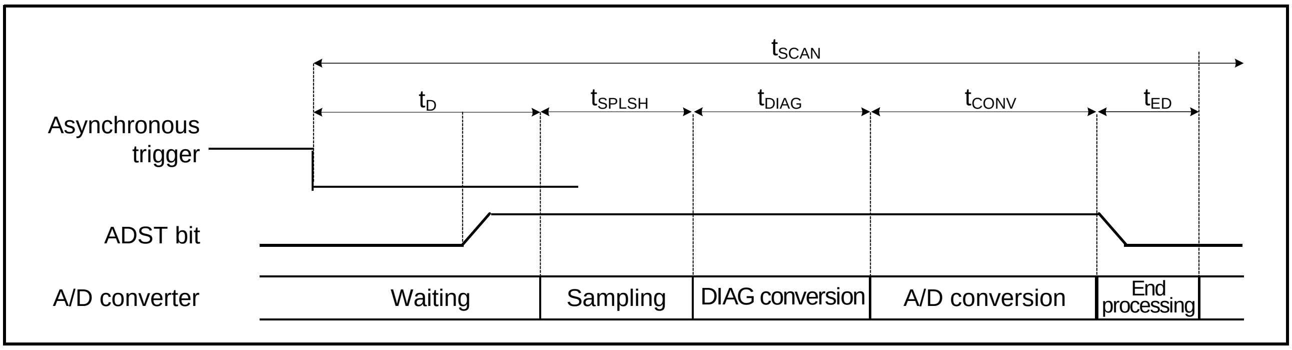


Figure 29.21 Scan Conversion Timing (Activated by Asynchronous Trigger)

29.3.6 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB) to 0000h when the A/D data registers are read by the CPU or DTC.
This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.
In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU or DTC, 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.

29.3.7 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition/average mode can be specified when A/D conversion of the channel select analog input or internal reference voltage is selected.

29.3.8 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state (reference voltage selected by the A/D high-potential/low-potential reference voltage control register) before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.
Figure 29.22 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 29.23 shows an example of disconnection detection when precharge is selected. Figure 29.24 shows an example of disconnection detection when discharge is selected.

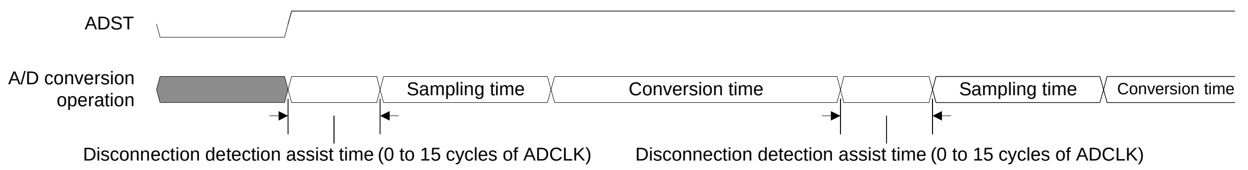
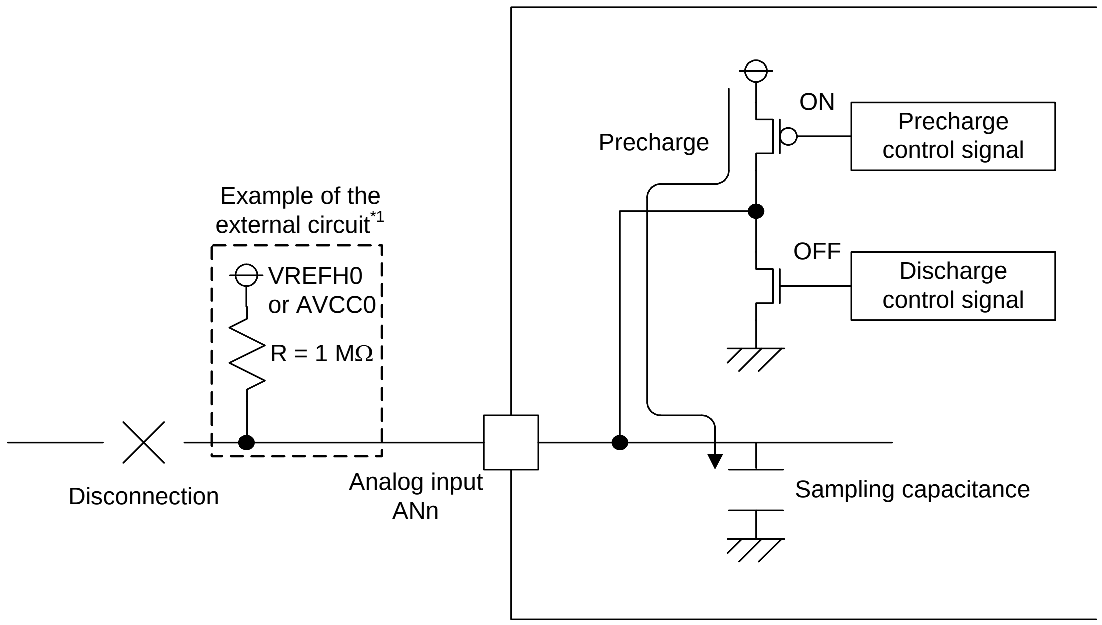
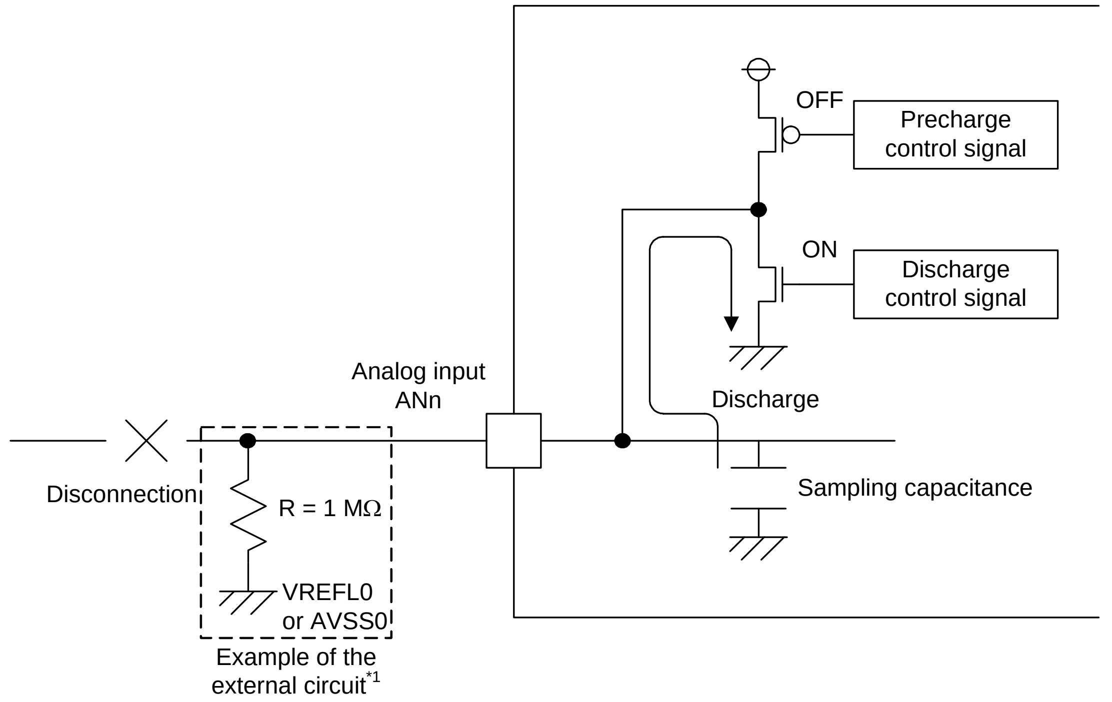


Figure 29.22 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 29.23 Example of Disconnection Detection When Precharge is Selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 29.24 Example of Disconnection Detection When Discharge is Selected

29.3.9 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[5:0]) should be set to 000000b, and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin). Then, the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 29.25 shows a timing of the asynchronous trigger input.
For the time from when the ADST bit is set to 1 until conversion starts, refer to section 29.7.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B used in group scan mode.

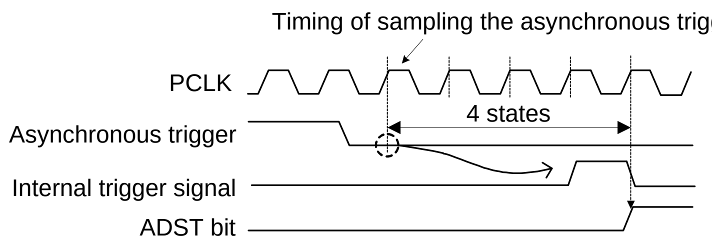


Figure 29.25 Timing of Sampling Asynchronous Trigger

29.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

29.4 Interrupt Sources and DTC Transfer Requests

29.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI and GBADI to the CPU.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively.
In addition, the DTC can be activated when an S12ADI or a GBADI interrupt is generated. Using an S12ADI or a GBADI interrupt to allow the DTC to read the converted data enables continuous conversion without burden on software.
For details on DTC settings, see section 17, Data Transfer Controller (DTCa).

29.5 Selecting Reference Voltage

In 64-pin package products, the high-potential reference voltage can be selected from VREFH0 and AVCC0 and the low-potential reference voltage can be selected from VREFL0 and AVSS0, respectively. Set these before starting A/D conversion. In 52-pin and 48-pin package products, only AVCC0 can be used as the high-potential reference voltage, and only AVSS0 can be used as the low-potential reference voltage. For details of this setting, see section 29.2.18, A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT).

29.6 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.0 μs, the analog input pins of this MCU 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 2.5 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. 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 29.26 shows an equivalent circuit of an analog input pin and an external sensor.

To perform A/D conversion accurately, charging of the internal capacitor C shown in Figure 29.26 must be completed within the specified period of time. This specified period is referred to as sampling time.

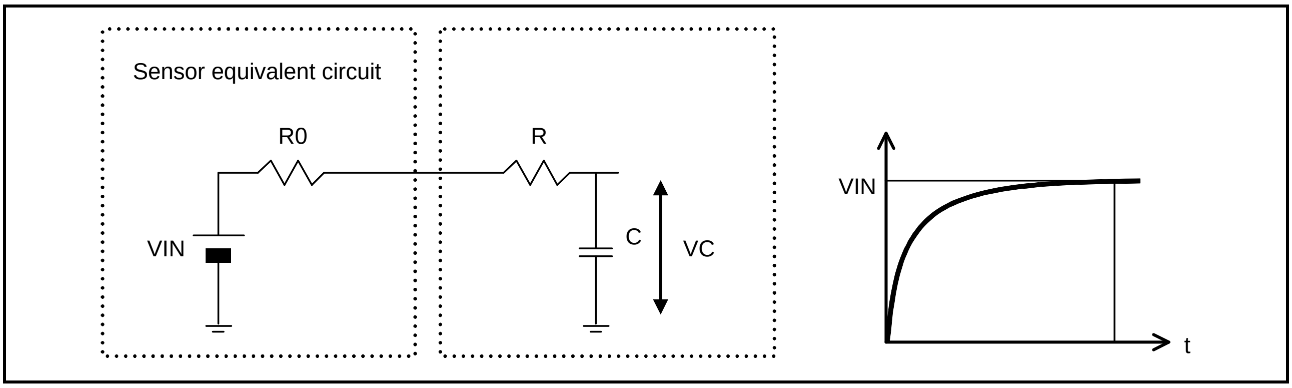


Figure 29.26 Equivalent Circuit of Analog Input Pin and External Sensor

29.7 Usage Notes

29.7.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B A/D 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 higher-order byte and lower-order byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in byte units.

29.7.2 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, follow the procedure in Figure 29.27.

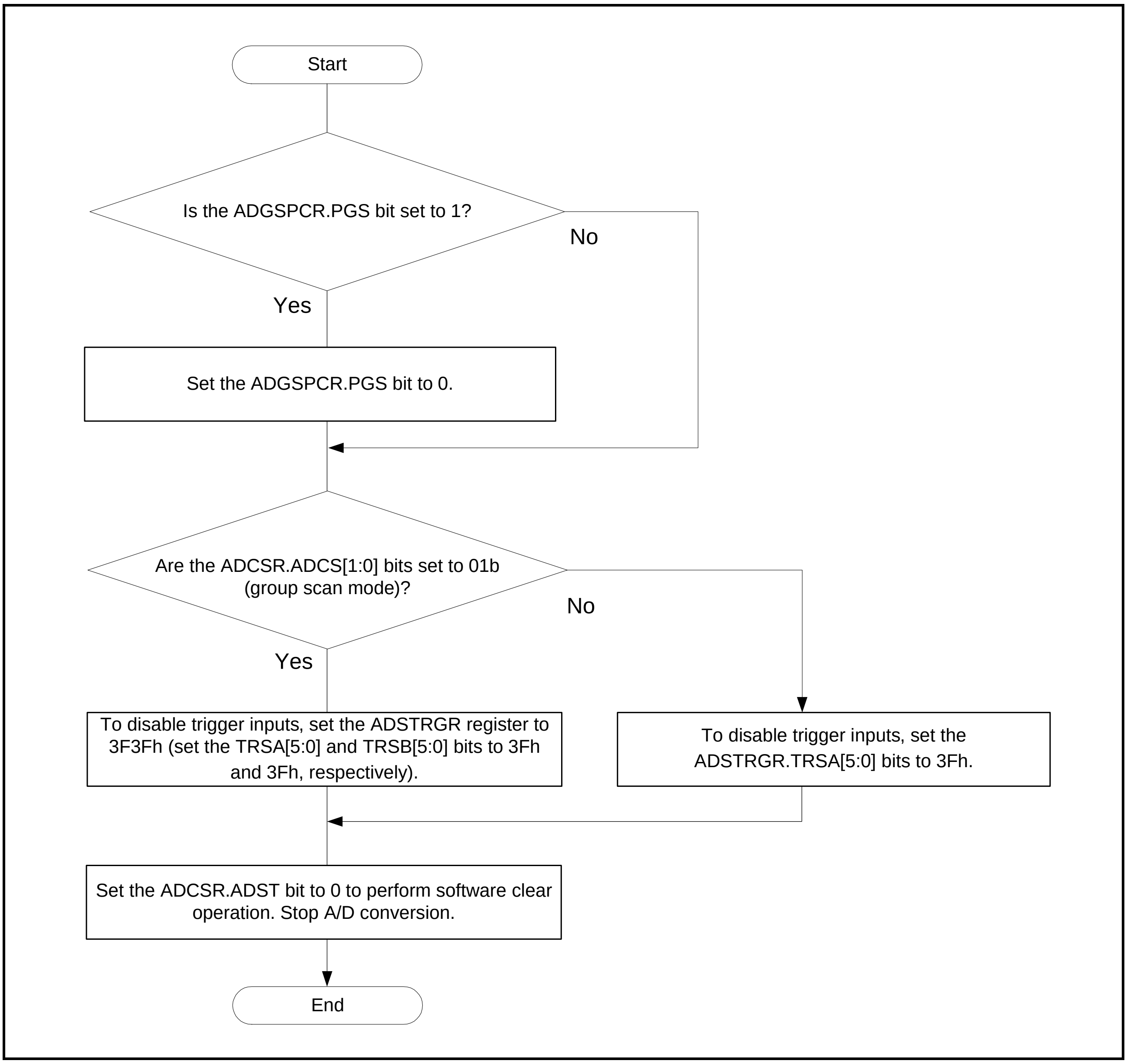


Figure 29.27 Procedure for Clear Operation by Software through the ADCSR.ADST Bit

29.7.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

29.7.4 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 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.

29.7.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

29.7.6 Notes on Entering Low Power Consumption States

Before entering the module stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in Figure 29.27. After that, wait for two clock cycles of ADCLK before entering the peripheral module stop state or software standby mode.

29.7.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

29.7.8 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = $4095 \times Rs / Rp$

29.7.9 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins ANn: AVSS0 ≤ VAN ≤ AVCC0
  Reference voltage range applied to pins VREFH0 and VREFL0: VREFH0 ≤ AVCC0, VREFL0 = AVSS0

Conversion will not succeed if the voltage applied to analog input pins ANn is greater than VREFH0 (see Figure 29.28).

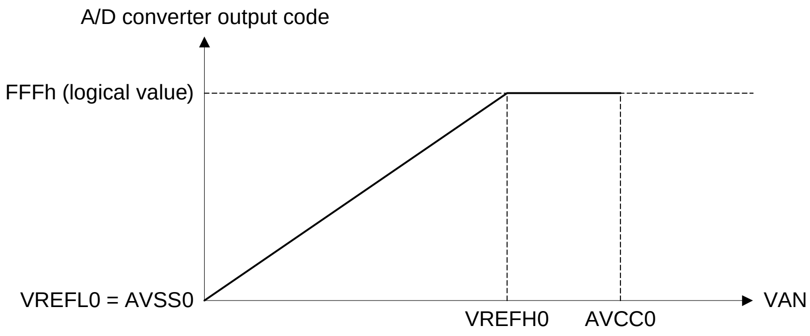


Figure 29.28 Relationship Between Voltage Applied to Analog Input Pins and Output Code

- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  The following condition should be satisfied: AVSS0 = VSS. When performing A/D conversion of analog input pin ANn (n = 016, 017), the following condition should be satisfied: AVCC0 = VCC. A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 29.29, and connection should be made so that the following conditions are satisfied at the supply side.
  VREFL0 = AVSS0 = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied.
  VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS

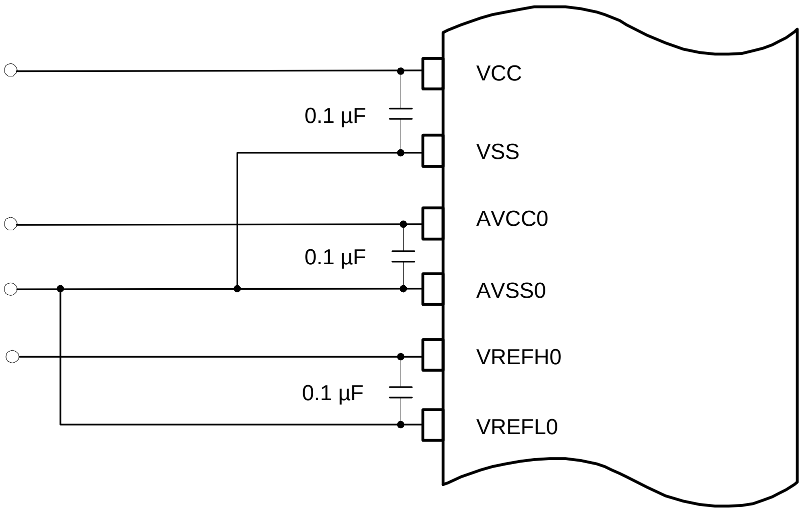


Figure 29.29 Power Supply Pin Connection Example

29.7.10 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 AN007, AN016, AN017), 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).

29.7.11 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN016, AN017) 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 AN007, AN016, AN017) as shown Figure 29.30.

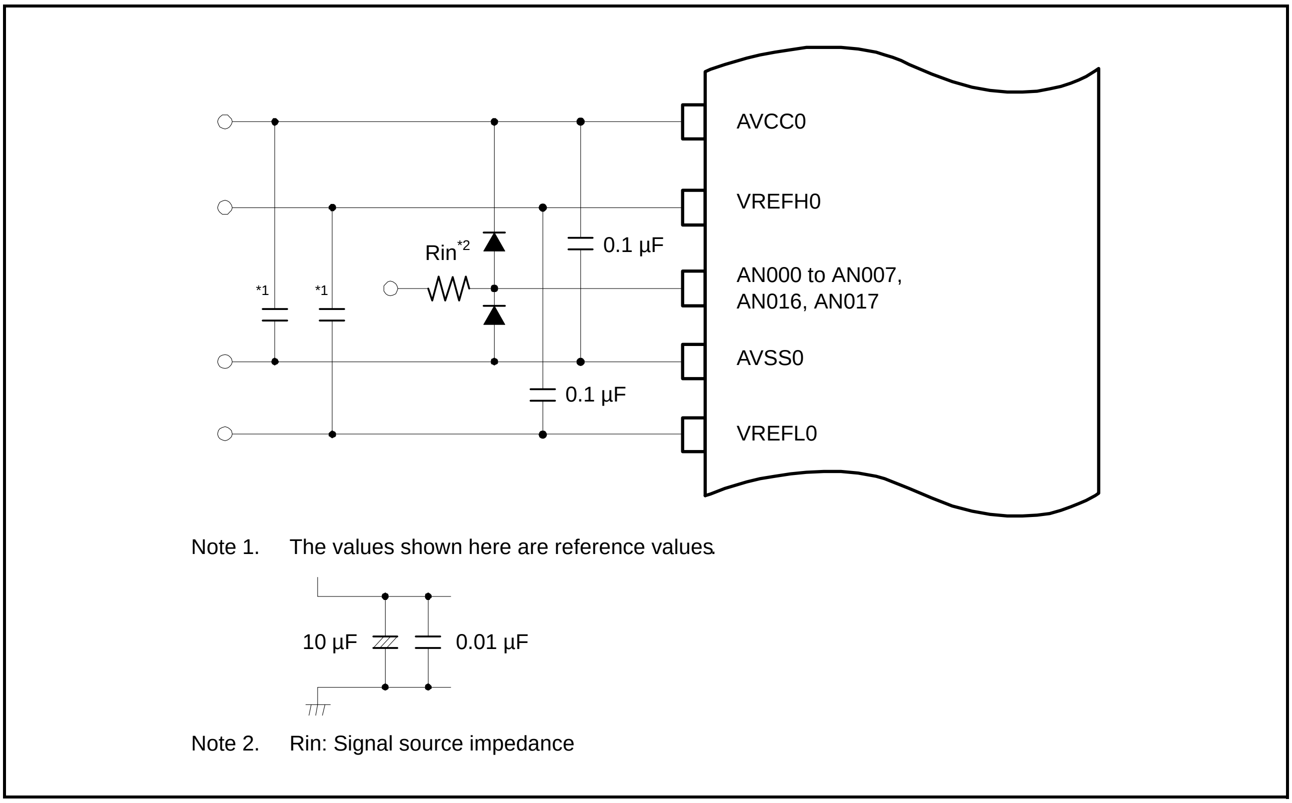


Figure 29.30 Sample Protection Circuit for Analog Inputs

30. D/A Converter for Generating Comparator C Reference Voltage (DA)

In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

This MCU includes one channel of 8-bit D/A converter. This D/A converter is used for generating comparator C reference voltage.

Table 30.1 lists the specifications of the 8-bit D/A converter and Figure 30.1 shows a block diagram of the 8-bit D/A converter.

Table 30.1 Specifications of 8-Bit D/A Converter

Item	Specifications
Resolution	8 bits
Output channels	One channel
Low power consumption function	Module stop state can be set.

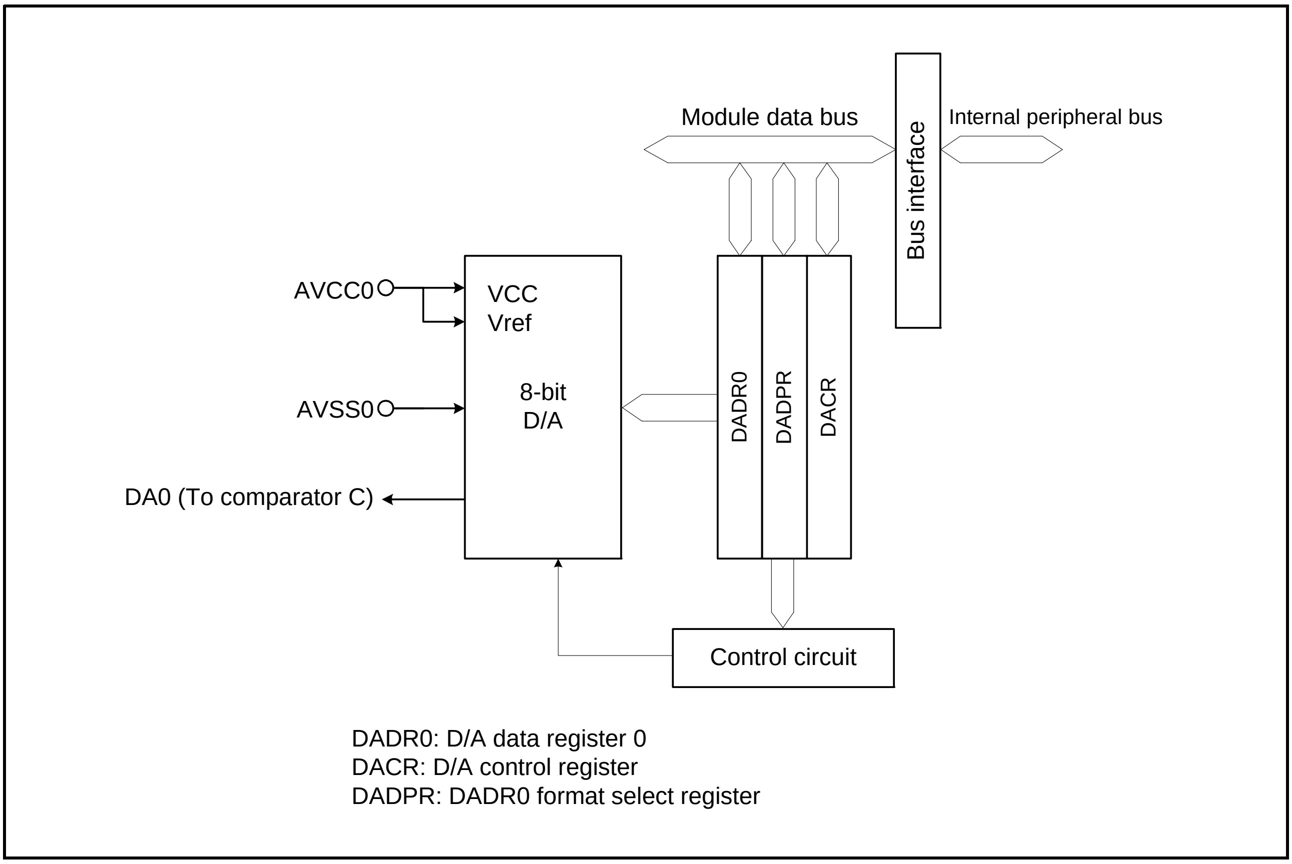


Figure 30.1 Block Diagram of 8-Bit D/A Converter

30.2 Register Descriptions

30.2.1 D/A Data Register 0 (DADR0)

Address(es): DA.DADR0 0008 80C0h

• DADPR.DPSEL bit = 0 (data is flush with the right end of the register)

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

• DADPR.DPSEL bit = 1 (data is flush with the left end of the register)

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

The DADR0 register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADR0 are converted. The reference voltage for comparator C is supplied.

8-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits “—” are read as 0. The write value should be 0.

30.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 80C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DAOE0	—	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	DAOE0	D/A Output Enable 0	0: D/A conversion is disabled. 1: D/A conversion is enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion.

30.2.3 DADR0 Format Select Register (DADPR)

Address(es): DA.DADPR 0008 80C5h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	DADR0 Format Select	0: Data is flush with the right end of the D/A data register. 1: Data is flush with the left end of the D/A data register.	R/W

30.3 Operation

When the DACR.DAOE0 bit is set to 1, D/A converter is enabled and the conversion result is output.
An operation example of D/A conversion is shown below. Figure 30.2 shows the timing of this operation.

1. Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output after the conversion time tDCONV has elapsed. The conversion result continues to be output until the DADR0 register is written to again or the DAOE0 bit is cleared to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Value of DADR0 register}}{256} \times \text{AVCC0}$$

3. If the DADR0 register is written to again, the conversion is started. The conversion result is output after the conversion time tDCONV has elapsed.
4. If the DAOE0 bit is set to 0, D/A conversion is disabled.

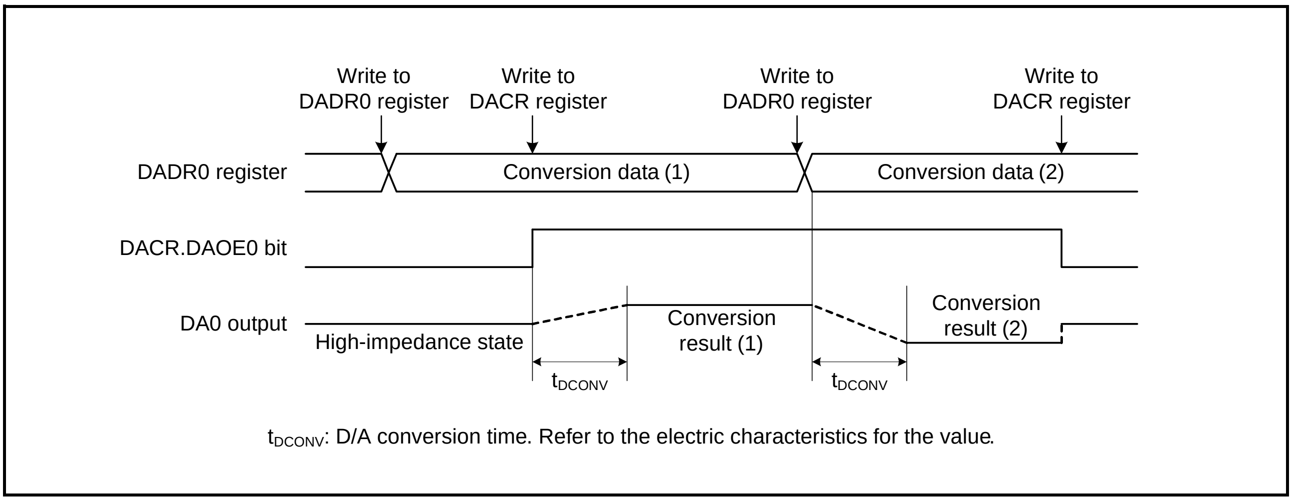


Figure 30.2 Example of 8-Bit D/A Converter Operation

30.4 Usage Notes

30.4.1 Module Stop Function Setting

Operation of the 8-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 8-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

30.4.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR. and DAOE0 bit to 0.

30.4.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR. and DAOE0 bit to 0.

30.4.4 Setting the D/A Converter

Set the D/A converter for generating comparator C reference voltage and wait for the D/A converter output settling time (t_{DCONV}) before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

31. Comparator C (CMPC)

31.1 Overview

Comparator C compares a reference input voltage to an analog input voltage.
The comparison result can be read by software and output externally, and an interrupt request can be generated upon any changes to the comparison result.
Comparator C reference input voltage can be selected from either an input to the CVREFC0/CVREFC1 pin or an output from on-chip D/A converter.
There are four analog inputs, one of which is to be selected.
Table 31.1 lists the specification of comparator C, Figure 31.1 shows a block diagram of comparator C, and Table 31.2 shows comparator C pin configuration.
In this section, "PCLK" is used to refer to PCLKB.

Table 31.1 Comparator C Specifications

Item	Specification
Number of channels	Three (comparator C0 to comparator C2)
Analog input voltages	• Input voltage to the CMPCnm pin (n = channel number; m = 0 to 2) • Internal reference voltage
Reference input voltage	Input voltage to the CVREFC0/CVREFC1 pin or on-chip D/A converter output voltage
Comparison result	The comparison result can be output externally.
Digital filter function	• One of three sampling periods can be selected. • The filter function can also be disabled. • A noise-filtered signal can be used to generate interrupt request output and POE source output, and comparison results can be read via registers.
Interrupt request	• An interrupt request is generated upon detecting a valid edge of the comparison result. • Rising edge, falling edge, or both edges of the comparison result can be selected.
Low power consumption function	Module stop state can be set.

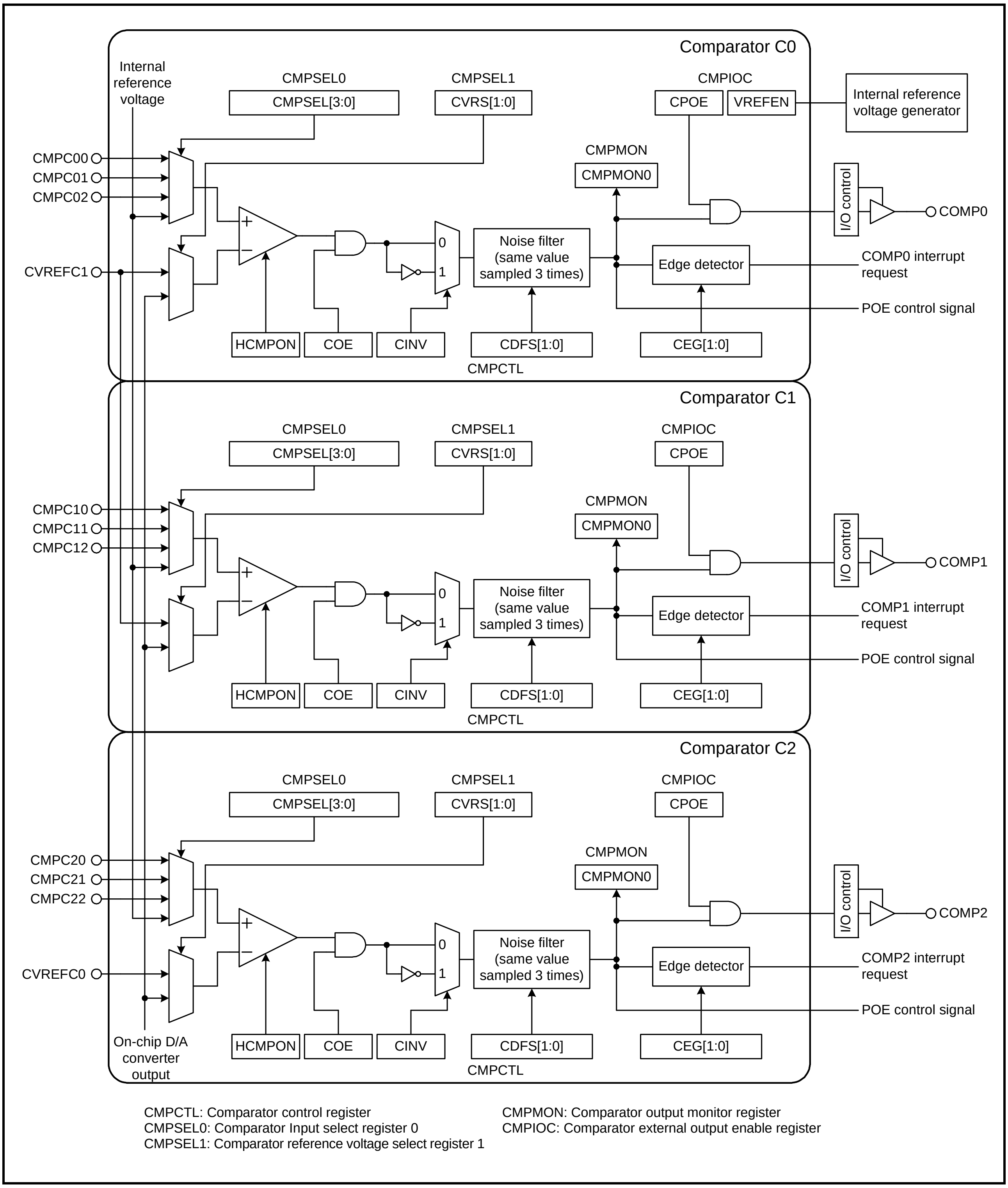


Figure 31.1 Comparator C Block Diagram

Table 31.2 Comparator C Pin Configuration

Pin Name	I/O	Function
CMPC00, CMPC01, CMPC02	Input	Comparator C0 analog input pins
CMPC10, CMPC11, CMPC12	Input	Comparator C1 analog input pins
CMPC20, CMPC21, CMPC22	Input	Comparator C2 analog input pins
CVREFC0	Input	Reference input voltage pin 0
CVREFC1	Input	Reference input voltage pin 1
COMP0	Output	Comparator C0 comparison result output pin
COMP1	Output	Comparator C1 comparison result output pin
COMP2	Output	Comparator C2 comparison result output pin

31.2 Register Descriptions

31.2.1 Comparator Control Register (CMPCTL)

Address(es): CMPC0.CMPCTL 000A 0C80h, CMPC1.CMPCTL 000A 0CA0h, CMPC2.CMPCTL 000A 0CC0h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CINV	Comparator Output Polarity Select*1, *4	0: Comparator output not inverted 1: Comparator output inverted	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is fixed to 0) 1: Comparator output enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG[1:0]	Comparator Edge Select	b4 b3 0 0: Interrupt request is not generated. 0 1: Rising edge 1 0: Falling edge 1 1: Both edges	R/W
b6, b5	CDFS[1:0]	Noise Filter Sampling Select*1, *2, *4	b6 b5 0 0: Noise filter not used 0 1: Sampling frequency is PCLK/8. 1 0: Sampling frequency is PCLK/16. 1 1: Sampling frequency is PCLK/32.	R/W
b7	HCMPON	Comparator Operation Enable*3	0: Operation stopped (the output signal is fixed to 0) 1: Operation enabled (input to the comparator pins is enabled)	R/W

Note 1. Rewrite the CDFS[1:0] and CINV bits only after disabling the comparator output (COE bit = 0).

Note 2. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), allow four sampling times to elapse until the filter output is updated, and then use the comparator interrupt request.

Note 3. The operation stabilization wait time is required after enabling comparator operation (HCMPON bit = 1).

Note 4. Rewriting the CINV and CDFS[1:0] bits may generate a comparator interrupt request and a POE source. Before changing these bits, set the registers in POE so that comparator output is not used for high-impedance control. After changing these bits, also set the corresponding interrupt status flag (IR) in the interrupt request register and the POE comparator channel n detection flag (n = 0 to 2) to 0.

CEG[1:0] Bits (Comparator Edge Select)

These bits select which edge of comparator output signal is used to generate an interrupt request.

The valid edge is set for the signal after the comparator polarity is selected by the CINV bit and the filter is selected by CDFS[1:0] bits.

31.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): CMPC0.CMPSEL0 000A 0C84h, CMPC1.CMPSEL0 000A 0CA4h, CMPC2.CMPSEL0 000A 0CC4h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPSEL[3:0]	Comparator Input Select*1	• Comparator C0 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC00 selected 0 0 1 0 : CMPC01 selected 0 1 0 0 : CMPC02 selected 1 0 0 0 : Internal reference voltage selected Settings other than above are prohibited. • Comparator C1 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC10 selected 0 0 1 0 : CMPC11 selected 0 1 0 0 : CMPC12 selected 1 0 0 0 : Internal reference voltage selected Settings other than above are prohibited. • Comparator C2 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC20 selected 0 0 1 0 : CMPC21 selected 0 1 0 0 : CMPC22 selected 1 0 0 0 : Internal reference voltage selected Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite the CMPSEL[3:0] bits in the following procedure. Writing a value other than 0000b while the value of these bits is not 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CMPSEL[3:0] bits to 0000b.
(3) Set a new value to the CMPSEL[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

31.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): CMPC0.CMPSEL1 000A 0C88h, CMPC1.CMPSEL1 000A 0CA8h, CMPC2.CMPSEL1 000A 0CC8h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CVRS[1:0]	Reference Input Voltage Select *1, *2	• Comparator C0 and comparator C1 b1 b0 0 0: No input 0 1: Input voltage to the CVREFC1 pin selected as reference input voltage 1 0: On-chip D/A converter output voltage selected as reference input voltage Settings other than above are prohibited. • Comparator C2 b1 b0 0 0: No input 0 1: Input voltage to the CVREFC0 pin selected as reference input voltage 1 0: On-chip D/A converter output voltage selected as reference input voltage Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the on-chip D/A converter output voltage is used, set the D/A converter for generating comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, refer to section 30, D/A Converter for Generating Comparator C Reference Voltage (DA).

Note 2. Rewrite the CVRS[1:0] bits in the following procedure. Be sure to set the CVRS[1:0] bits to 00b before changing the set value. Rewriting the value directly from 01b to 10b or 10b to 01b is not possible.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CVRS[1:0] bits to 00b.
(3) Set a new value to the CVRS[1:0]bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

31.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): CMPC0.CMPMON 000A 0C8Ch, CMPC1.CMPMON 000A 0CACh, CMPC2.CMPMON 000A 0CCCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPMON0	Comparator Output Monitor Flag *1	0: Comparator output is 0. 1: Comparator output is 1.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When comparator operation is enabled (CMPCTL.HCMPON and COE bits are 1) while the noise filter is disabled (CMPCTL.CDFS[1:0] bits are 00b), read the CMPMON0 bit twice and use the value only when the results match.

31.2.5 Comparator External Output Enable Register (CMPIOC)

- CMPC0

Address(es): CMPC0.CMPIOC 000A 0C90h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CPOE	External Pin Output Enable	0: Output to the comparator external pin is disabled (the output signal is low level) 1: Output to the comparator external pin is enabled	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VREFEN	Internal Reference Voltage On/Off Control*1	0: Internal reference voltage off 1: Internal reference voltage on	R/W

Note 1. Set this bit to 1 when using the internal reference voltage as input.
When the VREFEN bit is set to 1, input of the internal reference voltage is enabled for all channels.

- CMPC1, CMPC2

Address(es): CMPC1.CMPIOC 000A 0CB0h, CMPC2.CMPIOC 000A 0CD0h

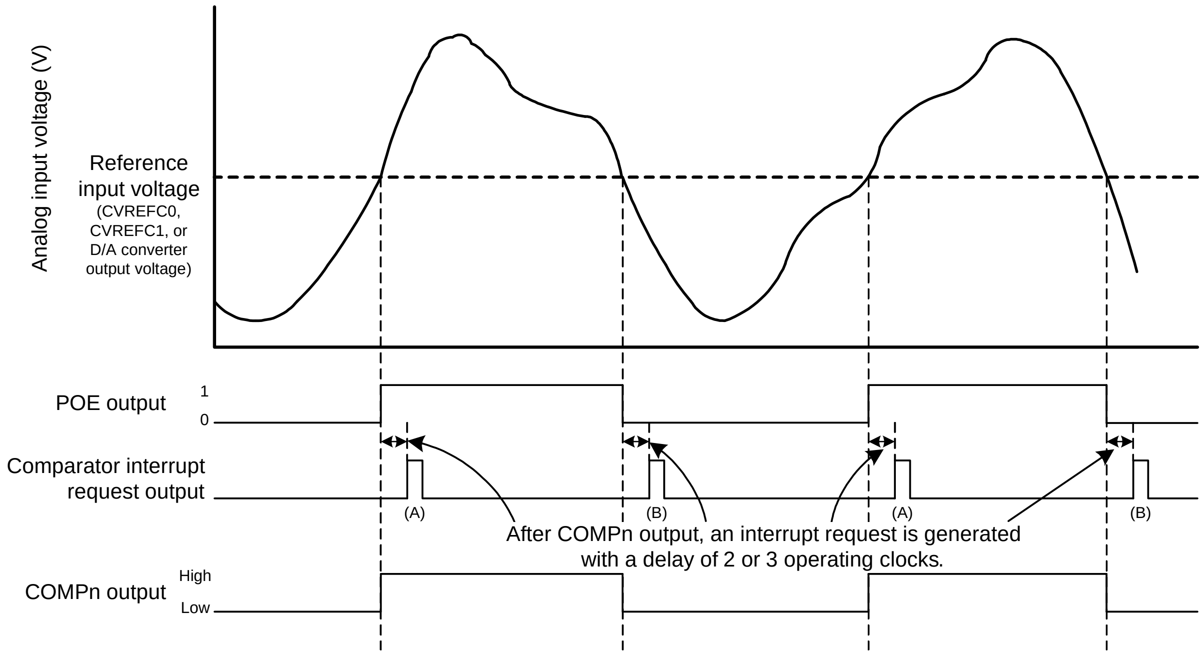
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPOE	External Pin Output Enable	0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

31.3 Operation

31.3.1 Comparator Operation Example

Figure 31.2 shows an operation example of the comparator. The COMPn (n = 0 to 2) output becomes high when the analog input voltage is higher than the reference input voltage, and the COMPn output becomes low when the analog input voltage is lower than the reference input voltage (when the CMPCTL.CINV bit is 0). When the comparator output changes, an interrupt request is output.



n = 0 to 2

Note: The above diagram applies when the CMPIOC.CPOE bit is 1 (pin output enabled), the CMPCTL.CDFS[1:0] bits are 00b (filter not used), the CMPCTL.CEG[1:0] bits are 11b (both edges selected), and the CMPCTL.CINV bit is 0 (comparator output not inverted).

Figure 31.2 Comparator Operation Example

31.3.2 Noise Filter

Comparator C contains a noise filter. The sampling clock can be selected by the CMPCTL.CDFS[1:0] bits.
The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, that value is determined as the noise filter output at the next sampling clock.
Figure 31.3 shows the configuration of the noise filter and edge detector and Figure 31.4 shows an example of the comparator noise filter and interrupt operation.

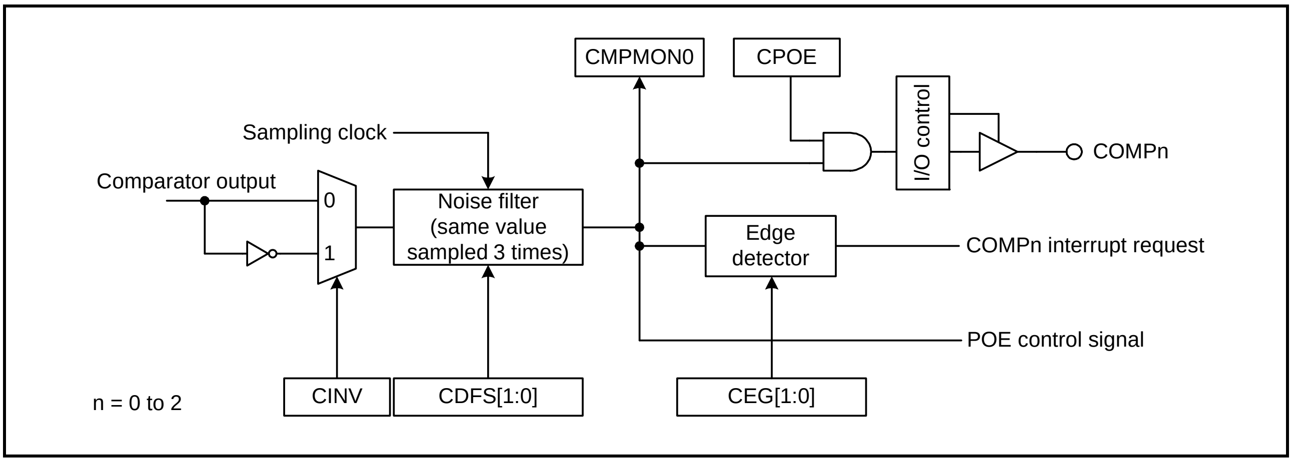
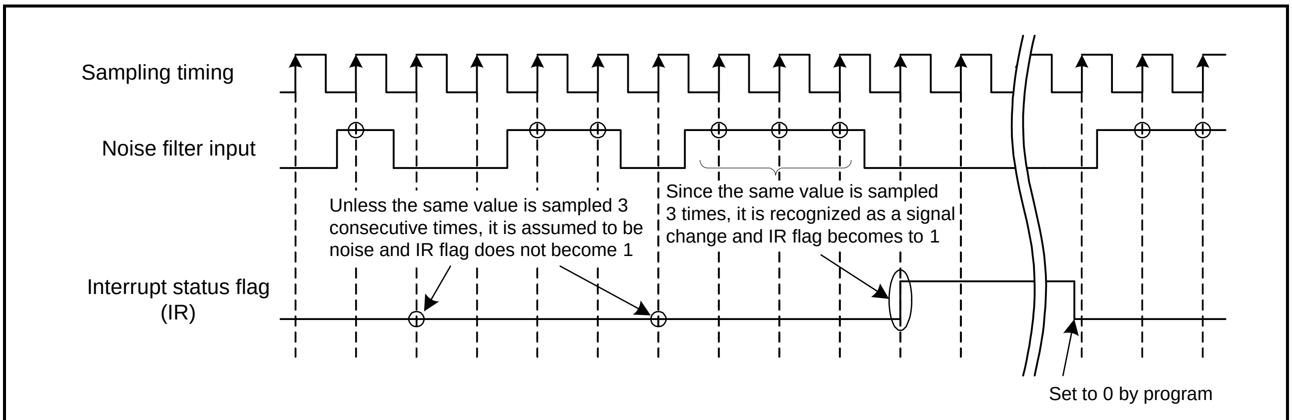


Figure 31.3 Noise Filter and Edge Detector Configuration



Note: The above operation example applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).
The noise canceling width can be changed by setting the value of the CMPCTL.CDFS[1:0] bits to select the sampling timing.

Figure 31.4 Comparator Noise Filter and Interrupt Operation Example

31.3.3 Comparator Interrupts

Comparator C generates an interrupt request upon detecting any changes in the comparison result.
When using the comparator interrupt, set at least one of bits CMPCTL.CEG[1:0] to 1 (to a value other than 00b (interrupt request is not generated)).
To use the comparator interrupt, use the following setting procedure. Note that steps (1), (2), and (3) can be set in any order.

(1) When using the on-chip D/A converter output voltage as the reference input voltage, set the D/A converter for generating comparator C reference voltage and enable operation.
(2) Set the CMPSEL0 or CMPSEL1 register to set the input of the comparator.
(3) Set the CMPCTL.CINV and CDFS[1:0] bits to select inversion or non-inversion processing and the sampling timing of the noise filter.
(4) Enable the edge detection for the comparator output (set the CMPCTL.CEG[1:0] bits to a value other than 00b).
(5) Enable input of the comparator (set the CMPCTL.HCMPON bit to 1) and wait for the time until the comparator operation is stabilized.
(6) Enable output of the comparator (set the CMPCTL.COE bit to 1).

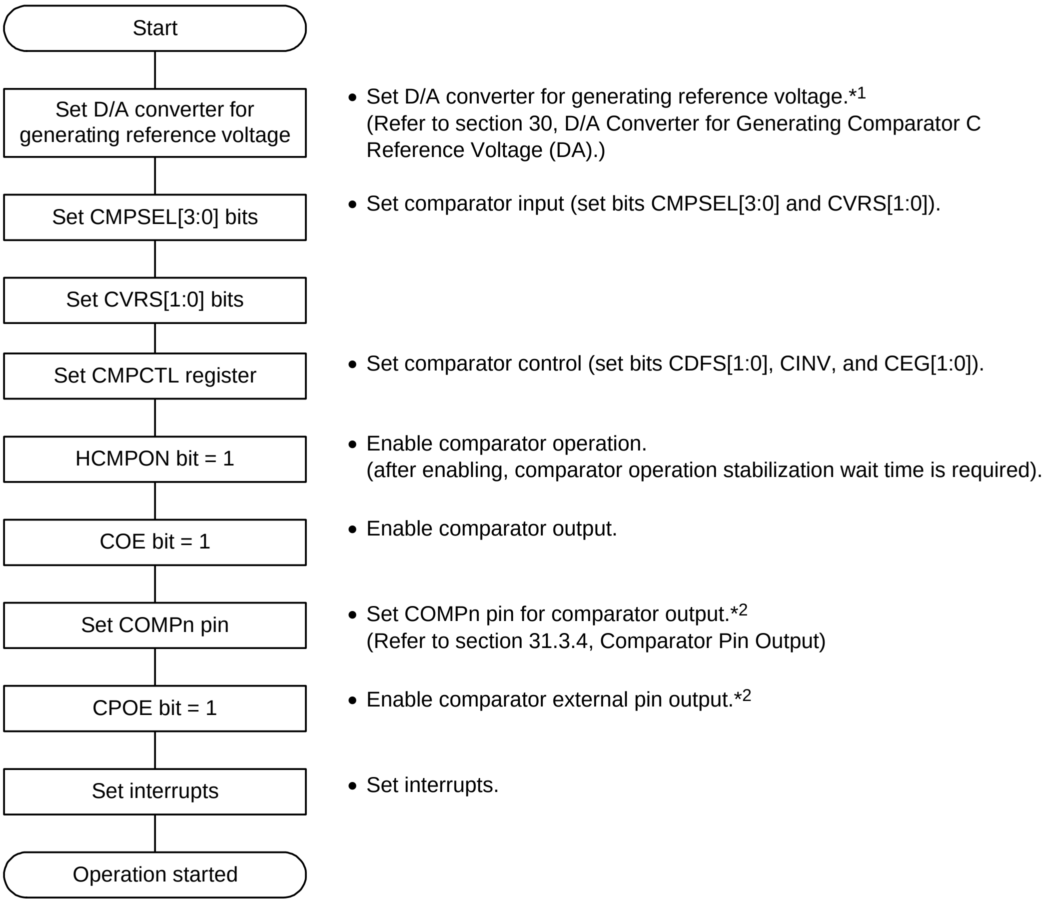
31.3.4 Comparator Pin Output

The comparison results can be output to the COMPn pins (n = 0 to 2). The CMPCTL.CINV bit can be used to set the output polarity (non-inverted output or inverted output), and the CMPIOC.CPOE bit can be used to enable or disable the output.
To output the comparison result to the external pin COMPn, use the following setting procedure. Note that the ports are set to input after reset.

(1) Execute steps (1) to (3) and steps (5) and (6) shown in section 31.3.3, Comparator Interrupts.
(2) Enable output of the comparison result to the external pin (set the CMPIOC.CPOE bit to 1).
(3) Set the port register and the pin function control register corresponding to each comparator output pin.

31.3.5 Comparator Setting Flowchart

Figure 31.5 shows the flowchart for setting the comparator-related registers.



n = 0 to 2
Note 1. This is not required when the external reference voltage is used.
Note 2. This is not required when the comparison result is not output to the external pin.

Figure 31.5 Comparator Operation Setting Flowchart

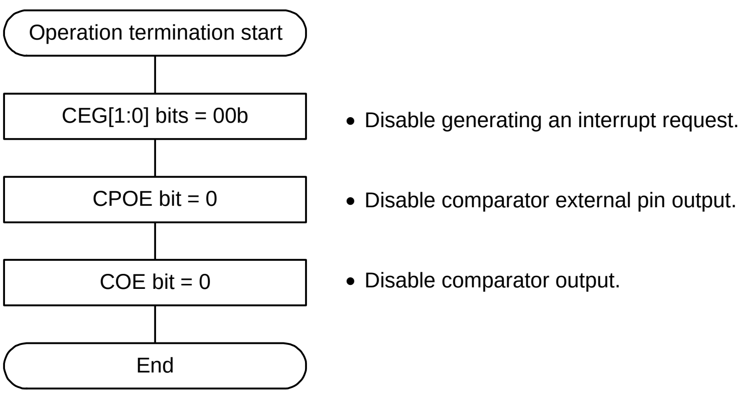


Figure 31.6 Comparator Operation Termination Flowchart

31.4 Usage Notes

31.4.1 Module Stop Function Setting

Operation of comparator C can be disabled or enabled using module stop control register B (MSTPCRB). After the reset, comparator C is halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

31.4.2 Comparator C Operation in Module Stop State

When the module stop state is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in the module stop state, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

31.4.3 Comparator C Operation in Software Standby Mode

When software standby mode is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in software standby mode, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

31.4.4 Setting the D/A Converter for Generating Comparator C Reference Voltage

Set the D/A converter for generating comparator C reference voltage and wait for the D/A converter output settling time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

32. Data Operation Circuit (DOC)

32.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

Table 32.1 lists the data operation circuit specifications and Figure 32.1 shows a block diagram of the data operation circuit.

16-bit data is compared and an interrupt can be generated when a selected condition applies.

Table 32.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module stop state can be set.
Interrupts	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

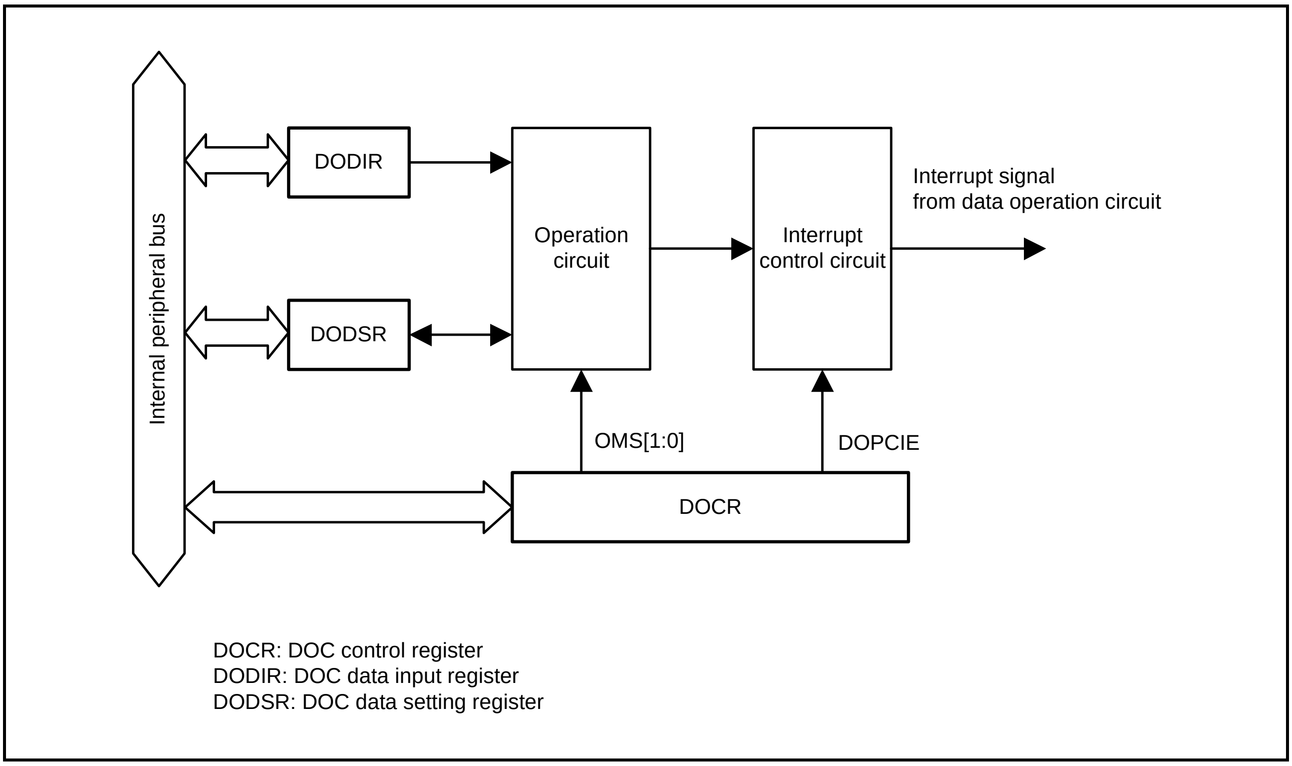


Figure 32.1 DOC Block Diagram

32.2 Register Descriptions

32.2.1 DOC Control Register (DOCR)

Address(es): 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL*1	Detection Condition Select	Result of data comparison 0: Data mismatch is detected. 1: Data match is detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation.	R
b6	DOPCFCL	DOPCF Clear	0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.
This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the data operation circuit.

DOPCF Flag (Data Operation Circuit Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL Bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.
This bit is read as 0.

32.2.2 DOC Data Input Register (DODIR)

Address(es): 0008 B082h

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

DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

32.2.3 DOC Data Setting Register (DODSR)

Address(es): 0008 B084h

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

DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

32.3 Operation

32.3.1 Data Comparison Mode

Figure 32.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

(1) Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
(2) The 16-bit reference data is set in DODSR.
(3) 16-bit data for comparison is written to DODIR.
(4) Writing of 16-bit data continues until all data for comparison have been written to DODIR.
(5) If a value written to DODIR does not match that in DODSR*1, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

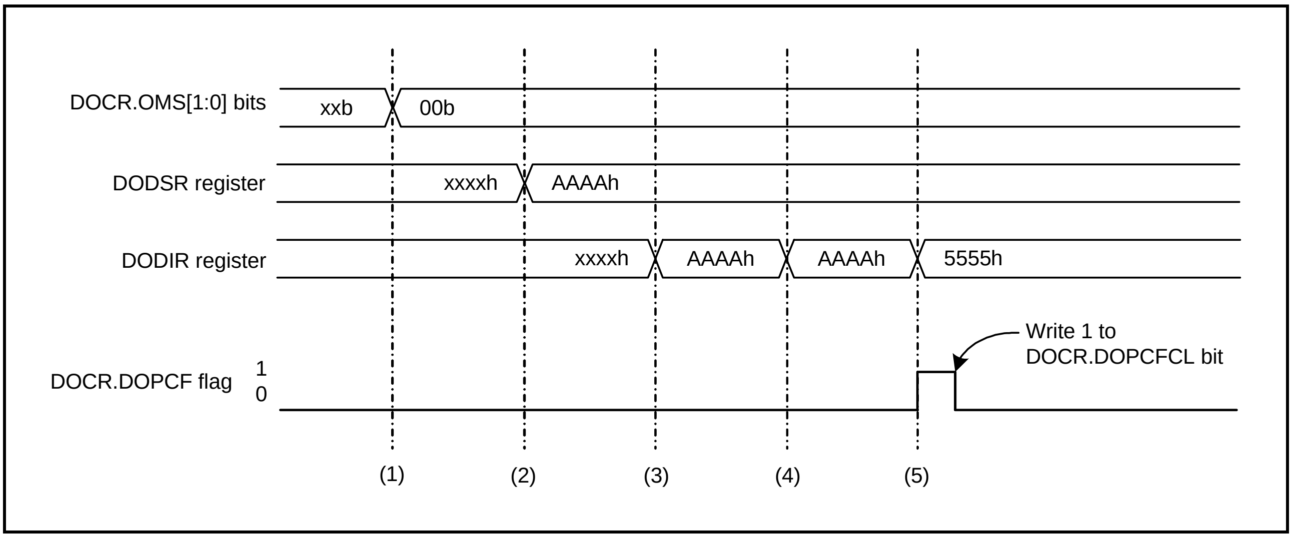


Figure 32.2 Example of Operation in Data Comparison Mode

32.3.2 Data Addition Mode

Figure 32.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for addition have been written to DODIR.
(5) If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

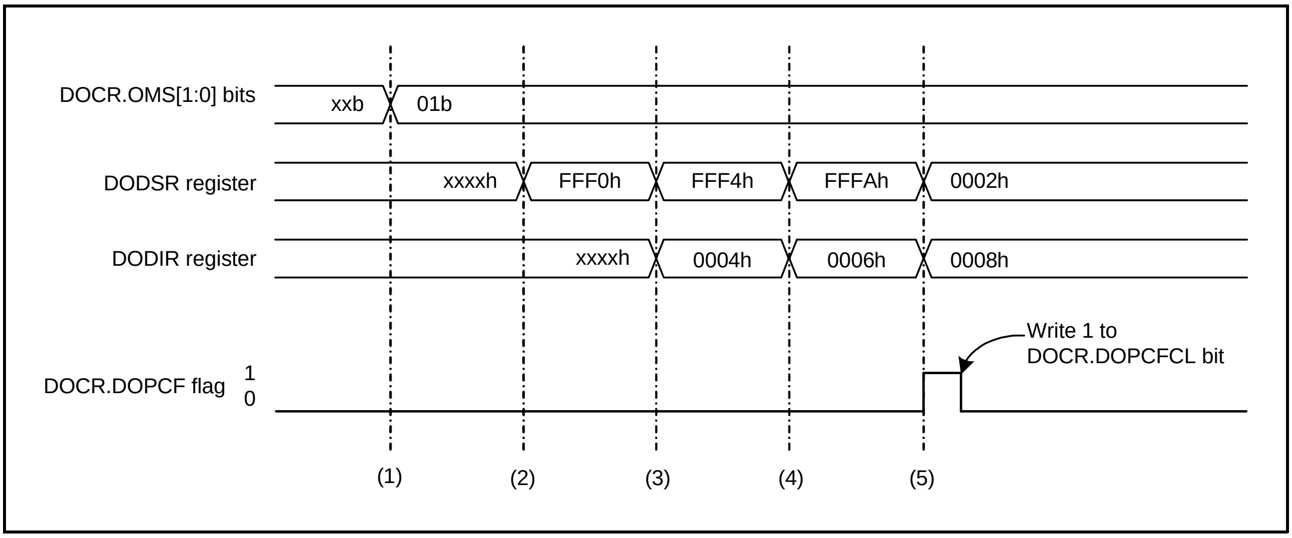


Figure 32.3 Example of Operation in Data Addition Mode

32.3.3 Data Subtraction Mode

Figure 32.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
(5) If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

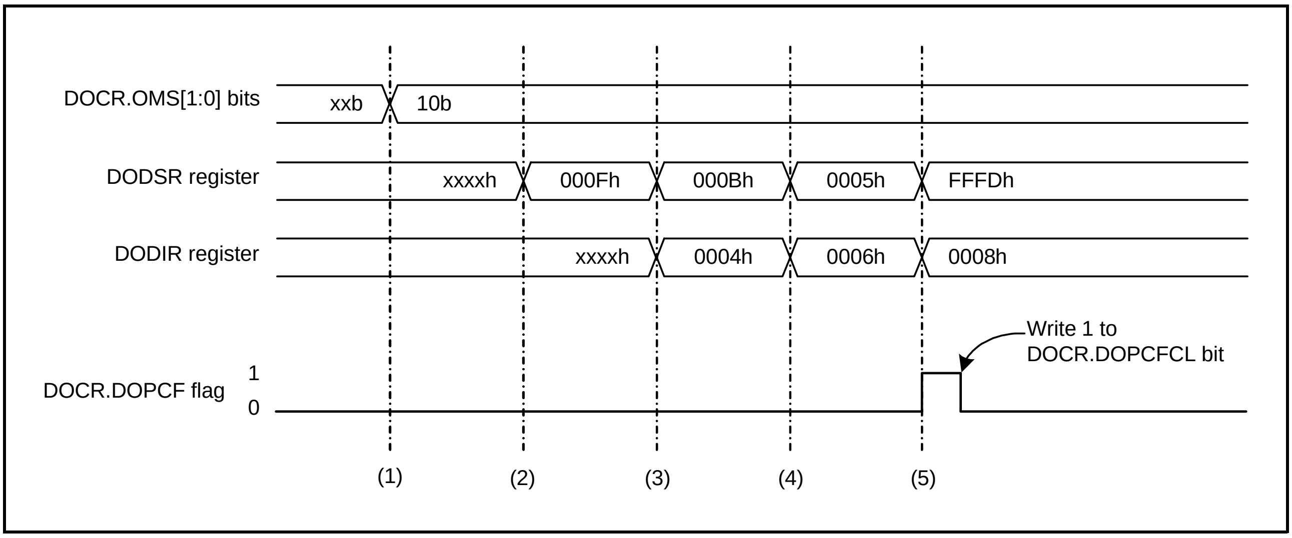


Figure 32.4 Example of Operation in Data Subtraction Mode

32.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1. Table 32.2 describes the interrupt request.

Table 32.2 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Generation Timing
Data operation circuit interrupt	DOPCF	• The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

32.5 Usage Note

32.5.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

33. RAM

This MCU has an on-chip high-speed static RAM.

33.1 Overview

Table 33.1 lists the specifications of the RAM.

Table 33.1 Specifications of RAM

Item	Description
RAM capacity	12 Kbytes (RAM0: 12 Kbytes)
RAM address	RAM0: 0000 0000h to 0000 27FFh, 0000 4000h to 0000 4A7Fh
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 for RAM0.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.2, System Control Register 1 (SYSCR1).

33.2 Operation

33.2.1 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 RAM0.
Stopping supply of the clock signal places the RAM0 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.

34. Flash Memory

This MCU has packages with 64 and 128 Kbyte flash memory (ROM) for storing code for storing data.
In this section, "PCLK" is used to refer to PCLKB.

34.1 Overview

Table 34.1 lists the Flash Memory Specifications.
Table 34.7 lists the I/O Pins Used in Boot Mode.

Table 34.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 128 Kbytes • Extra area: Stores the start-up area information, access window information, and unique ID
Software commands	• The following commands are implemented: Program, blank check, block erase, all-block erase • The following commands are implemented for programming the extra area: Start-up area information program, access window information program
Value after erase	• ROM: FFh
Interrupt	An interrupt (FRDYI) is generated upon completion of software command processing or forced stop processing.
On-board programming	Boot mode (SCI)*1 • Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication. • The user area is rewritable. Boot mode (FINE interface)*1 • The FINE is used. • The user area is rewritable. Self-programming in single-chip mode • The user area is rewritable using the flash rewrite routine in the user program.
Off-board programming	The user area is rewritable using a flash programmer compatible with this MCU.
ID code protection	• Connection with the serial programmer can be enabled or disabled using ID codes in boot mode. • Connection with the on-chip debugging emulator can be enabled or disabled using ID codes.
Start-up program protection	This function is used to safely rewrite block 0 to block 7.
Area protection	This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming.

Note 1. Refer to "PG-FP5 Flash Memory Programmer User's Manual" and "Renesas Flash Programmer Flash memory programming software User's Manual" for more details.

34.2 ROM Area and Block Configuration

The maximum ROM size of this MCU is 128 Kbytes. The ROM area is divided into blocks. A block is 2-Kbyte area. When executing the block erase command, the memory is erased by the block. Figure 34.1 shows the ROM Area and Block Configuration.

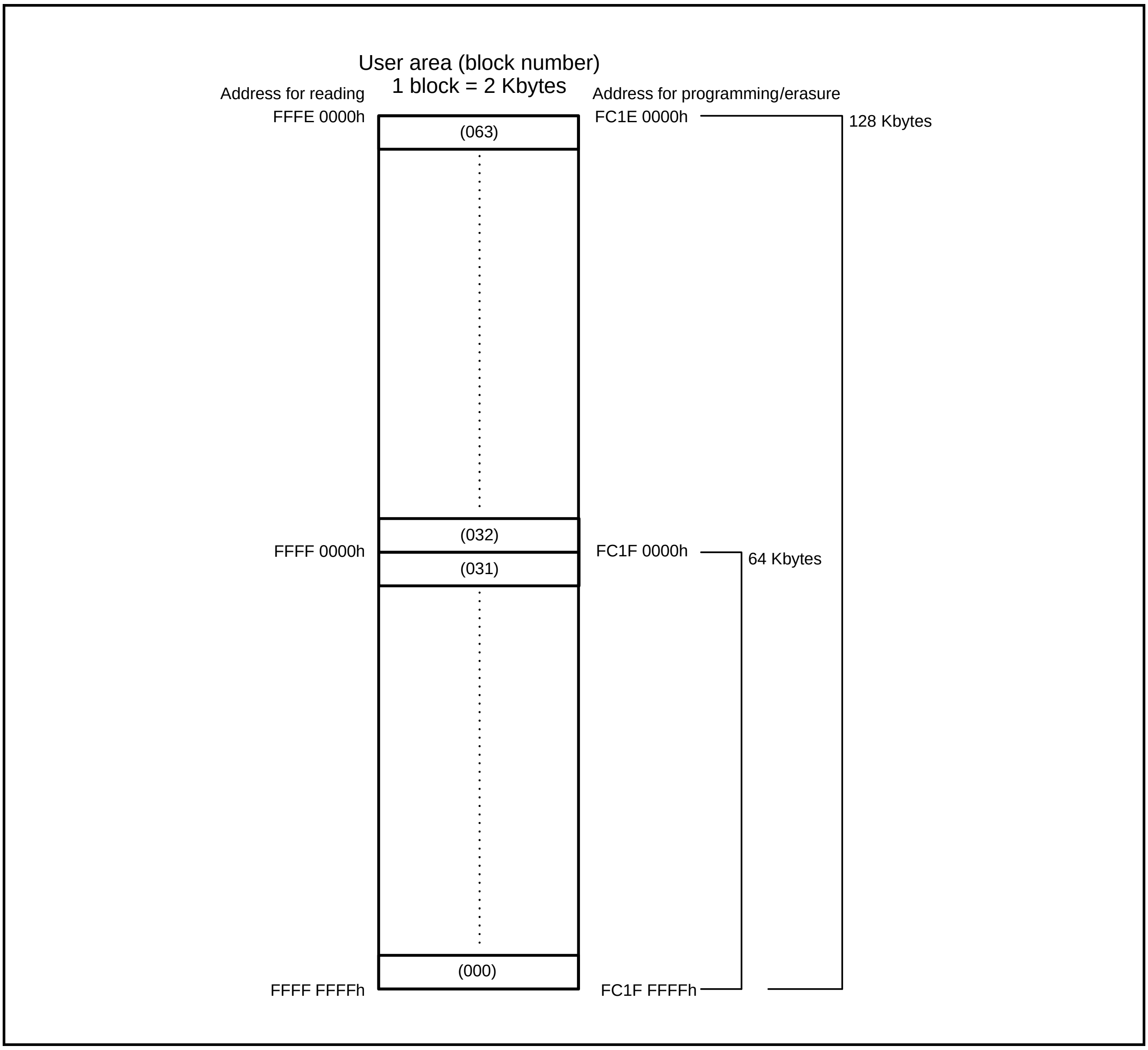


Figure 34.1 ROM Area and Block Configuration

Table 34.2 Correspondence Between ROM Capacity and Addresses for Reading

ROM Capacity	Addresses for Reading
128 Kbytes	FFFE 0000h to FFFF FFFFh
64 Kbytes	FFFF 0000h to FFFF FFFFh

34.3 Register Descriptions

34.3.1 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]								—	—	—	—	—	—	—	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: ROM is in read mode. 1: ROM can be placed in P/E mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FEKEY[7:0]	Key Code	The FEKEY[7:0] bits are used to control rewiring of the FENTRYR register. When rewriting the value of the low-order 8 bits, set the FEKEY[7:0] bits to AAh at the same time (write this register in 16 bits). The FEKEY[7:0] bits are read as 00h.	R/W

To rewrite the ROM, the FENTRY0 bit must be set to 1 to place the ROM in P/E mode.
When returning to read mode, set the FENTRYR register and confirm that its value has been rewritten before reading the ROM.
Refer to section 34.6.1, Sequencer Modes for details on P/E mode and read mode.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place the ROM in P/E mode.
[Setting condition]

- AA01h is written to the FENTRYR register when the FENTRYR register is 0000h.

Note: When entering ROM P/E mode, the instruction fetch address must be transferred to an area other than the ROM so that instruction fetching is not executed to the ROM. Copy necessary instruction code to the internal RAM and jump to the RAM.

[Clearing condition]

- AA00h is written to the FENTRYR register.

34.3.2 Protection Unlock Register (FPR)

Address(es): 007F C180h

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

x: Undefined

This write-only register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control. Writing to the FMCR register is enabled only when the following procedure is used to access the register.

Procedure to unlock protection

(1) Write A5h to the FPR register.
(2) Write a set value to the FPMCR register.
(3) Write the inverted set value to the FPMCR register.
(4) Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

34.3.3 Protection Unlock Status Register (FPSR)

Address(es): 007F C184h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PERR	Protect Error Flag	0: No error 1: An error occurs.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

PERR Flag (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection.

[Clearing condition]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 34.3.2, Protection Unlock Register (FPR).

34.3.4 Flash P/E Mode Control Register (FPMCR)

Address(es): 007F C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMS2	LVPE	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	0	0	0	0	1	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	FMS0	Flash Operating Mode Select 0	FMS2 FMS1 FMS0 0 0 0: ROM read mode 0 1 1: Discharge mode 1 1 0 1: ROM P/E mode 1 1 1: Discharge mode 2 Settings other than above are prohibited.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RPDIS	ROM P/E Disable	0: ROM programming/erasure enabled 1: ROM programming/erasure disabled	R/W
b4	FMS1	Flash Operating Mode Select 1	See the FMS0 bit.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LVPE	Low-Voltage P/E Mode Enable	0: Low-voltage P/E mode disabled 1: Low-voltage P/E mode enabled	R/W
b7	FMS2	Flash Operating Mode Select 2	See the FMS0 bit.	R/W

The FPMCR register is used to set the operating mode of the flash memory.
This register is protected. Set its value using the procedure to unlock protection. For details, refer to section 34.3.2, Protection Unlock Register (FPR).
When entering discharge mode 2 or ROM P/E mode, or during either of these modes, an instruction must be executed on the RAM.

FMS0, FMS1, and FMS2 Bits (Flash Operating Mode Select 0 to Flash Operating Mode Select 2)

These bits are used to set the operating mode of the flash memory.
[Transition from read mode to ROM P/E mode]
Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
Wait for ROM mode transition wait time 1 (tDIS, refer to section 35, Electrical Characteristics).
Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
Set the FMS2 bit = 1, the FMS1 bit = 0, the FMS0 bit = 1, and the RPDIS bit = 0.
Wait for ROM mode transition wait time 2 (tMS, refer to section 35, Electrical Characteristics).
[Transition from ROM P/E mode to read mode]
Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
Wait for ROM mode transition wait time 1 (tDIS, refer to section 35, Electrical Characteristics).
Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.
Wait for ROM mode transition wait time 2 (tMS, refer to section 35, Electrical Characteristics).

RPDIS Bit (ROM P/E Disable)

This bit is used to disable the execution of ROM programming/erasure with software.

LVPE Bit (Low-Voltage P/E Mode Enable)

Set this bit to 0 for programming/erasure in high-speed mode, and set this bit to 1 for programming/erasure in middle-speed mode.

34.3.5 Flash Initial Setting Register (FISR)

Address(es): 007F C1D8h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PCKA[4:0]	Peripheral Clock Notification	These bits are used to set the frequency of the FlashIF clock (FCLK).	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	SAS[1:0]	Start-Up Area Select	b7 b6 0 x: The start-up area is selected according to the start-up area settings of the extra area. 1 0: The start-up area is switched to the default area temporarily. 1 1: The start-up area is switched to the alternate area temporarily.	R/W

x: Don't care

Data can be written to the FISR register in ROM P/E mode.

PCKA[4:0] Bits (Peripheral Clock Notification)

These bits are used to set the frequency of the FlashIF clock (FCLK) when programming/erasing the ROM.
Set the FCLK frequency in the PCKA[4:0] bits before programming/erasure. Do not change the frequency during programming/erasure of the ROM.
[When FCLK is higher than 4 MHz]
Set a rounded-up value for a non-integer frequency.
For example, set 32 MHz (PCKA[4:0] bits = 11111b) when the frequency is 31.5 MHz.
[When FCLK is 4 MH or lower]
Do not use a non-integer frequency.
Use the FCLK at a frequency of 1, 2, 3, or 4 MHz.

Note: When the PCKA[4:0] bits are set to a frequency different from the FCLK, the data in the ROM may be damaged.

Table 34.3 Example of FlashIF Clock Frequency Settings

FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting	FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting	FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting
32	11111b	31	11110b	30	11101b
29	11100b	28	11011b	27	11010b
26	11001b	25	11000b	24	10111b
23	10110b	22	10101b	21	10100b
20	10011b	19	10010b	18	10001b
17	10000b	16	01111b	15	01110b
14	01101b	13	01100b	12	01011b
11	01010b	10	01001b	9	01000b
8	00111b	7	00110b	6	00101b
5	00100b	4	00011b	3	00010b
2	00001b	1	00000b	—	—

SAS[1:0] Bits (Start-Up Area Select)

These bits are used to select the start-up area. To change the start-up area, the following three methods can be used.

(1) When selecting the start-up area according to the start-up area settings of the extra area

With the SAS[1:0] bits set to 00b or 01b, the start-up area is selected according to the start-up area settings of the extra area. The settings are enabled after a reset is released.

(2) When switching the start-up area to the default area temporarily

When 10b is written to the SAS[1:0] bits, the start-up area is switched to the default area immediately after data is written to the register, regardless of the start-up area settings of the extra area.
When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

(3) When switching the start-up area to the alternative area temporarily

When 11b is written to the SAS[1:0] bits, the start-up area is switched to the alternative area, regardless of the start-up area settings of the extra area.
When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

34.3.6 Flash Reset Register (FRESETR)

Address(es): 007F C124h

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

Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: Flash control circuit reset is released. 1: Flash control circuit is reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FRESET Bit (Flash Reset)

When this bit is set to 1, registers FASR, FSARH, FSARL, FEARH, FEARL, FWB0, FWB1, FWB2, FWB3, FCR, and FEXCR are reset. Also, the values of registers FEAMH and FEAML are undefined. Do not access these registers during a reset. To release the reset, set this bit to 0.

Do not write to this register while executing a software command or rewriting the extra area.

34.3.7 Flash Area Select Register (FASR)

Address(es): 007F C104h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EXS	Extra Area Select	0: User area 1: Extra area	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Data can be written to the FASR register in ROM P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1.

Data cannot be written to this register while the FRESETR.FRESET bit is 1.

EXS Bit (Extra Area Select)

Set this bit to 1 before issuing a software command (start-up area information program or access window information program) for the extra area. Set this bit to 0 before issuing a software command (program, blank check, block erase, or all-block erase) for the user area.

After issuing a software command, do not change the value until changing it for issuing the next software command.

34.3.8 Flash Control Register (FCR)

Address(es): 007F C114h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMD[3:0]	Software Command Setting	b3 b0 0 0 0 1: Program 0 0 1 1: Blank check 0 1 0 0: Block erase 0 1 1 0: All-block erase Settings other than above are prohibited.*1	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	STOP	Forced Processing Stop	When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FCR register to 00h when the FSTATR1.FRDY flag is 1.

Data can be written to the FCR register when in ROM P/E mode and the ROM can be programmed/erased.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[3:0] Bits (Software Command Setting)

These bits are used to set a software command (program, blank check, block erase, or all-block erase).
The function of each command is described below.
[Program]

- Write the value set in registers FWB0, FWB1, FWB2, and FWB3 to the address set in registers FSARH and FSARL.

[Blank check]

- Check whether there is data in the area from the address set in registers FSARH and FSARL to the address set in registers FEARH and FEARL. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.

[Block erase]

- Erase consecutive areas specified in the flash memory by the blocks. Set the beginning address of the block in registers FSARH and FSARL and the end address in registers FEARH and FEARL.

[All-block erase]

- Erase all blocks in the ROM.
  All-block erase requires less time to erase the memory compared to block erase. When erasing the whole of the ROM area, set the beginning address of the ROM area in registers FSARH and FSARL, and the end address in registers FEARH and FEARL. Table 34.4 lists the setting address for all-block erase.

Table 34.4 Setting Address for All-Block Erase

Target	Memory Size	FSARH/FSARL	FEARH/FEARL
ROM	128 Kbytes	FC1E0000h	FC1FFFFFh
	64 Kbytes	FC1F0000h	FC1FFFFFh

STOP Bit (Forced Processing Stop)

This bit is used to forcibly stop the processing (blank check, block erase, or all-block erase) being executed.
After setting this bit to 1, wait until the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FRDY flag is 0 before executing the next processing.

34.3.9 Flash Extra Area Control Register (FEXCR)

Address(es): 007F C1DCh

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Software Command Setting	b2 b0 0 0 1: Start-up area information program 0 1 0: Access window information program Settings other than above are prohibited.*1	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FEXCR register to 00h when the FSTATR1.EXRDY flag is 1.

Data can be written to the FEXCR register when in ROM P/E mode and the ROM can be programmed/erased.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[2:0] Bits (Software Command Setting)

These bits are used to set a software command (start-up area information program or access window information program).
The details of each command are described below.
[Start-up area information program]
This command is used to switch the start-up area used for start-up program protection.

- When setting the start-up area to the default area
  Set registers FWB0, FWB1, FWB2, and FWB3 to FFFFh, and execute this command.
- When setting the start-up area to the alternative area
  Set the FWB0 register to FEFFh, set the FWB1 register to FFFFh, set registers FWB2 and FWB3 to FFFFh, and execute this command.

When registers FWB0, FWB1, FWB2, and FWB3 are set to values other than the above, do not execute the start-up area

information program.
[Access window information program]
This command is used to set the access window used for area protection.
Set the access window in block units.
Specify the access window start address, which is the beginning address of the access window in the FWB0 register, specify the access window end address, which is the next address of the last address of the access window in the FWB1 register, and issue this command. Set bit 21 to bit 10 of the address for programming/erasure in each register.
If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.EXRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FSTATR1.EXRDY flag is 0 before executing the next processing.
Writing to the extra area is started by writing 1 to the OPST bit. Do not write to the CMD[2:0] bits while a software command is being executed.

34.3.10 Flash Processing Start Address Register H (FSARH)

Address(es): 007F C110h

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

The FSARH register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.11 Flash Processing Start Address Register L (FSARL)

Address(es): 007F C108h

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

The FSARL register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
To set the ROM area, set bit 2 to bit 0 to 000b.
Data can be written to this register in ROM P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
This register is incremented by 8h if the code flash memory is specified after a program command is executed. Therefore, it is not necessary to set the target address to be written to this register when executing a program command sequentially.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.12 Flash Processing End Address Register H (FEARH)

Address(es): 007F C120h

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

The FEARH register is used to set the end address of the target processing range in the flash memory when a software command is executed.

Set bit 31 to bit 25 and bit 20 to bit 16 of the flash memory address for programming/erasure in this register.

Data can be written to this register in ROM P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.13 Flash Processing End Address Register L (FEARL)

Address(es): 007F C118h

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

The FEARH register is used to set the end address of the target range for processing when a software command is executed.

Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.

When setting the ROM area, set bit 2 to bit 0 to 000b.

Data can be written to this register in ROM P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.14 Flash Write Buffer n Register (FWBn) (n = 0 to 3)

Address(es): FWB0 007F C130h, FWB1 007F C138h, FWB2 007F C140h, FWB3 007F C144h

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

This register is used to set the data for programming the ROM, or extra area. The data can be written in ROM P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

The read value of this register is undefined while executing a software command set by the FCR register or the FEXCR register.

When programming the extra area, set the 4-byte data for programming in registers FWB0 and FWB1.

When programming the ROM, set the 8-byte data for programming in registers FWB0 to FWB3. Figure 34.2 shows the relationship between the addresses indicated by registers FSARH and FSARL and the data set in the FWBn register.

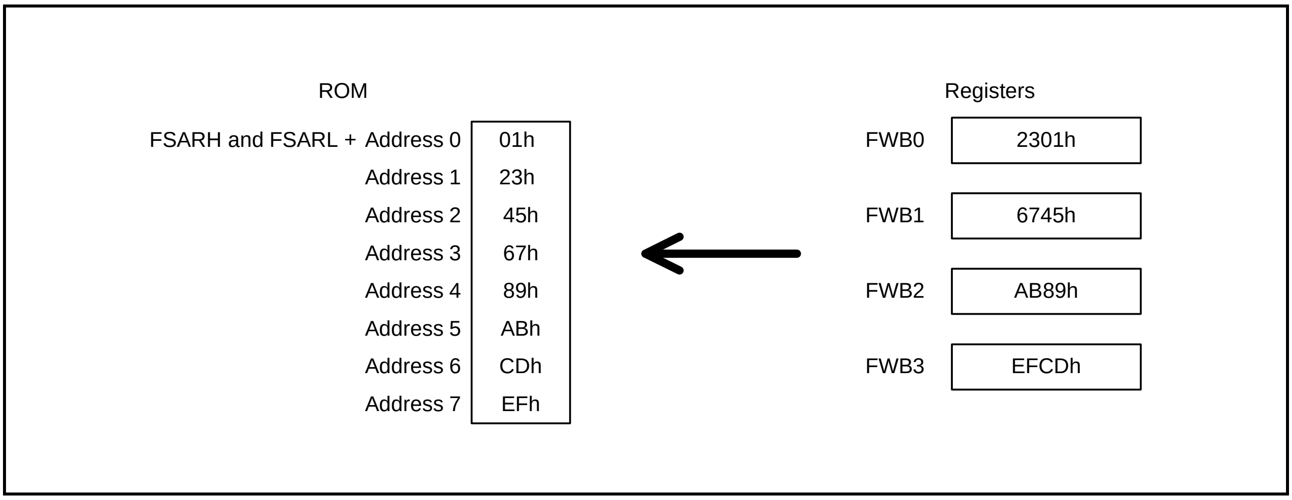


Figure 34.2 FWBn Register Setting Values and Data Allocation in the ROM

34.3.15 Flash Status Register 0 (FSTATR0)

Address(es): 007F C1F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EILGLERR	ILGLERR	BCERR	—	PRGERR	ERERR
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERERR	Erase Error Flag	0: Erasure terminates normally. 1: An error occurs during erasure.	R
b1	PRGERR	Program Error Flag	0: Programming terminates normally. 1: An error occurs during programming.	R
b2	—	Reserved	The read value is undefined.	R
b3	BCERR	Blank Check Error Flag	0: Blank checking terminates normally. 1: An error occurs during blank checking.	R
b4	ILGLERR	Illegal Command Error Flag	0: No illegal software command or illegal access is detected. 1: An illegal command or illegal access is detected.	R
b5	EILGLERR	Extra Area Illegal Command Error Flag	0: No illegal command or illegal access to the extra area is detected. 1: An illegal command or illegal access to the extra area is detected.	R
b7, b6	—	Reserved	The read value is undefined.	R

This register is a status register used to confirm the result of executing a software command. Each error flag is set to 0 when the next software command is executed.

ERERR Flag (Erase Error Flag)

This flag indicates the result of the erase processing for the ROM.

[Setting condition]

- An error occurs during erasure.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR Flag (Program Error Flag)

This flag indicates the result of the program processing for the ROM.

[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The next software command is executed.

BCERR Flag (Blank Check Error Flag)

This flag indicates the result of the blank check processing for the ROM.

[Setting condition]

- An error occurs during blank checking.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank checking.

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates the result of executing a software command.

[Setting conditions]

- Programming/erasure is executed to an area other than the access window range.
- A blank check or block erase command is executed when the set value of registers FSARH and FSARL is larger than the set value of registers FEARH and FEARL.
- Program and block erase commands are executed when the FASR.EXS bit is 1.
- An all-block erase command is executed while the access window is set.
- An all-block erase command is executed without setting registers FSARH and FSARL and registers FEARH and FEARL properly.

[Clearing condition]

- The next software command is executed.

EILGLERR Flag (Extra Area Illegal Command Error Flag)

This flag indicates the result of executing a software command for the extra area.

[Setting condition]

- A software command for the extra area is executed when the FASR.EXS bit is 0.

[Clearing condition]

- The next software command is executed.

34.3.16 Flash Status Register 1 (FSTATR1)

Address(es): 007F C12Ch

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0.	R
b2	—	Reserved	This bit is read as 1.	R
b5 to b3	—	Reserved	These bits are read as 0.	R
b6	FRDY	Flash Ready Flag	0: Other than below 1: 00h can be written to the FCR register (processing to complete the software command).	R
b7	EXRDY	Extra Area Ready Flag	0: Other than below 1: 00h can be written to the FEXCR register (processing to complete the software command).	R

This register is a status register used to confirm the result of executing a software command. Each flag is set to 0 when the next software command is executed.

FRDY Flag (Flash Ready Flag)

This flag is used to confirm whether a software command is executed.

This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.

Also, an interrupt (FRDYI) is generated when this flag becomes 1.

EXRDY Flag (Extra Area Ready Flag)

This flag is used to confirm whether a software command for the extra area is executed.

This flag is set to 1 when processing of the executed software command is completed, and 0 when the FEXCR.OPST bit is set to 0.

Also, an interrupt (FRDYI) is generated when this flag becomes 1.

34.3.17 Flash Error Address Monitor Register H (FEAMH)

Address(es): 007F C1E8h

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

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 31 to bit 25 and bit 20 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 31 to bit 25 and bit 20 to bit 16 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
If the software command terminates normally, this register stores bit 31 to bit 25 and bit 20 to bit 16 of the end address at execution of the command.
Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.18 Flash Error Address Monitor Register L (FEAML)

Address(es): 007F C1E0h

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

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address of the area where the error has occurred for the block erase command or all-block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.
When executing a software command for the ROM, low-order 2 bits become 00b.
Refer to Figure 34.1 for details on the addresses of the flash memory.

34.3.19 Flash Start-Up Setting Monitor Register (FSCMR)

Address(es): 007F C1C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SASMF	—	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	0	1	1	Value set by user*1	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.	R
b8	SASMF	Start-Up Area Setting Monitor Flag	0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
b10, b9	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b11	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b14 to b12	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bit 8 in the FWB1 register after the start-up area information program command is executed.

SASMF Flag (Start-Up Area Setting Monitor Flag)

This flag is used to confirm the settings of the start-up area.
When this flag is 0, the user program is set to start up using the alternative area.
When this flag is 1, the user program is set to start up using the default area.

34.3.20 Flash Access Window Start Address Monitor Register (FAWSMR)

Address(es): 007F C1C8h

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

Note 1. The value of the blank product is 1. It is set to the same value set in bit 11 to bit 0 the FWB0 register after the access window information program command is executed.

This register is used to confirm the set value of the access window start address used for area protection.

34.3.21 Flash Access Window End Address Monitor Register (FAWEMR)

Address(es): 007F C1D0h

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

Note 1. The value of the blank product is 1. It is set to the same value set in bit 11 to bit 0 in the FWB1 register after the access window information program command is executed.

This register is used to confirm the set value of the access window end address used for area protection.

34.3.22 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): UIDR0 007F C350h, UIDR1 007F C354h, UIDR2 007F C358h, UIDR3 007F C35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip															

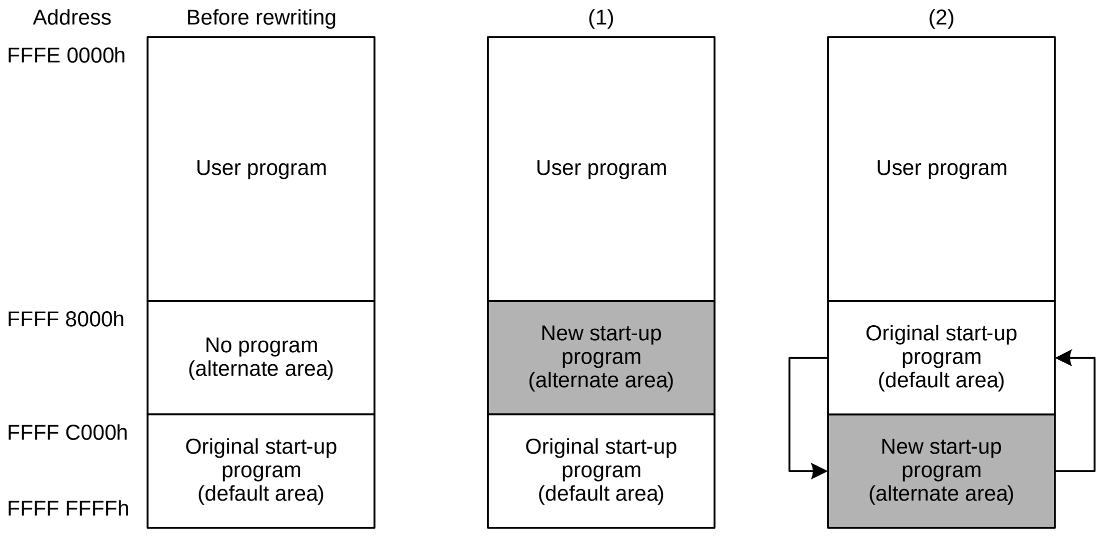
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

The UIDRn register stores a 16-byte ID code (unique ID) for identifying the individual MCU.
The unique ID is stored in the extra area of the flash memory and cannot be rewritten by the user.

34.4 Start-Up Program Protection

When rewriting the start-up program*1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly.
This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.
Figure 34.3 shows the Overview of the Start-Up Program Protection. In this figure, the default area indicates block 0 to block 7, and the alternate area indicates block 8 to block 15.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.



(1) Program a new start-up program in the alternate area. If the alternate area fails to be rewritten, the new start-up program can be rewritten again after starting up using the default area, because the original start-up program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the start-up area information program command. After that, the program in the alternate area starts after a reset.

Figure 34.3 Overview of the Start-Up Program Protection

34.5 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming.

Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.

Figure 34.4 shows the Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 128-Kbyte ROM).

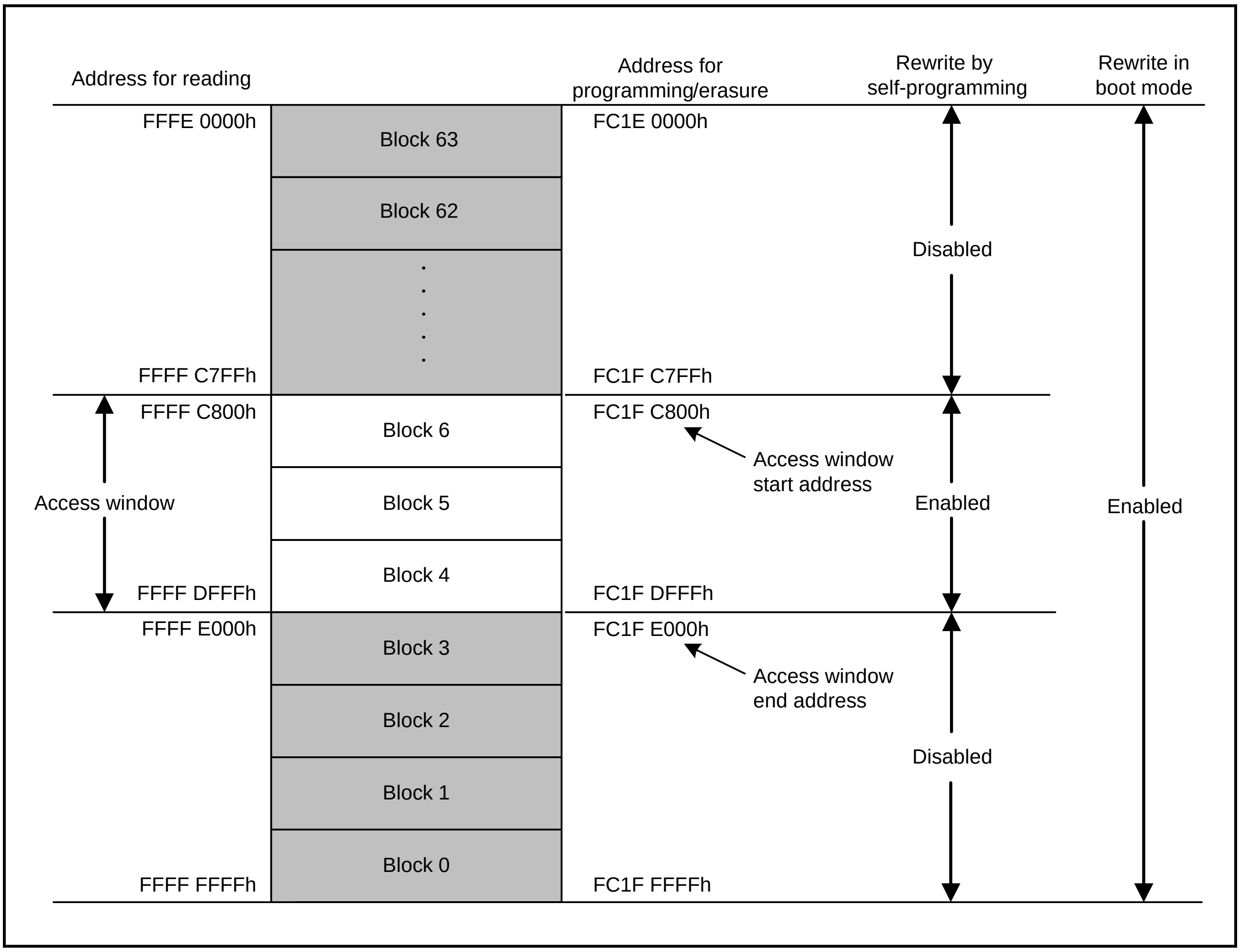


Figure 34.4 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 128-Kbyte ROM)

34.6 Programming and Erasure

The ROM can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.
The mode transitions and commands required to program or erase the ROM are described below. The descriptions apply in common to boot mode and single-chip mode.

34.6.1 Sequencer Modes

The sequencer has two modes. Transitions between modes are caused by writing to the FENTRYR register and setting the FPMCR register. Figure 34.5 is a diagram of mode transitions of the flash memory.

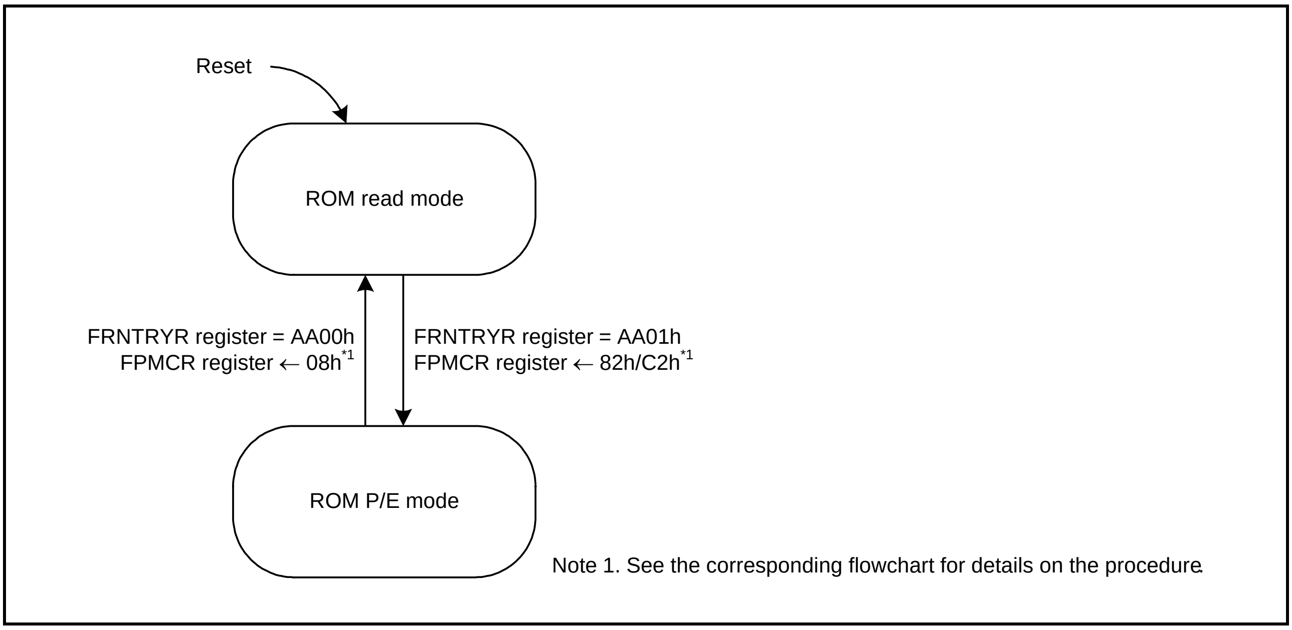


Figure 34.5 Mode Transitions of the Flash Memory

34.6.1.1 Read Mode

Read mode is for high-speed reading of the ROM. Reading from a ROM address for reading can be accomplished in one ICLK clock.

(1) ROM Read Mode

In this mode, both the ROM is in read mode. The sequencer enters this mode from P/E mode when setting the FPMCR register to 08h and setting the FENTRYR.FENTRY0 bit to 0.

34.6.1.2 P/E Mode

The P/E mode is for programming and erasure of the ROM.

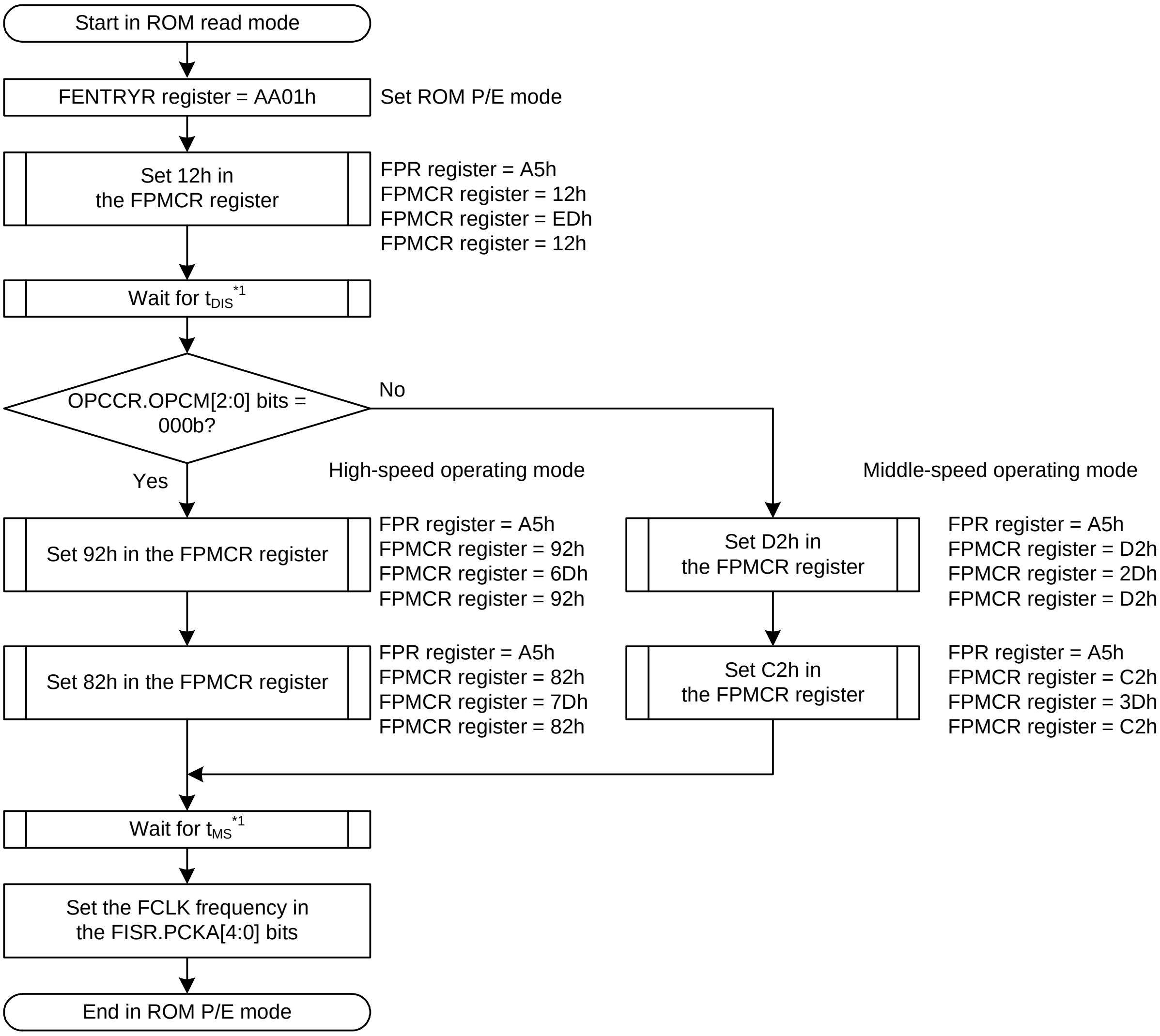
(1) ROM P/E Mode

In this mode, the ROM is in P/E mode. The sequencer enters this mode when setting the FENTRYR.FENTRY0 bit to 1 and setting the FPMCR register 82h or C2h.

34.6.2 Mode Transitions

34.6.2.1 Transition from Read Mode to P/E Mode

Switching to ROM P/E mode is required before executing a software command for the ROM.
Figure 34.6 shows the Procedure for Transition from ROM Read Mode to ROM P/E Mode.



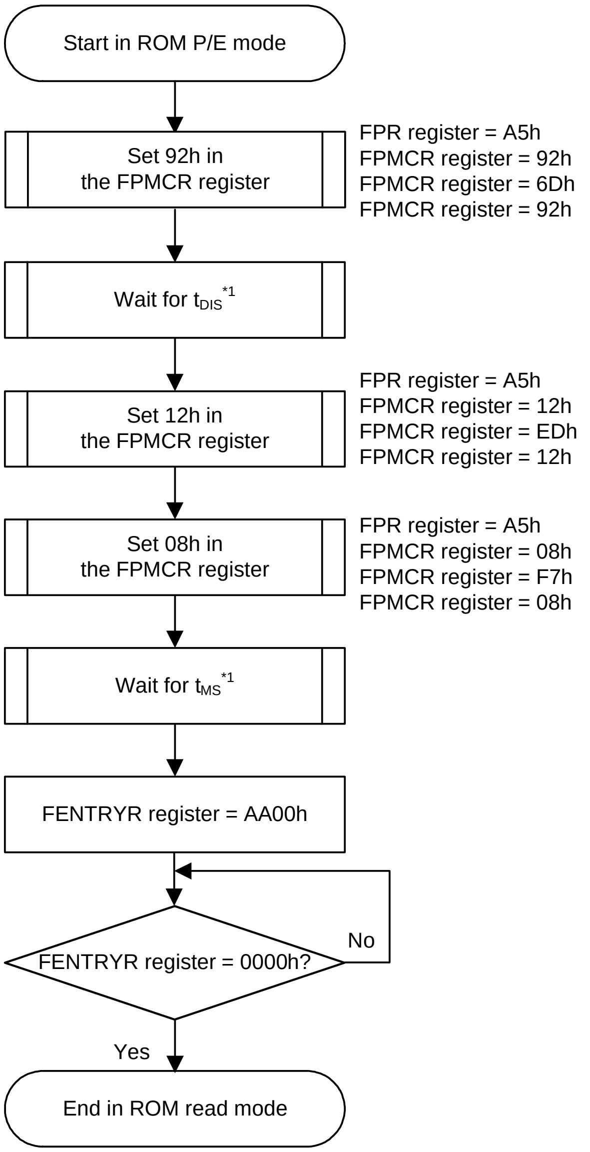
Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 34.6 Procedure for Transition from ROM Read Mode to ROM P/E Mode

34.6.2.2 Transition from P/E Mode to Read Mode

High-speed reading of the ROM requires switching to ROM read mode.

Figure 34.7 shows the Procedure for Transition from ROM P/E Mode to ROM Read Mode.



Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 34.7 Procedure for Transition from ROM P/E Mode to ROM Read Mode

34.6.3 Software Commands

Software commands consist of commands for programming and erasure and commands for programming start-up program area information and access window information. Table 34.5 lists the software commands for use with the flash memory.

Table 34.5 Software Commands

Command	Function
Program	ROM programming (8 bytes)
Block erase	ROM erasure
All-block erase	Erasure of all blocks in the ROM
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Start-up area information program	Rewrite the start-up area switching information used for start-up program protection.
Access window information program	Set the access window used for area protection.

34.6.4 Software Command Usage

This section describes how to use each software command, using flowcharts.

34.6.4.1 Program

Figure 34.8 shows the procedure to issue the program command.

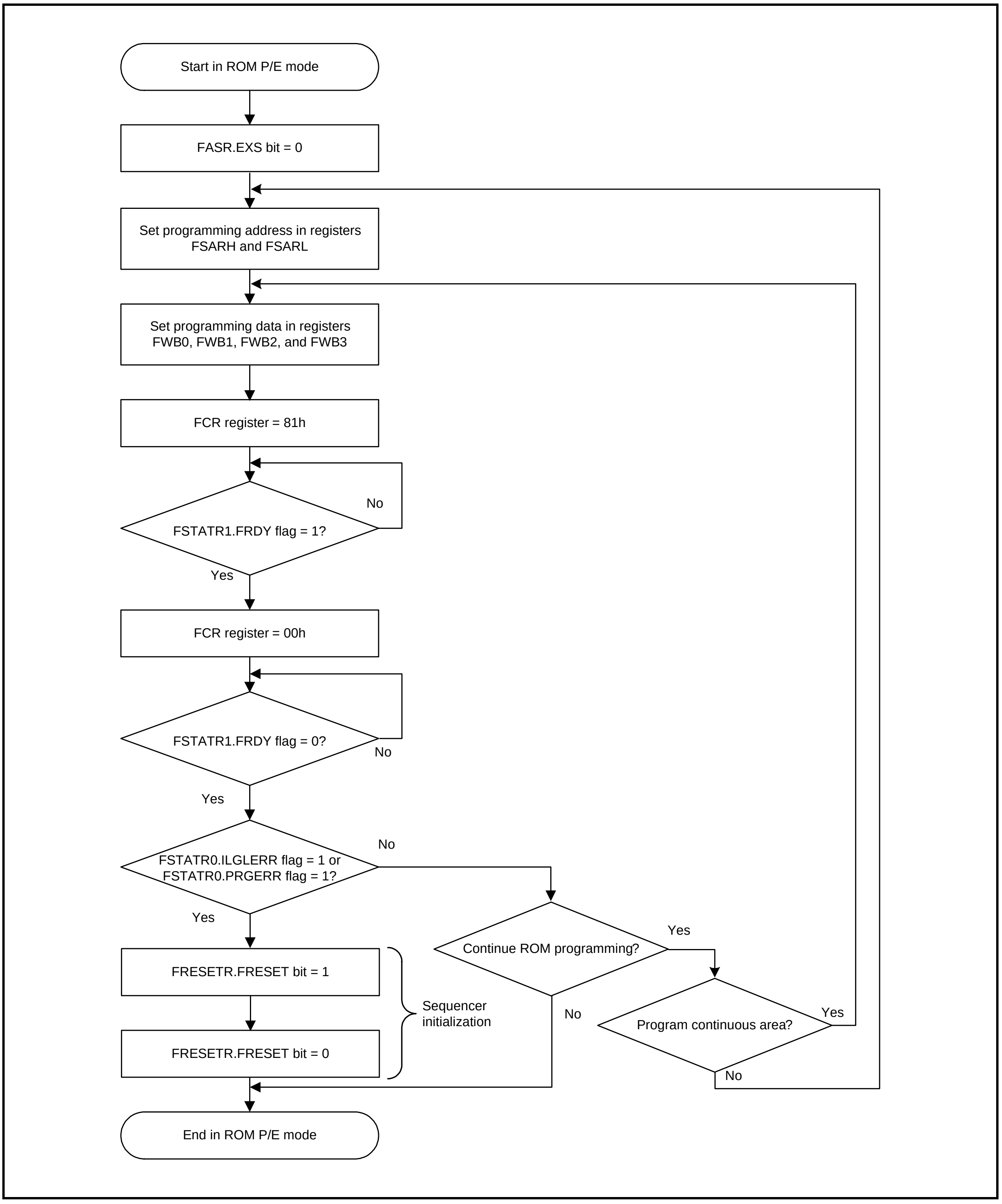


Figure 34.8 Procedure to Issue the Program Command

34.6.4.2 Block Erase

Figure 34.9 shows the procedure to issue the block erase command.

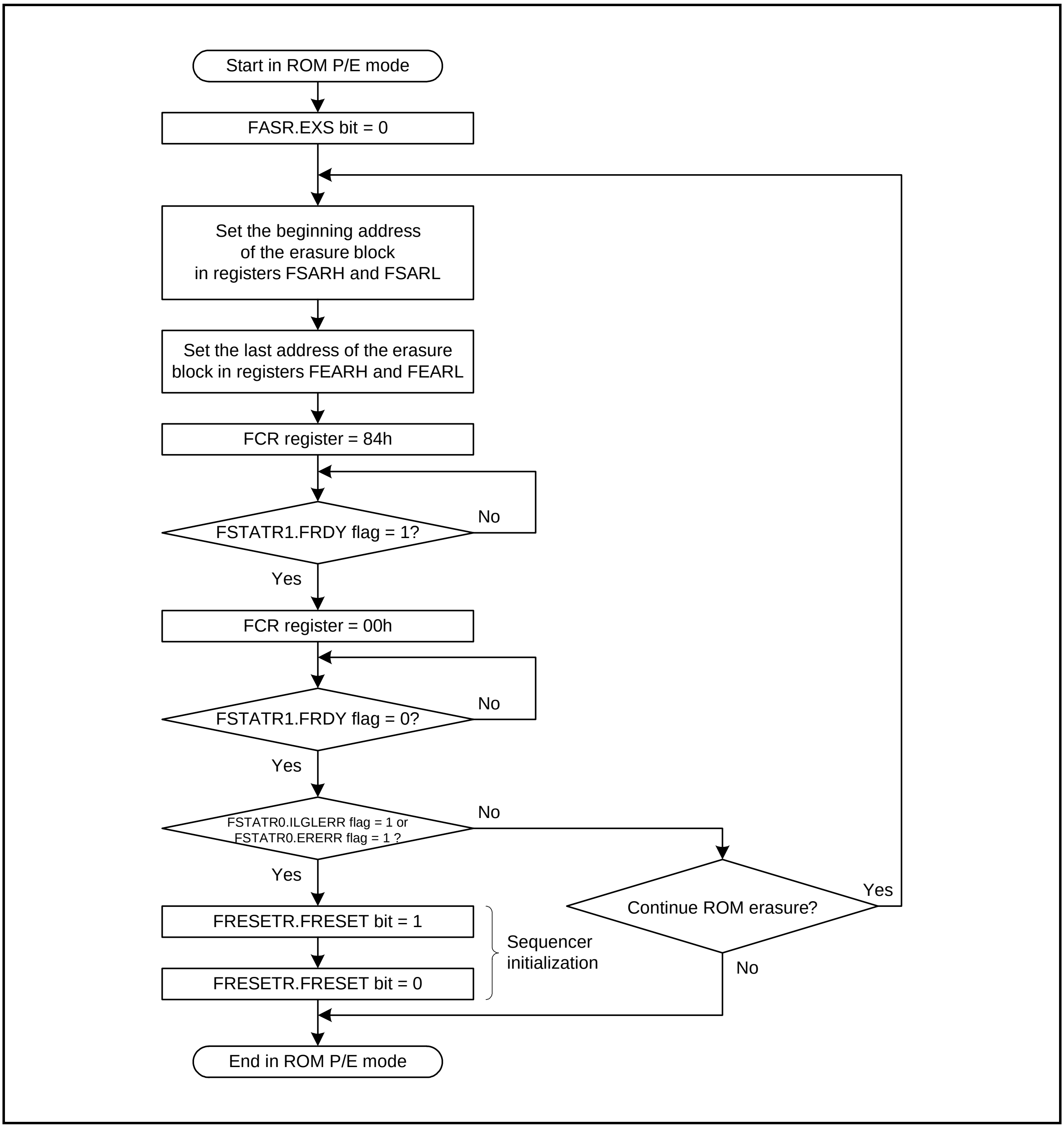


Figure 34.9 Procedure to Issue the Block Erase Command

34.6.4.3 All-Block Erase

Figure 34.10 shows the procedure to issue the all-block erase command.

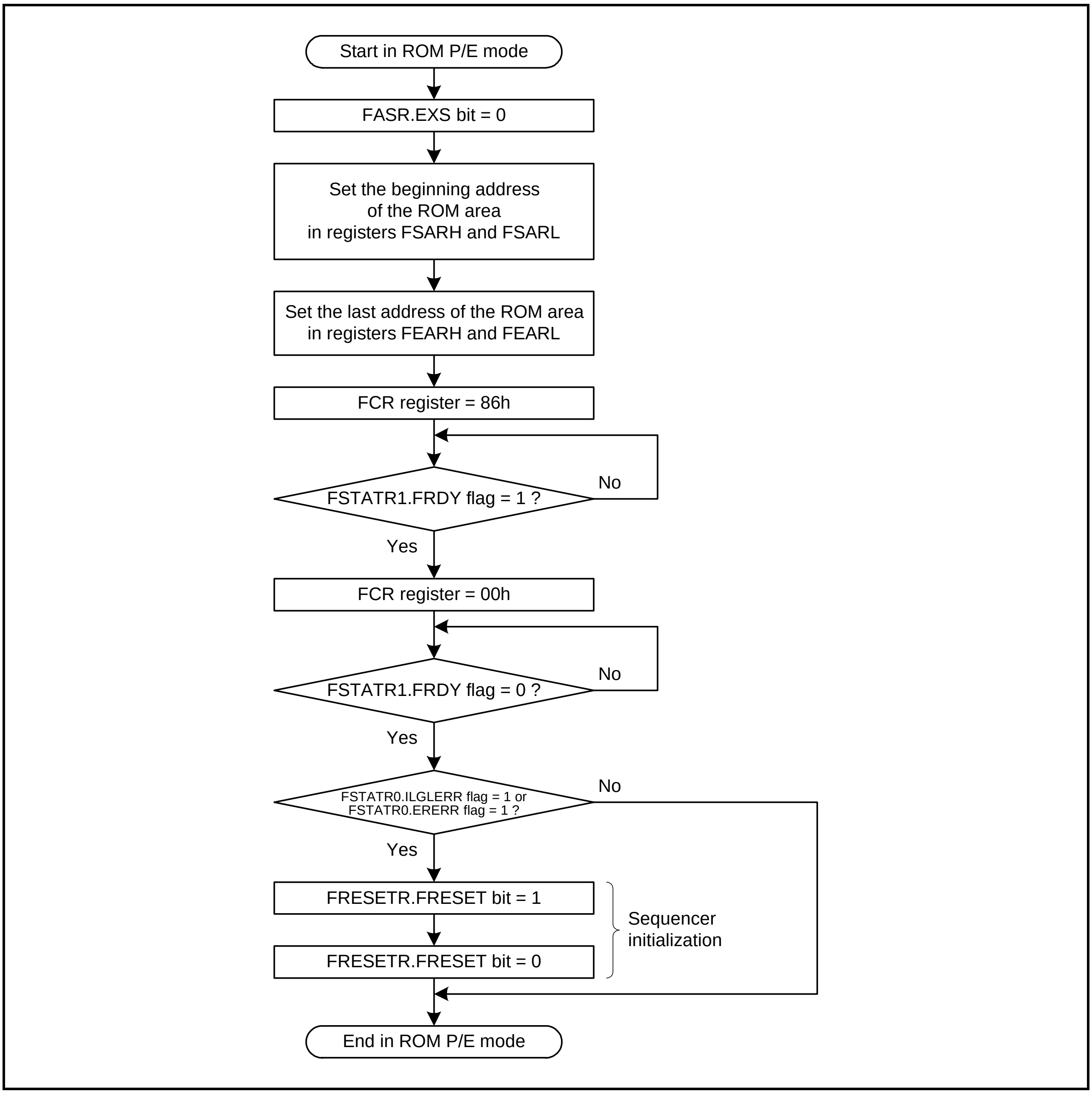


Figure 34.10 Procedure to Issue the All-Block Erase Command

34.6.4.4 Blank Check

Figure 34.11 shows the procedure to issue the blank check command.

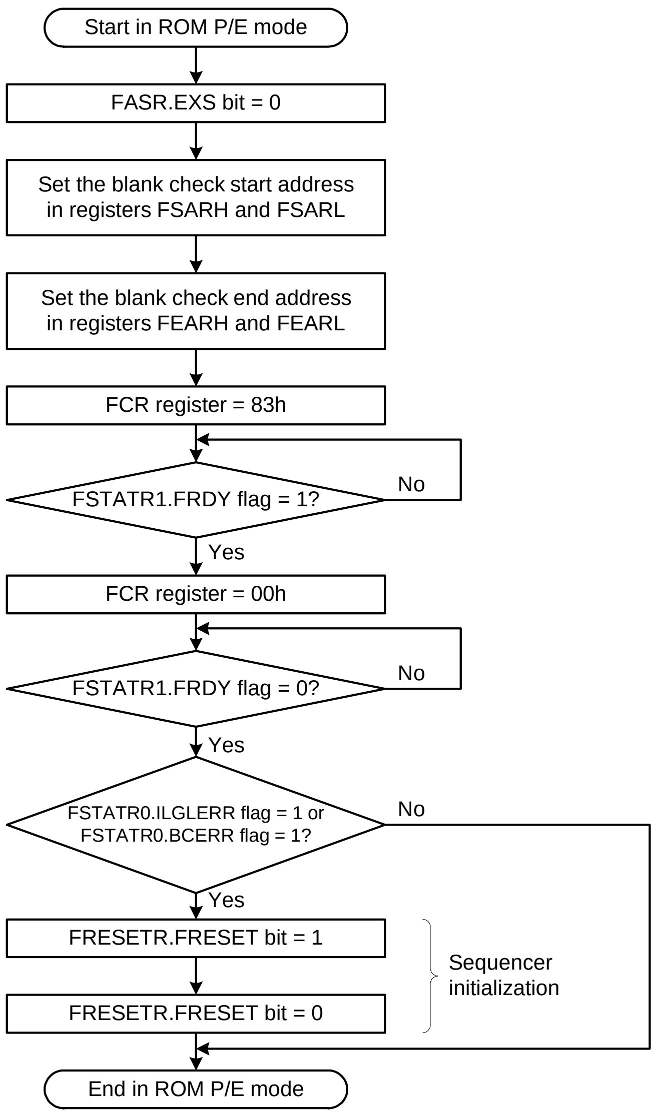


Figure 34.11 Procedure to Issue the Blank Check Command

34.6.4.5 Start-Up Area Information Program/Access Window Information Program

Figure 34.12 shows the procedure to issue the start-up area information program command and access window information program command.

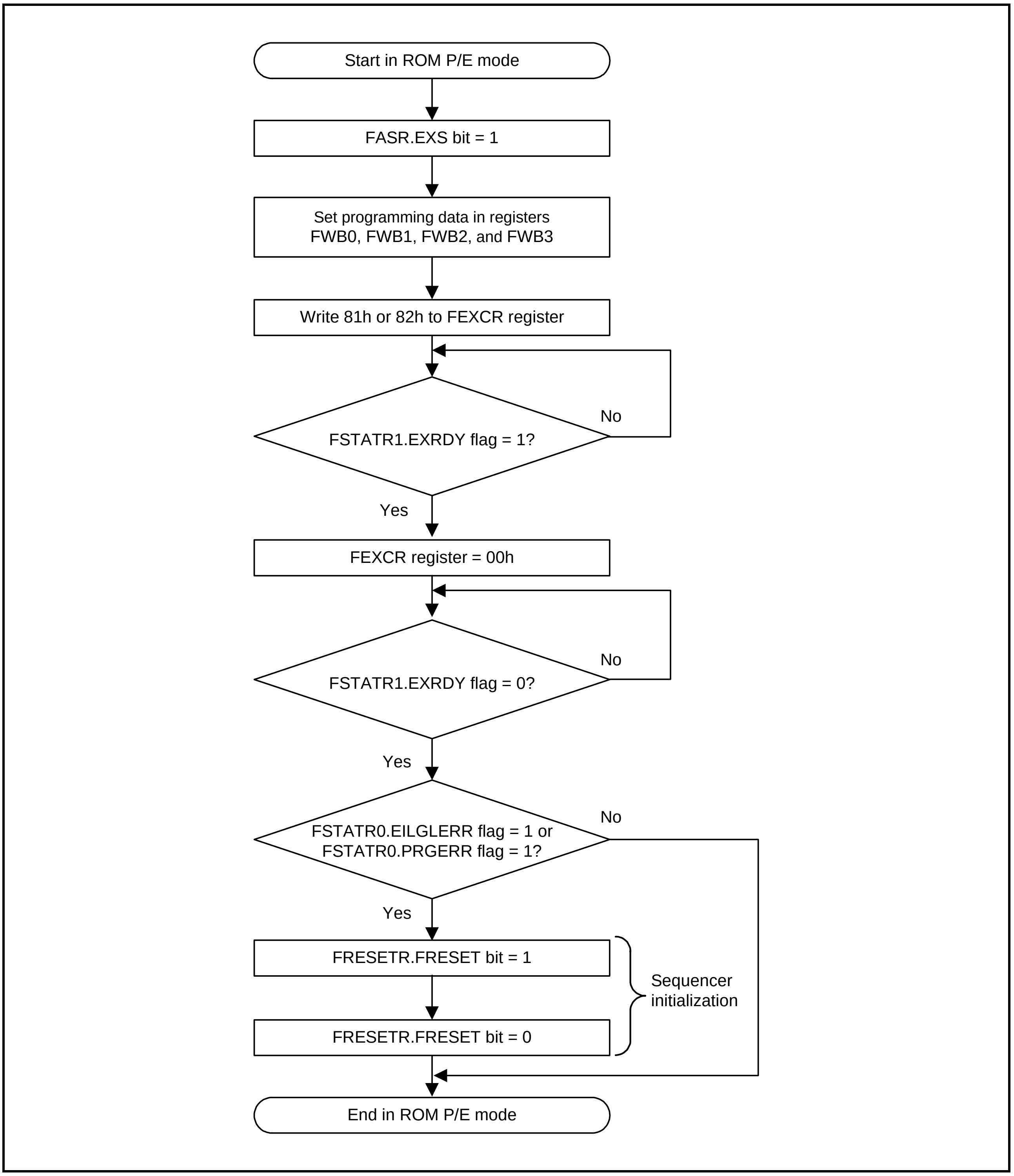


Figure 34.12 Procedure to Issue the Start-Up Area Information Program Command/Access Window Information Program Command

34.6.4.6 Forced Stop of Software Commands

Perform the procedure shown in Figure 34.13 to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

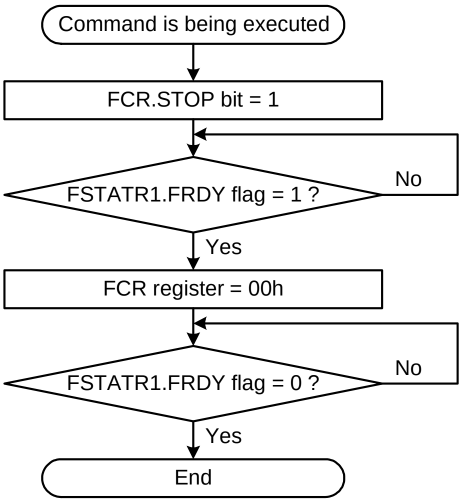


Figure 34.13 Procedure for Forced Stop of Software Commands

34.6.5 Interrupt

When software command processing or forced stop processing is completed, an interrupt (FRDYI) is generated.
When the FSTATR1.FRDY flag becomes 0 by setting the FCR.OPST bit to 0 and the FSTATR1.EXRDY flag becomes 0 by setting the FEXCR.OPST bit to 0, the next interrupt (FRDYI) can be accepted.
Clear the IRn.IR flag before setting the IERm.IEN bit of the ICU corresponding to this interrupt.

34.7 Boot Mode

The SCI or FINE interface is used in boot mode.

Table 34.6 lists the Programmable and Erasable Areas and Peripheral Modules Used in Boot Mode. Table 34.7 lists the I/O Pins Used in Boot Mode.

Table 34.6 Programmable and Erasable Areas and Peripheral Modules Used in Boot Mode

	Boot Mode	
Item	**SCI Interface**	**FINE Interface**
Programmable and erasable areas	User area	User area
Peripheral module	SCI1 (asynchronous serial communication)	FINE

Table 34.7 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode	Description
MD	Input	Boot mode	Select operating mode (refer to section 3, Operating Modes).
MD/FINED	I/O	Boot mode (FINE interface)	Select operating mode, FINE data I/O
PD5/RXD1	Input	Boot mode (SCI)	Receive data*1
PD3/TXD1	Output		Transmit data*1

Note 1. Connect (pull up) this pin to VCC via a resistor.

34.7.1 Boot Mode (SCI)

The flash memory can be programmed and erased using asynchronous serial communication in boot mode (SCI). The user area can be rewritten.
When a reset is released while the MD pin is low, the MCU starts in boot mode (SCI).
Contact the manufacturer for details on the serial programmer.

34.7.1.1 Operating Conditions in Boot Mode (SCI)

SCI1 is used to communicate with the serial programmer in boot mode (SCI).
Figure 34.14 shows an Example of Pin Connections in Boot Mode (SCI). Table 34.8 lists Pin Handling in Boot Mode (SCI).
The examples of pin connections shown in Figure 34.14 are simplified circuits. Operations are not guaranteed in all systems.

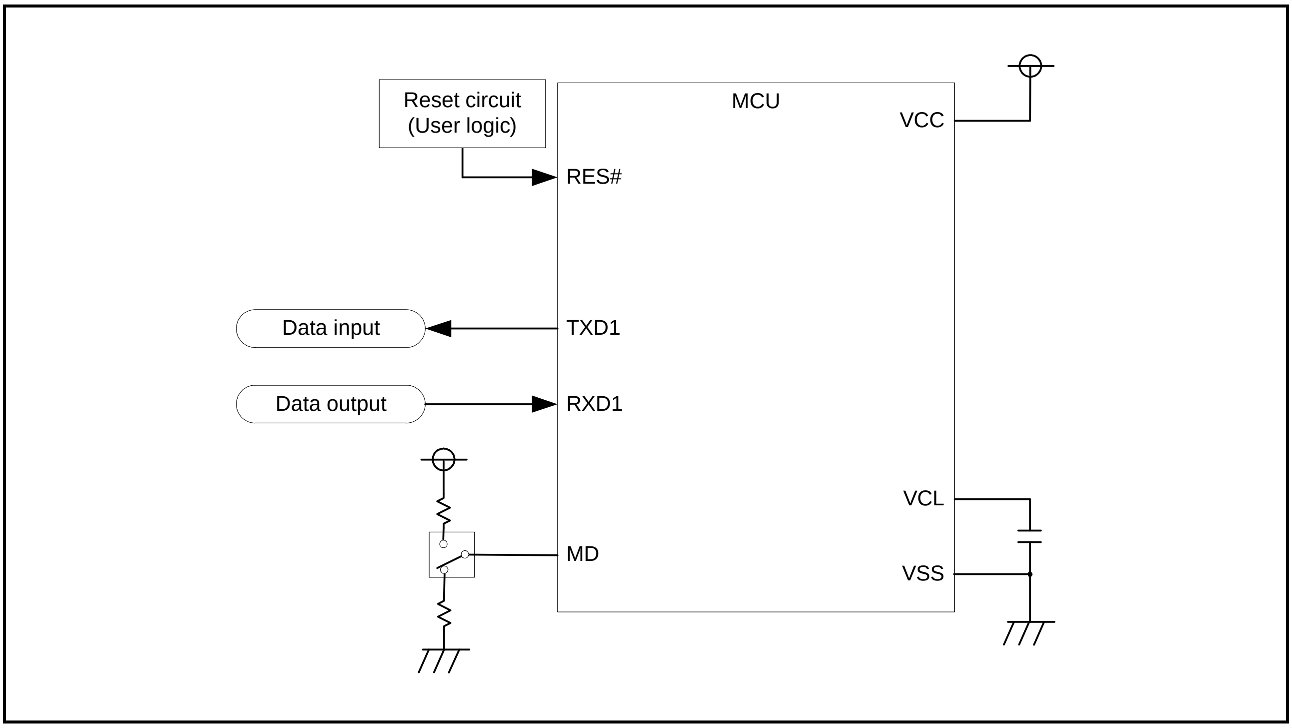


Figure 34.14 Example of Pin Connections in Boot Mode (SCI)

Table 34.8 Pin Handling in Boot Mode (SCI)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD	Operating mode control	Input	Input low.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
PD5/RXD1	Data input RXD	Input	Input pin for serial data
PD3/TXD1	Data output TXD	Output	Output pin for serial data

As shown in Figure 34.15, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the serial programmer.

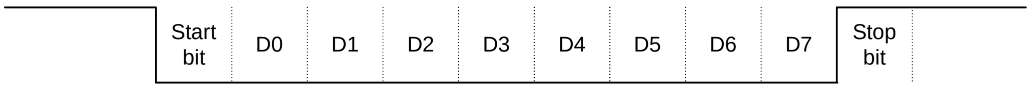


Figure 34.15 Communication Format

Communication with the programmer is performed at 9,600 or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.
Table 34.9 lists the maximum communication bit rates for communication in boot mode (SCI).

Table 34.9 Conditions for Communication

Operating Voltage	Maximum Communication Bit Rate
2.7 V or higher, and lower than 3.0 V	500 kbps
3.0 V or higher, and 5.5 V or lower	2 Mbps

34.7.1.2 Starting Up in Boot Mode (SCI)

To start up in boot mode (SCI), release the reset (drive the RES# pin high from low) while the MD pin is low. After starting up in boot mode (SCI), wait at least 400 ms holding the RES# pin high until communication is enabled in boot mode (SCI).
As shown in Figure 34.16, keep the signal of each pin unchanged for 400 ms after the reset is released. Use resets according to the range described in section 35.3.2, Reset Timing.

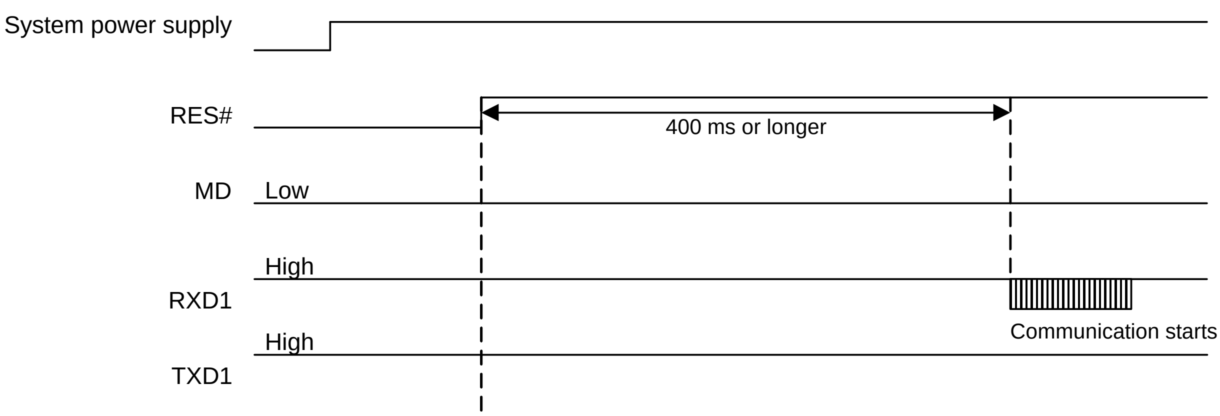


Figure 34.16 Pin States until Communication Becomes Possible in Boot Mode (SCI)

34.7.2 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The user area can be rewritten.
Contact the manufacturer for details on the serial programmer.

34.7.2.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).
Figure 34.17 shows an Example of Pin Connections in Boot Mode (FINE Interface). Table 34.10 lists Pin Handling in Boot Mode (FINE Interface).
The example of pin connections shown in Figure 34.17 is a simplified circuit. Operations are not guaranteed in all systems.

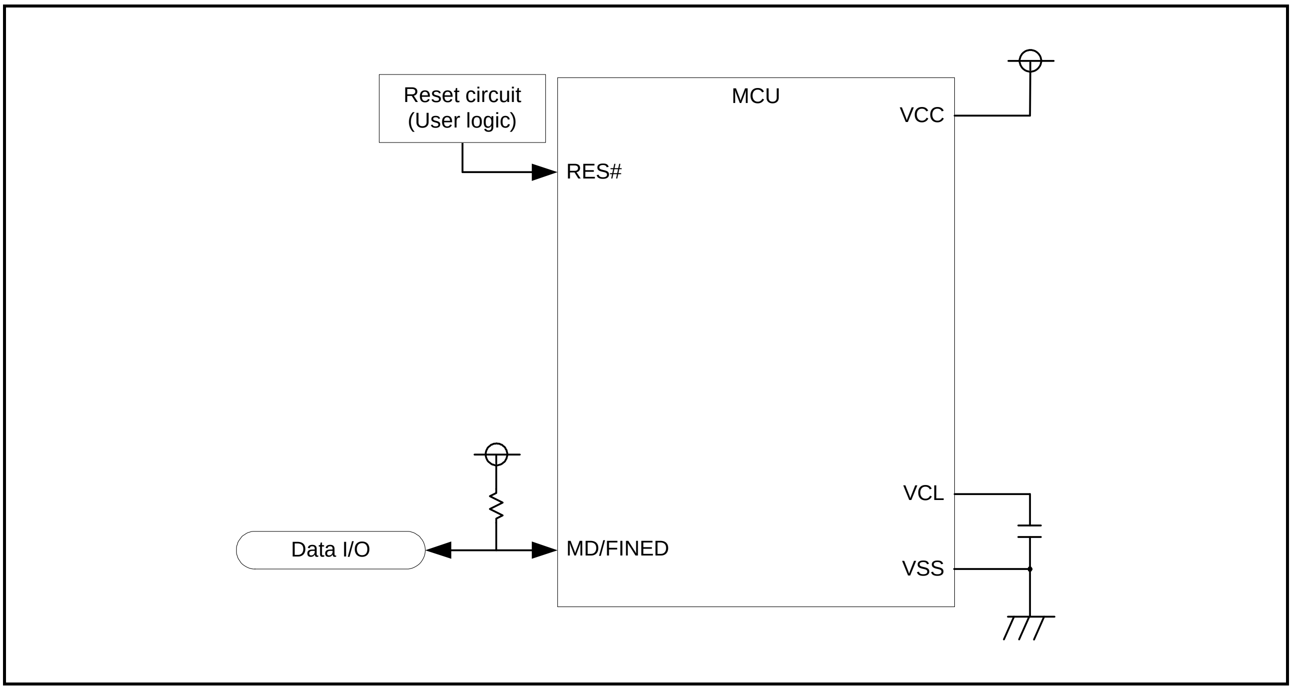


Figure 34.17 Example of Pin Connections in Boot Mode (FINE Interface)

Table 34.10 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD/FINED	Operating mode control/ data I/O	I/O	Connect the VSS pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

34.8 Flash Memory Protection

Flash memory protection prevents the flash memory from being read or rewritten by the third party.
Boot mode ID code protection is for connecting the serial programmer, and on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator.

34.8.1 ID Code Protection

There are two types of ID code protection: Boot mode ID code protection for connecting the serial programmer and on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator. The same ID codes are used for both functions, but operations differ.
ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes to 32-bit 4-word data in 32-bit units.
Figure 34.18 shows the ID Code Configuration.

	31 – 24	23 – 16	15 – 8	7 – 0
FFFF FFA0h	Control code	ID code 1	ID code 2	ID code 3
FFFF FFA4h	ID code 4	ID code 5	ID code 6	ID code 7
FFFF FFA8h	ID code 8	ID code 9	ID code 10	ID code 11
FFFF FFACh	ID code 12	ID code 13	ID code 14	ID code 15

Figure 34.18 ID Code Configuration

The following shows a program example for setting ID codes.
This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE [4] = {0x45010203, 0x04050607,0x08090A0B, 0x0C0D0E0F};
```

Assembly language:

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

34.8.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area when the serial programmer is connected by the third party.
When the control code indicates 45h or 52h (boot mode ID code protection is enabled), the MCU compares 16-byte ID code sent from the serial programmer with the ID code in the user area. According to the comparison result, reading and programming the user area is enabled.
When the control code indicates a value other than 45h and 52h (boot mode ID code protection is disabled), all blocks in the user area are erased, and reading and programming the user area are enabled.

The control code is used to enable or disable protection. Table 34.11 lists the specifications of boot mode ID code protection, and Figure 34.19 shows the authentication flow of boot mode ID code protection.
ID code 1 to ID code 15 can be set to any desired value.
However, only when disabling connection with the serial programmer, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

Table 34.11 Boot Mode ID Code Protection Specifications

ID Code				
Control Code	**ID Code 1 to ID Code 15**	**Protection**	**ID Code Matching Result**	**Operation**
45h	Any desired value	Enabled	Matched	Exit the boot mode ID code authentication state and enter the program/erase host command wait state.
			Not matched	Continue the boot mode ID code authentication state.
			Not matched three times consecutively	Erase all blocks in the user area and continue boot mode ID code authentication state.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh (8 byes are all FFh)	Enabled	N/A	Disable reading or rewriting of the flash memory, regardless of the codes sent from the serial programmer.
	Other than above		Matched	Exit the boot mode ID code authentication state and enter the program/erase state.
			Not matched	Continue the boot mode ID code authentication state.
Other than above	Any desired value	Disabled	N/A	Erase all blocks in the user area.

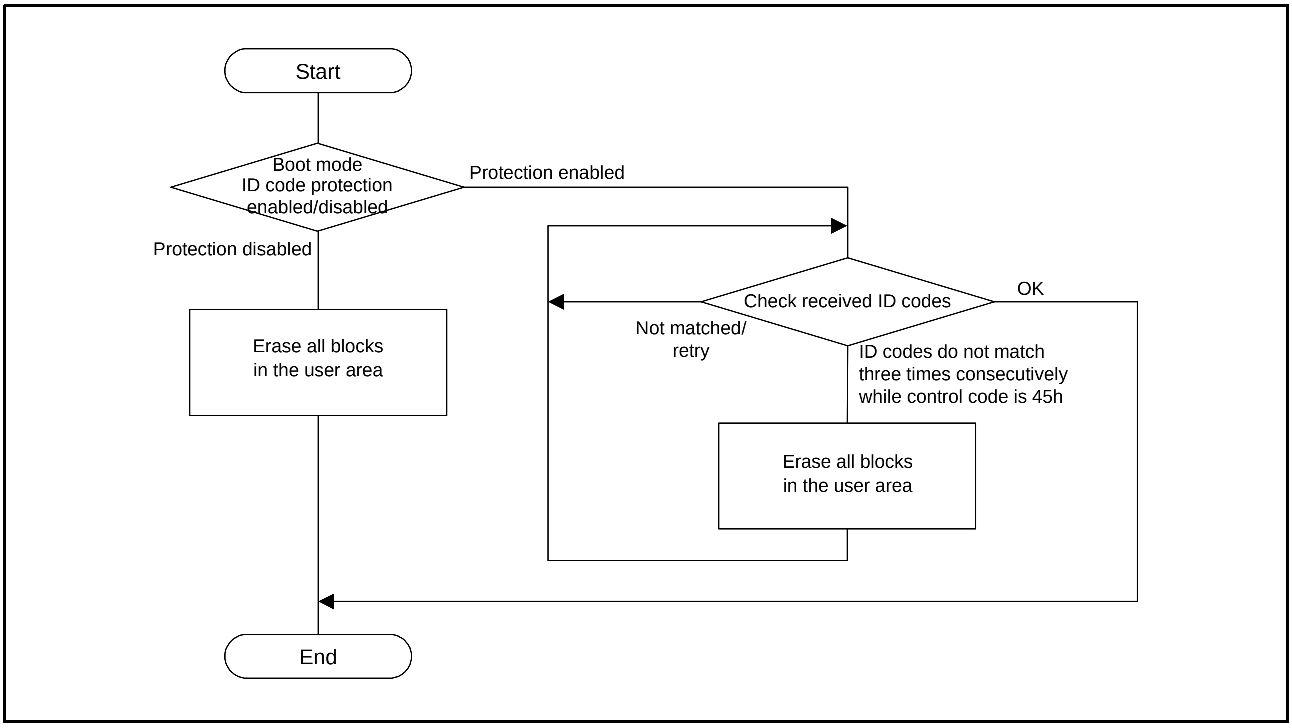


Figure 34.19 Authentication for Boot Mode ID Code Protection

34.8.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When 16-byte ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled.

Table 34.12 lists the specifications of on-chip debugging emulator ID code protection.

Table 34.12 On-Chip Debugging Emulator ID Code Protection Specifications

ID Code				
Control Code	**ID Code 1 to ID Code 15**	**Protection**	**ID Code Matching Result**	**Operation**
FFh	FFh,…, and FFh (15 byes are all FFh)	Disabled	N/A	Enable connection with the on-chip debugging emulator.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes	Enabled	N/A	Disable connection with the on-chip debugging emulator, regardless of the codes sent from the on-chip debugging emulator.
Other than above	Other than above	Enabled	Matched	Enable connection with the on-chip debugging emulator.
			Not matched	Continue the ID code wait state.

34.9 Communication Protocol

This section describes the protocol used in boot mode. When developing a serial programmer, control with this communication protocol.

34.9.1 State Transition in Boot Mode (SCI)

Figure 34.20 shows the Boot Mode (SCI) State Transition.

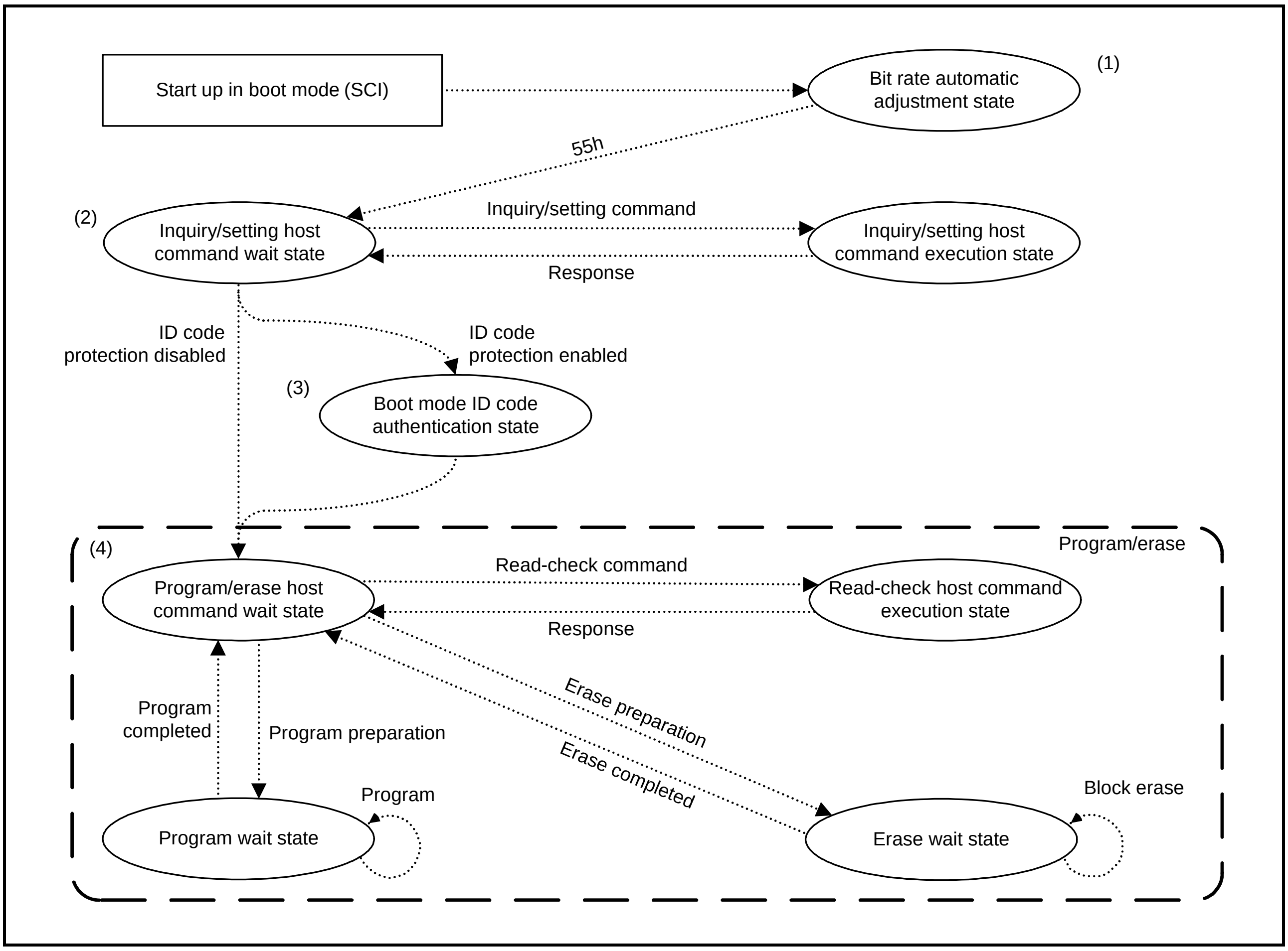


Figure 34.20 Boot Mode (SCI) State Transition

(1) Bit rate automatic adjustment state
In this state, the bit rate is automatically adjusted to 9,600 or 19,200 bps for communication with the host.
When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and enters the inquiry/setting host command wait state.
The host must not send data until 400 ms elapse after a reset of the MCU is released.

(2) Inquiry/setting host command wait state
In this state, the host can make inquiries for the MCU information including block configuration, size, and addresses where the user area is allocated, and select the endian of data and a bit rate.
When the MCU receives the program/erase host state transition command from the host, it determines whether boot mode ID code protection is enabled or disabled. If boot mode ID code protection is disabled, the MCU enters the inquiry/setting host command wait state. If boot mode ID code protection is enabled, the MCU enters the boot mode ID code authentication state.

Refer to section 34.9.5, Inquiry Commands and section 34.9.6, Setting Commands for details on inquiry/ setting commands.

(3) Boot mode ID code authentication state
In this state, the MCU accepts the ID code authentication command.
If boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.
Refer to section 34.8.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 34.9.7, ID Code Authentication Command for details on the ID code authentication command.

(4) Program/erase state
In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.
Refer to section 34.9.8, Program/Erase Commands for details on program/erase commands. Refer to section 34.9.9, Read-Check Commands for details on read-check commands.

34.9.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host.
Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.
A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.
“Size” indicates the number of transmit/receive data bytes excluding Command (the first byte), Size, and SUM.
“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.
The flash memory addresses for reading are used as the following addresses: the program address selected in the program command, the block start address selected in the block erase command, and the AW start and end addresses selected in the access window information program command, and the AW start and end addresses received in the access window read command.

34.9.3 Response to Undefined Commands

When the MCU receives an undefined command, it sends a command error as a response. The contents of the response are shown below. The command in the error response stores the first byte of the command sent from the MCU.

Error response	80h	Command

34.9.4 Boot Mode Status Inquiry

This command is used to check the current state and the previous error of the boot program.
The MCU returns a code from Table 34.13 and Table 34.14 as the current state and the previous error.
The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase state.

Command	4Fh				
Response	5Fh	Size	State	Error	SUM

Size (1 byte): Total bytes of “State” and “Error” (the value is always 02h)
State (1 byte): MCU's current state (see Table 34.13)
Error (1 byte): Information about the error occurred in the MCU (see Table 34.14)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Table 34.13 Information Regarding the States

Code	State*1	Description
11h	Inquiry/setting host command wait state	Device selection wait state
12h/13h		Operating frequency selection wait state
1Fh		Program/erase state transition command wait state
31h	Boot mode ID code authentication state	The user area is being erased
3Fh	Program/erase host command wait state	Program/erase command wait state
4Fh		Program data reception wait state
5Fh		Block erase specification wait state

Note 1. Refer to Figure 34.20 for details on the state transitions.

Table 34.14 Error Information

Code	Description
00h	No error
11h	SUM error
21h	Device code error
24h	Bit rate selection error
29h	Block start address error
2Ah	Address error
2Bh	Data length error
51h	Erase error
52h	Not blank (blank check error)
53h	Program error
61h	ID code do not match
63h	ID code do not match and erase error
80h	Command error
FFh	Bit rate automatic adjustment error

34.9.5 Inquiry Commands

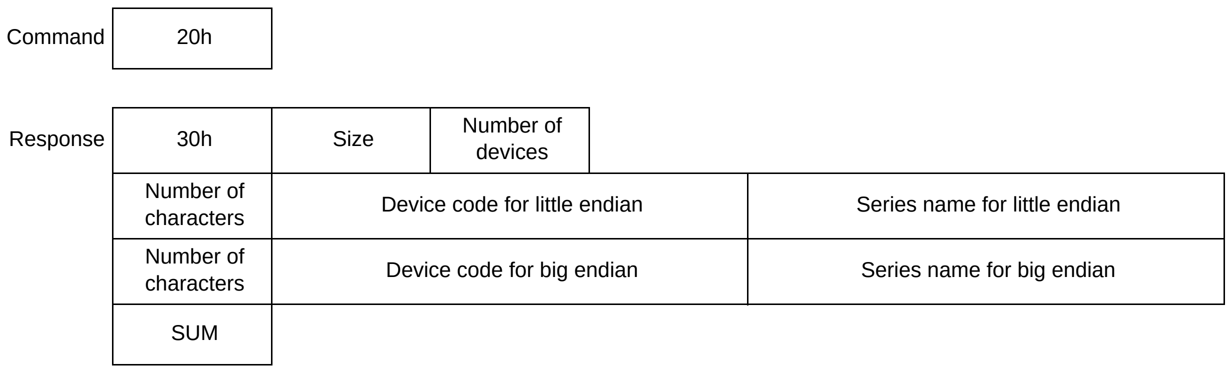
Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 34.15 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

Table 34.15 Inquiry Commands

Command	Description
Supported device inquiry	Inquiry for the device code and series name
Data area availability inquiry	Inquiry for the availability of the data area
User area information inquiry	Inquiry for the number of user areas, and the start and end addresses of the user area
Block information inquiry	Inquiry for the start and end addresses of the user areas, the block size, and the number of blocks

34.9.5.1 Supported Device Inquiry

This command is used to obtain the device information for identifying the endian of developed software.
After the MCU receives this command, it sends the device information when developed software uses little endian data and the device information when developed software uses big endian data in this order.



Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name
Number of devices (1 byte): Number of endian types of program data (the value is always 02h)
Number of characters (1 byte): Number of characters for the device code and device name
Device code (4 bytes): Identification code indicating the endian of developed software
Series name (n bytes): Little endian/big endian (ASCII code) of the series name of the MCU
SUM (1 byte): Value that is calculated so the sum of response data is 00h

34.9.5.2 Data Area Availability Inquiry

When the MCU receives this command, it sends the result indicating the data area is not available, area protection can be used, and data area program command is not available.

Command	2Ah

Response	3Ah	Size	Availability	SUM

Size (1 byte): Number of characters of Availability (the value is always 01h)
Availability (1 byte): Availability of the data area (the value is always 1Dh)
1Dh represents the data area is not available, area protection can be used, data area program command is not available.
SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always A8h)

34.9.5.3 User Area Information Inquiry

When the MCU receives this command, it sends the number of user areas and addresses.

Command	25h

Response			
	35h	Size	Number of areas
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of user areas (the value is always 01h)
Area start address (4 bytes): Start address of the user area
Area end address (4 bytes): End address of the user area
SUM (1 byte): Value that is calculated so the sum of the response data is 00h

34.9.5.4 Block Information Inquiry

When the MCU receives this command, it sends the start address, the size of one block, and the number of blocks in the user area.

Command	26h

Response			
	36h	Size	DDh
	Start address of the user area		
	Block size of one block for the user area		
	Number of blocks of the user area		
	Dummy data		
	Dummy data		

Dummy data	
SUM	

Size (2 bytes): Total bytes of data from DDh to Number of blocks of the user area (the value is always 00 19h)
Start address of the user area (4 bytes): Start address of the user area
Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 08 00h)
Number of blocks of the user area (4 bytes): Number of blocks in the user area
Dummy data (12 bytes): Dummy data
SUM (1 byte): Value that is calculated so the sum of response data is 00h

34.9.6 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU.
Table 34.16 lists Setting Commands. These commands can be used only in the inquiry/setting host command wait state.

Table 34.16 Setting Commands

Command	Function
Device select	Select a device code.
Operating frequency select	Change the bit rate for communication.
Program/erase host command wait state transition	Enter the program/erase host command wait state or boot mode ID code authentication state.

34.9.6.1 Device Select

This command is used to specify the endian of developed software. Select a device code from among the device codes obtained in the response to the support device inquiry command.
If the received device code matches the supported device, the MCU sends a response (46h).
If the device is not supported or the SUM of the received command does not match, the MCU sends an error response.

Command	10h	Size	Device code	SUM

Size (1 byte): Number of characters of the device code (the value is always 04h)
Device code (4 bytes): Identification code indicating the device
(code in the response to the support device inquiry command)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	46h

Error response	90h	Error

Error (1 byte): Error code
11h: SUM error
21h: Device code error

34.9.6.2 Operating Frequency Select

This command is used to specify the operating frequency of the MCU and a bit rate for communication with the flash memory programmer. The bit rate selected in this command should be set to a value with error of less than 4% compared to the bit rate obtained by dividing 32 or 8 MHz that corresponds to the operating voltage.
If the specified settings can be supported, the MCU sends a response (06h). If the bit rate error is 4% or more or the SUM of the received command does not match, the MCU sends an error response.

After the host receives a response, wait for at least a 1-bit period at the old bit rate, and send communication confirmation data at the new bit rate.
If the MCU successfully receives communication confirmation data, the MCU sends a response (06h). If the MCU fails to receive the communication confirmation data, the MCU sends an error response.

Command	3Fh	Size	Bit rate	Dummy data
	Number of clocks	Multiplier 1	Multiplier 2	
	SUM			

Size (1 byte): Total bytes of data of Bit rate, Dummy data, Number of clocks, and Multiplier (the value is always 07h)
Bit rate (2 bytes): New bit rate (e.g. 00C0h: 19,200 bps)
The value is calculated by dividing the bit rate by 100 (Example: Set 00C0h for 19200 bps)
Dummy data (2 bytes): The value should always be set to 0000h
Number of clocks (1 byte): Types of clocks for multiplier setting (the value is always 02h)
Multiplier 1 (1 byte): Multiplier of the system clock (ICLK) (the value is always 01h)
Multiplier 2 (1 byte): Multiplier of the peripheral module clock (PCLK) (the value is always 01h)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h

Error response	BFh	Error

Error (1 byte): Error code
11h: SUM error
24h: Bit rate selection error

Communication confirmation	06h

Response	06h

Error response	FFh

- Bit rate selection error
  A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the new bit rate specified with the operating frequency select command is B, and 32 (MHz) or 8 (MHz) corresponding to the operating voltage is Pφ, the bit rate error is calculated by the following formula:

$$\text{Error [\%]} = \left(\frac{P\varphi \times 10^6}{B \times 32 \times N} - 1\right) \times 100$$

$$N = \text{INT}\left(\frac{P\varphi \times 10^6}{B \times 32}\right)$$

Pφ: 32 (MHz) when the operating voltage is 3.0 V or above
    8 (MHz) when the operating voltage is below 3.0 V
B: New bit rate (bps)
N: Ratio between Pφ and the new bit rate multiplied by 32 (however, $1 \le N \le 256$)

34.9.6.3 Program/Erase Host Command Wait State Transition

This command is used for the transition from the inquiry/setting host command wait state to the program/erase host command wait state.
When the MCU receives this command, it determines whether boot mode ID code protection is enabled or disabled.

When boot mode ID code protection is disabled, all blocks in the user area is erased.
When all blocks are successfully erased, the MCU sends a response (06h) and enters the program/erase host command wait state. If not all blocks are successfully erased, the MCU sends an error response.
When boot mode ID code protection is enabled, the MCU sends a response (16h) and enters boot mode ID code authentication state.

Command	40h

Response	ACK

ACK (1 byte): ACK code
06h: ID code protection is disabled.
16h: ID code protection is enabled.

Error response	C0h	Error

Error (1 byte): Error code
51h: Erase error

34.9.7 ID Code Authentication Command

This command is used for ID code authentication when boot mode ID code protection is enabled.
Table 34.17 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

Table 34.17 ID Code Authentication Command

Command	Function
ID code check	Compare the 16-byte code sent from the host and ID code.

34.9.7.1 ID Code Check

This command is used to unlock boot mode ID code protection.
The comparison ID code specified with the command should be set to the same value as the control code and ID code 1 to ID code 15.
If the comparison ID code sent from the host matches the ID code programmed in the user area, the MCU sends a response (06h) and enters program/erase host command wait state.
If the codes do not match or the SUM of the received command does not match, the MUC sends an error response.
When the ID codes do not match three times consecutively while the control code is 45h, all blocks in the user area is erased. If an error occurs during erasure, the MUC sends an error response.
Also, even if all blocks are successfully erased, the MCU sends an error response and continues the boot mode ID code state. Reset the MCU to enter the program/erase host command wait state.

Command

60h	Size
Comparison ID code (control code + ID code 1 to ID code 15)	
SUM	

Size (1 byte): Number of bytes of ID codes (the value is always 10h)
ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)
SUM (1 byte): Value that is calculated so the sum of the command data is 00h

Response

ACK

ACK (1 byte): ACK code
06h: The MCU enters the program/erase host command wait state.

E0h	Error

Error (1 byte): Error code
11h: SUM error
61h: ID codes do not match
63h: ID codes do not match and erase error

34.9.8 Program/Erase Commands

Program/erase commands are used to program or erase the user area based on the response to inquiry commands. Table 34.18 lists commands used in the program/erase command wait state, program wait state, and erase wait state. Table 34.19 lists commands that can be accepted in each state.
When a command that cannot be accepted is received in the state listed in Table 34.19, the MCU sends a command error response.

Table 34.18 Program/Erase Commands

Command	Function
User area program preparation	Select the user area to program, and enter the program wait state.
Program	Program the specified data to the selected area in the user area. Or enter the program/erase host command wait state (end of program).
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program/erase host command wait state (end of erase).

Table 34.19 Acceptable Commands for Each State

State	Acceptable Command
Program/erase host command wait state	User area program preparation command, erase preparation command
Program wait state	Program command
Erase wait state	Block erase command

34.9.8.1 User Area Program Preparation

This command is used to prepare for accepting the program command.
When the MCU receives this command, it recognizes that an instruction to prepare for the program command is issued from the host. Then, the MCU enters the program wait state, where only the program command to the user area can be accepted, and sends a response (06h).

Command	43h
Response	06h

34.9.8.2 Program

This command is used to program the specified data to the user area. Set the low-order 8 bits to 0 for the program address selected in this command. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 50h FFh FFh FFh FFh B4h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.

Command	50h	Program address
	Program data	
	SUM	

Program address (4 bytes): Address for program destination
Set the low-order 8 bits to 0
Set FFFF FFFFh for end of program
Program data (n bytes): Program data (n = 256 in boot mode, 0 for end of program)
When the program is less than n bytes, set FFh for the missing data.
No program data for the end of program
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h

Error response	D0h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
53h: Program error (the data or program data cannot be programmed.)

34.9.8.3 Erase Preparation

This command is used to prepare for accepting the block erase command. When the MCU receives this command, it recognizes that an instruction to prepare for the erase command is issued from the host. Then, the MCU enters the erase wait state, where only the block erase command can be accepted, and sends a response (06h).

Command	48h
Response	06h

34.9.8.4 Block Erase

This command is used to erase the selected block in the user area. Specify the block start address selected in the command by calculating the address based on the response to the block information inquiry command.
When the selected block in the block start address is successfully erased, the MCU sends an error response (06h). If the SUM of the received command does not match or an error occurs during an erase operation, the MCU sends an error response.

To enter the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response (06h).

Command	59h	Size
	Block start address	
Response	SUM	

Size (1 byte): Total bytes of Block start address (the value is always 04h)
Block start address (4 bytes): Start address of the block that is erased
Set FFFF FFFFh for end of erase
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	D9h	Error

Error (1 byte): Error code
11h: SUM error
29h: Block start address error
51h: Erase error (the selected block cannot be erased)

34.9.9 Read-Check Commands

Read-check commands are used to read data or check whether data is programmed in the user area in the MCU based on the response to inquiry commands.
Table 34.20 lists read-check commands used in the program/erase host command wait state.

Table 34.20 Read-Check Commands

Command	Function
Memory read	Read data from the user area.
User area checksum	Obtain the checksum of the entire user area.
User area blank check	Check whether data is programmed in the user area.
Access window information program	Set the access window.
Access window read	Read the settings of the access window.

34.9.9.1 Memory Read

This command is used to read data programmed in the user area. For a read start address selected in the command, set a value within the range from the area start address to the area end address received in the response to the user area information inquiry command.
For a read size selected in the command, set a value so the sum of the read start address and the read size is within the range from the start address to the end address received in the response to the user area information inquiry command.
When the MCU performs a read successfully, it sends data of the specified range. If the SUM of the received command does not match or the MCU fails to perform a read successfully, it sends an error response.

Command			
52h	Size	Area	
Read start address			
Read size			
SUM			

Size (1 byte): Total bytes for Read start address and Read size
Area (1 byte): Area that is read
01h: User area
Read start address (4 bytes): Start address of the area that is read
Read size (4 bytes): Size of data that is read (in bytes)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	
52h	Read size
Read data	
SUM	

Read size (4 bytes): Size of Data that is read (in bytes)
Read data (n bytes): Data read from the selected address (n = read size)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	D2h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error
- A value other than 01h is set for area.
- The read start address is not in the selected area.

2Bh: Size error
- The read size is set to 0000 0000h.
- The read size exceeds the area size.
- The address calculated from the read start address and read size is not in the selected area.

34.9.9.2 User Area Checksum

This command used to obtain the checksum of the entire user area. When the MCU receives this command, it adds data from the start address to the end address in bytes, and sends the calculated result (checksum) as a response.

Command	4Bh

Response	5Bh	Size
	User area checksum	
	SUM	

Size (1 byte): Number of bytes for checksum of the user area (the value is always 04h)
User area checksum (4 bytes): Calculated result of the data in the user area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

34.9.9.3 User Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of data, the MCU sends an error response.

Command	4Dh

Response	06h

Error response	CDh	Error

Error (1 byte): Error code
52h: Not blank

34.9.9.4 Access Window Information Program

This command is used to set the access window used for area protection. For the access window start address selected in the command, set the start address of the start block. For the access window end address, set the end address of the end block.

When the selected access window settings are successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during the access window settings, the MCU sends an error response.

For details on the access window, see section 34.5, Area Protection.

Command	74h	05h	Access window	
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	SUM			

Access window (1 byte): Select the access window or clear the access window settings
Set 00h to select the access window
Set FFh to clear the access window settings

Access window start address LH (1 byte): Start address of the access window (A15 to A8)
Set A15 to A8 of the block start address.
Set FFh, FFh to clear the access window settings

Access window start address HL (1 byte): Start address of the access window (A23 to A16)
Set A23 to A16 of the block start address.
Set FFh, FFh to clear the access window settings

Access window end address LH (1 byte): End address of the access window (A15 to A8)
Set A15 to A8 of the block end address.
Set FFh, FFh to clear the access window settings

Access window end address HL (1 byte): End address of the access window (A23 to A16)
Set A23 to A16 of the block end address.
Set FFh, FFh to clear the access window settings

SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h

Error response	F4h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (address is not in the selected area)
53h: Program error (access window cannot be set)

34.9.9.5 Access Window Read

This command is used to check the set range of the access window.

When the MCU successfully obtains the access window range, the MCU sends the access window start address and end address that it read. If the SUM of the received command does not match, the MCU sends an error response.

Command	73h	01h	FFh	8Dh

Response	73h	05h		
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	FFh			
	SUM			

Access window start address LH (1 byte): Start address of the access window range (A15 to A8)
Access window start address HL (1 byte): Start address of the access window range (A23 to A16)
Access window end address LH (1 byte): End address of the access window range (A15 to A8)
Access window end address HL (1 byte): End address of the access window range (A23 to A16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	F3h	Error

Error (1 byte): Error code
11h: SUM error

34.10 Serial Programmer Operation in Boot Mode (SCI)

The following describes the procedure for the serial programmer to program/erase the user area in boot mode (SCI).

1. Automatically adjust the bit rate
2. Receive the MCU information*1
3. Select the device and change the bit rate
4. Enter the program/erase host command wait state
5. Unlock boot mode ID code protection
6. Erase the user area*2, *3
7. Program the user area*2, *3
8. Check data in the user area*2
9. Set the access window in the user area
10. Reset the MCU

Note 1. If the necessary information has been already received, step 2 can be skipped.
Note 2. Any step from 6 to 9 can be skipped, and their order can be changed.
Note 3. When a timeout occurs or invalid response data is received, stop the operation and perform step 10 (reset the MCU).

Refer to section 34.9.5, Inquiry Commands, section 34.9.6, Setting Commands, section 34.9.7, ID Code Authentication Command, section 34.9.8, Program/Erase Commands, and section 34.9.9, Read-Check Commands for details on the commands used in the above steps 2 to 9.

34.10.1 Bit Rate Automatic Adjustment Procedure

The MCU measures the low width of data 00h that is sent from the serial programmer at 9,600 or 19,200 bps to automatically adjust the bit rate.

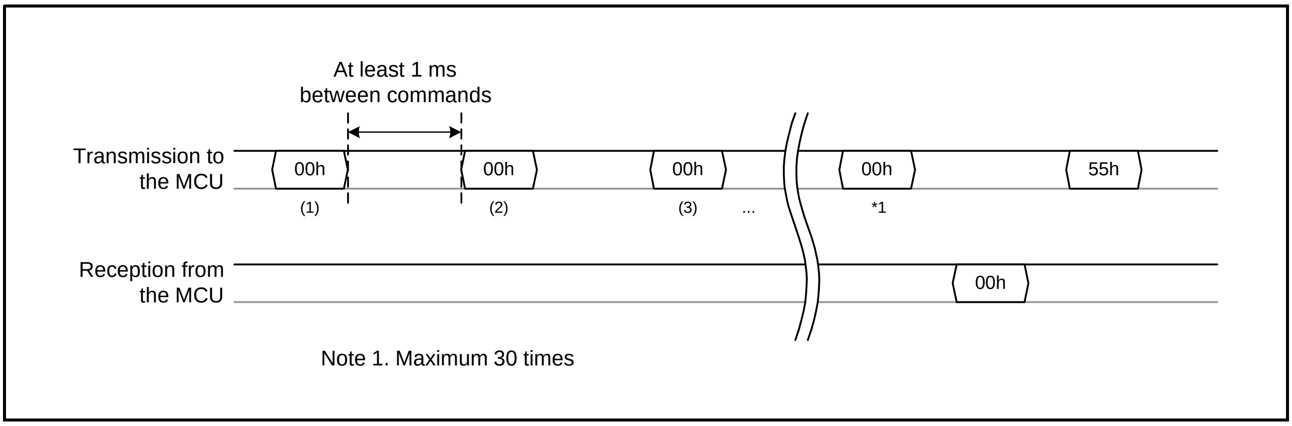


Figure 34.21 Transmit/Receive Data for Bit Rate Automatic Adjustment

After starting up in boot mode, wait for at least 400 ms and then send 00h to the MCU from the serial programmer. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer can not receive 00h, wait for at least 1 ms and send 00h to the MCU again. When the programmer fails to receive 00h even if it send 00h 30 times, restart the MCU in boot mode and adjust the bit rate again.
When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. If the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and adjust the bit rate again.

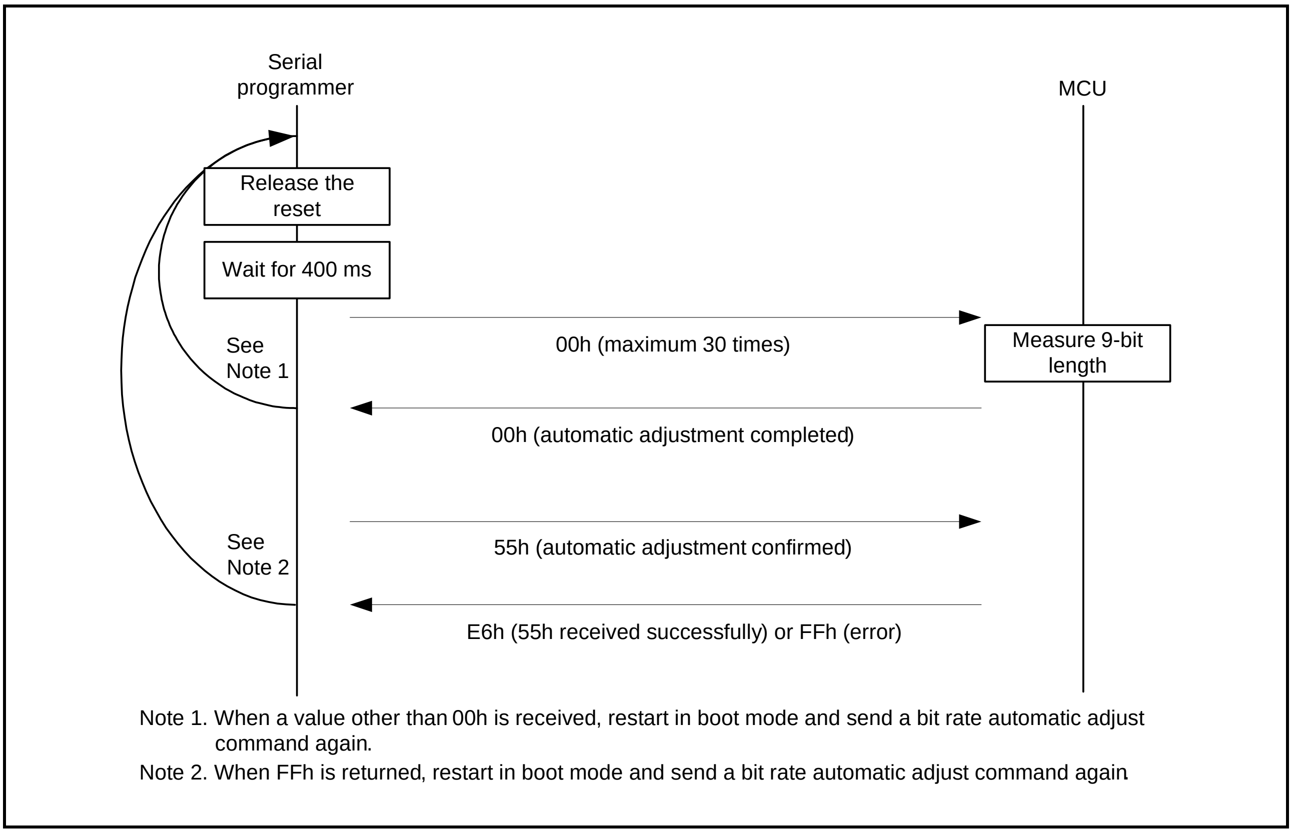


Figure 34.22 Bit Rate Automatic Adjustment Procedure

34.10.2 Procedure to Receive the MCU Information

Send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands.

(1) Send a support device inquiry command (20h) to check which device to connect. The MCU returns the device code and series name.
(2) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
(3) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area.

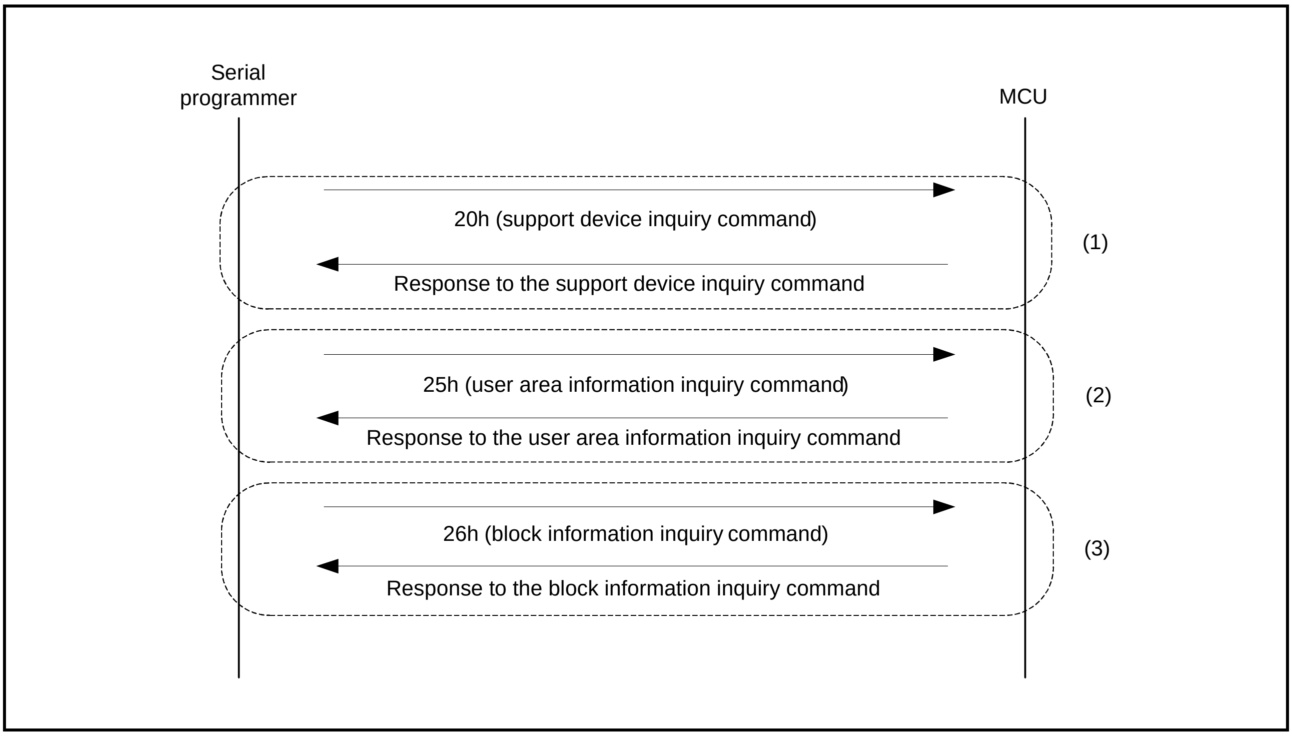


Figure 34.23 Procedure to Receive the MCU Information

34.10.3 Procedure to Select the Device and Change the Bit Rate

Select the device to connect with the serial programmer and change the bit rate for communication.

(1) Send the device select command (10h). Select the device code according to the endian of developed software.

(2) Send the operating frequency select command (3Fh) to change the communication bit rate from 9,600 or 19,200 bps.

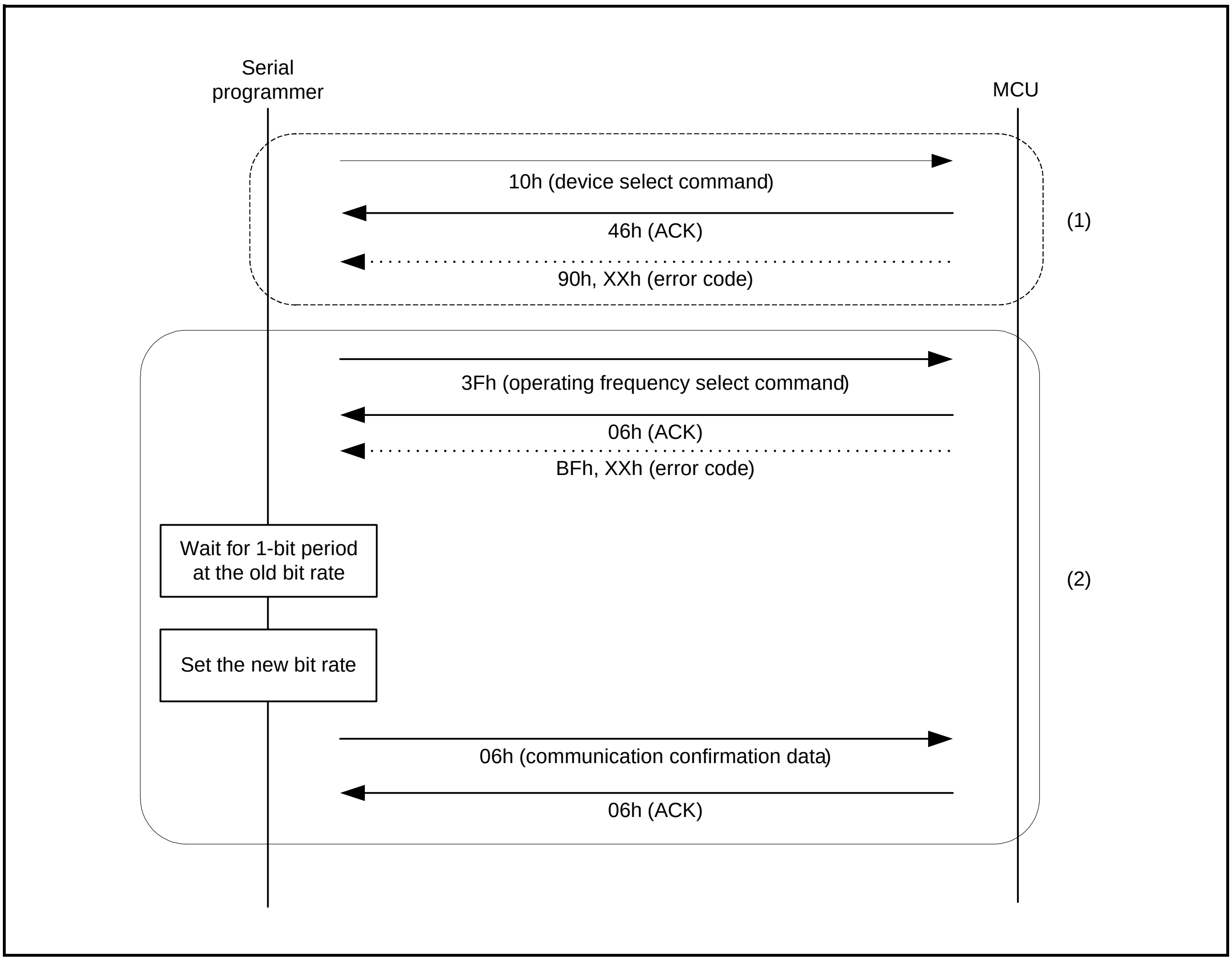


Figure 34.24 Procedure to Select the Device and Change the Bit Rate

34.10.4 Transition to the Program/Erase Host Command Wait State

Send the program/erase host command wait state transition command to perform program/erase operations. The MCU sends a response according to whether boot mode ID code protection is enabled or disabled.

(1) When boot mode ID code protection is disabled, the MCU sends a response (06h), and enters the program/erase host command wait state. Use the serial programmer to start from the operation described in section 34.10.6, Erase the User Area.

(2) When the boot mode ID code protection is enabled, the MCU sends a response (16h), and enters the ID code authentication wait state. Use the serial programmer to start from the operation described in section 34.10.5, Unlock Boot Mode ID Code Protection.

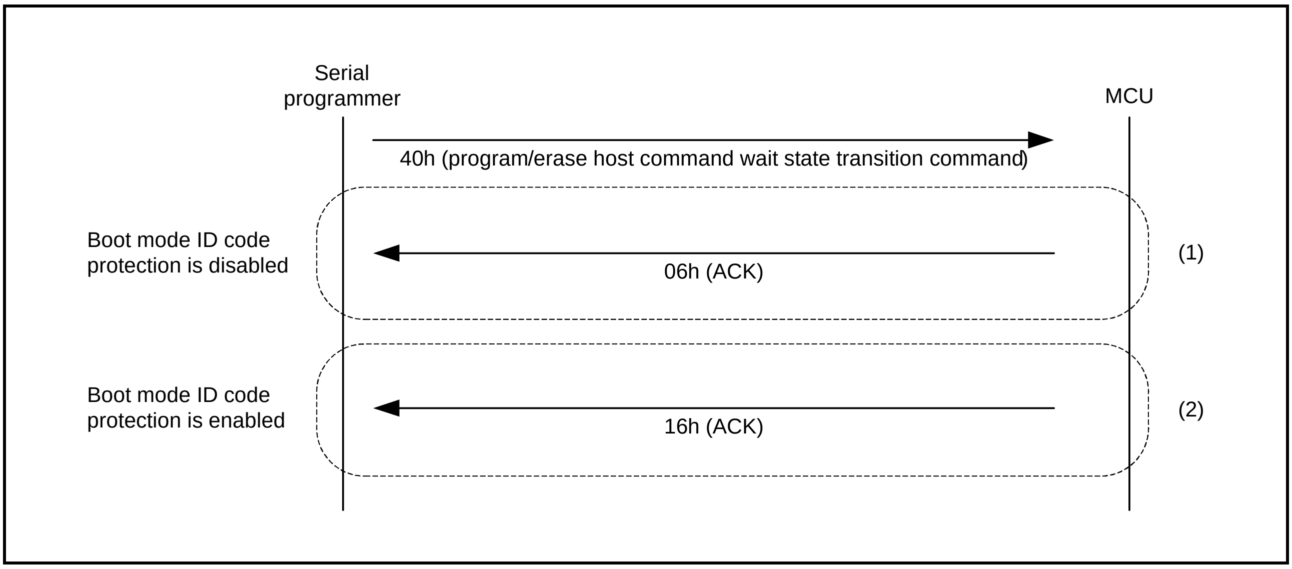


Figure 34.25 Procedure to Enter the Program/Erase Host Command Wait State

34.10.5 Unlock Boot Mode ID Code Protection

Send the ID code check command to unlock boot mode ID code protection.

(1) When ID codes match, the MCU enters the program/erase host command wait state. Data in the user area is not erased. Use the serial programmer to start from the operation described in section 34.10.6, Erase the User Area.

(2) If ID codes do not match consecutively, the MCU remains in the boot mode ID code authentication state. Reset the MCU, and then use the serial programmer to start again from section 34.10.1, Bit Rate Automatic Adjustment Procedure.

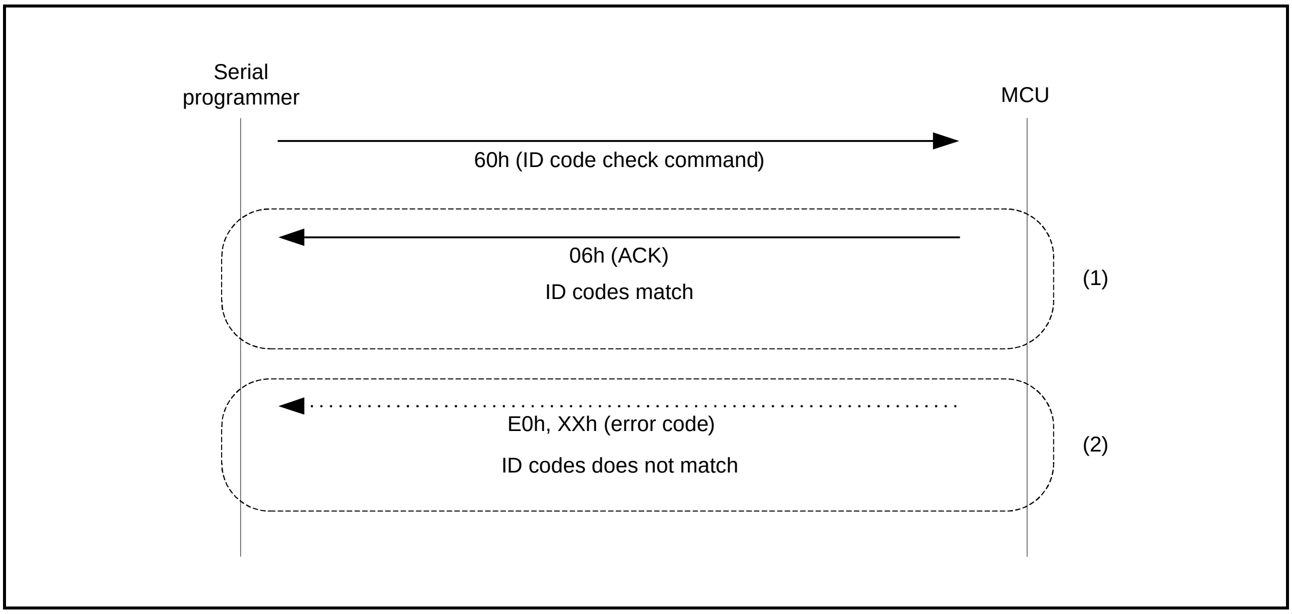


Figure 34.26 Procedure to Unlock ID Code Protection

34.10.6 Erase the User Area

Erase blocks that are programmed in the user area to program a user program.

(1) Send an erase preparation command (48h).
(2) Send a block erase command (59h).
(3) To place the MCU in the program/erase host command wait state, send a block erase command for end of erase (59h 04h FFh FFh FFh FFh A7h).

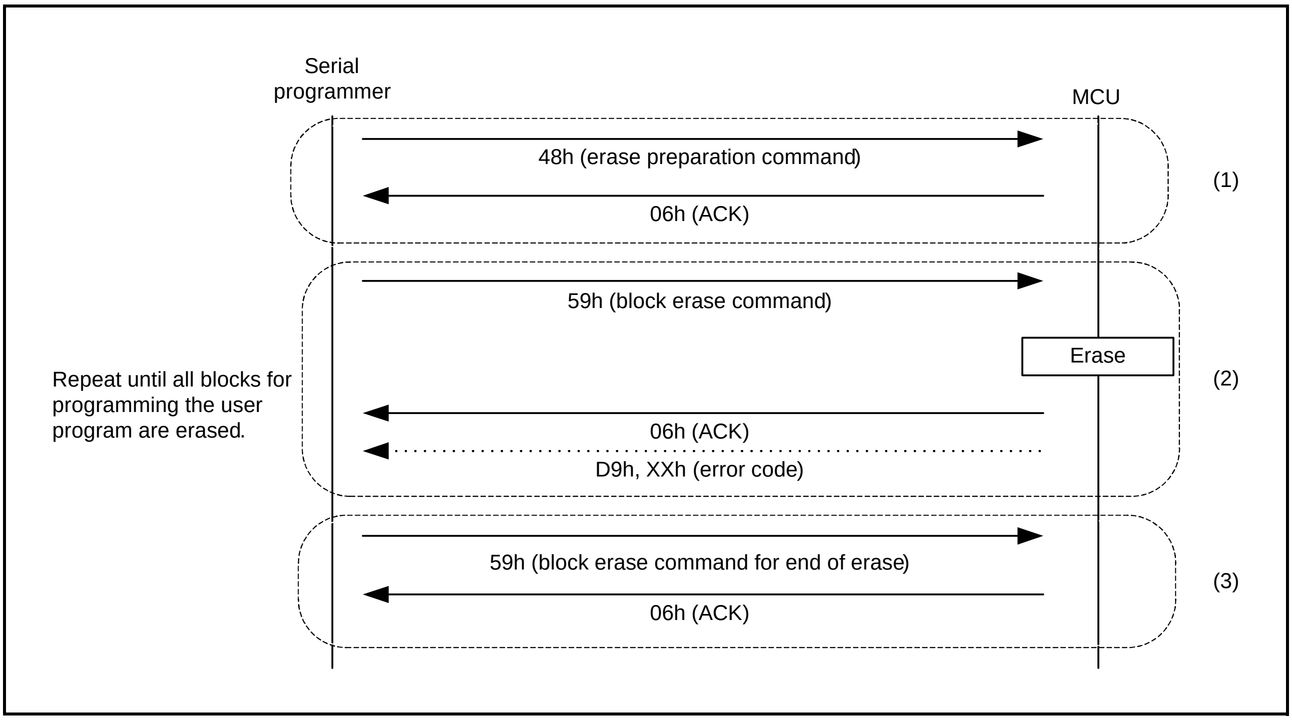


Figure 34.27 Procedure to Erase the User Area

34.10.7 Program the User Area

Program a user program in the user area.

(1) Send the user area program preparation command (43h).
(2) Send the program command (50h).
(3) To place the MCU in the program/erase host command wait state, send the program command (50h FFh FFh FFh B4h).

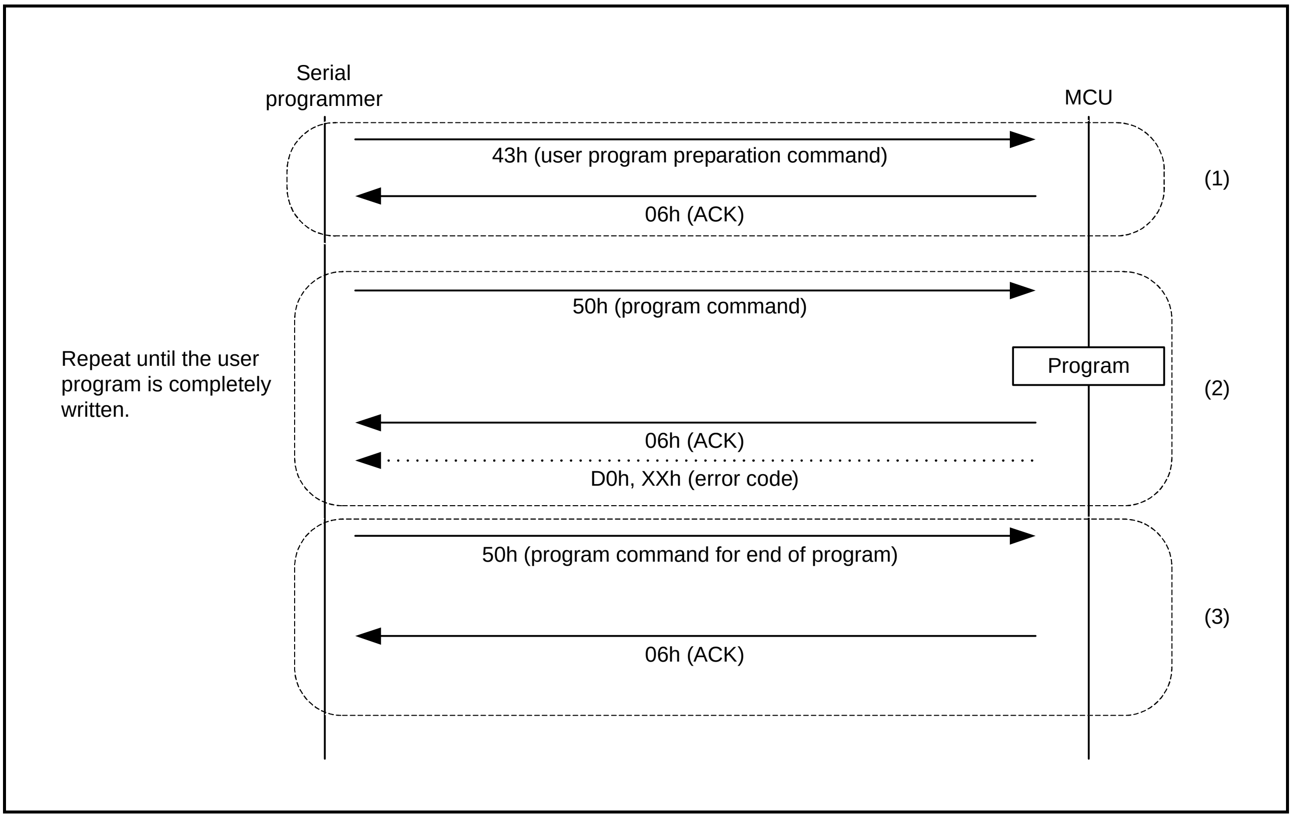


Figure 34.28 Procedure to Program the User Area

34.10.8 Check Data in the User Area

Read and check, checksum, and blank check the user area to check the programmed data in the user area.

(1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area.

(2) Send the user area checksum command (4Bh) to check program data using the checksum of user area.

(3) Send a user area blank check command (4Dh) to check if the user area has data.

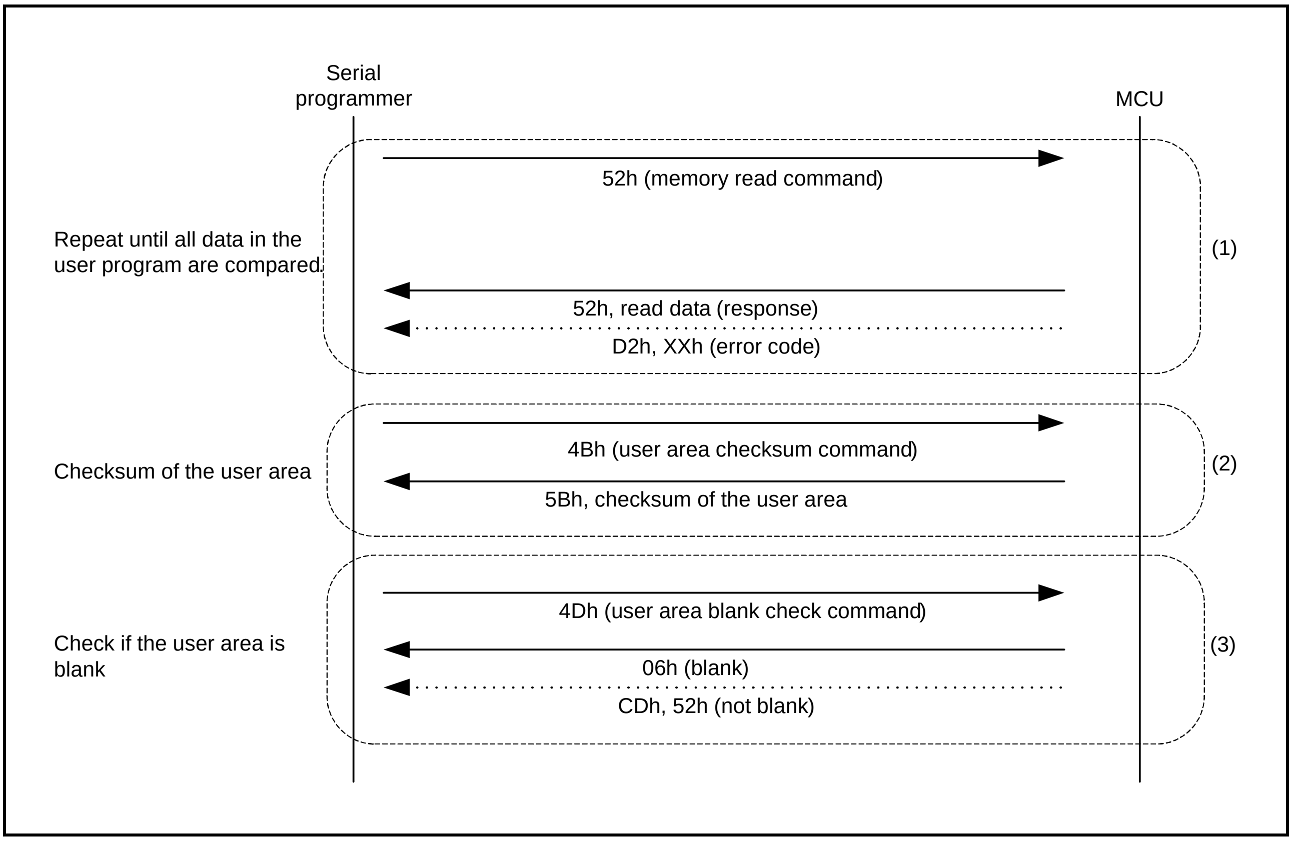


Figure 34.29 Procedure to Check Data in the User Area

34.10.9 Set the Access Window in the User Area

Set the access window to avoid unintentionally rewriting the user area during the self-programming.

(1) Send the access window program command (74h) to set the access window settings.
(2) Send the access window read command (73h) to confirm the access window settings.

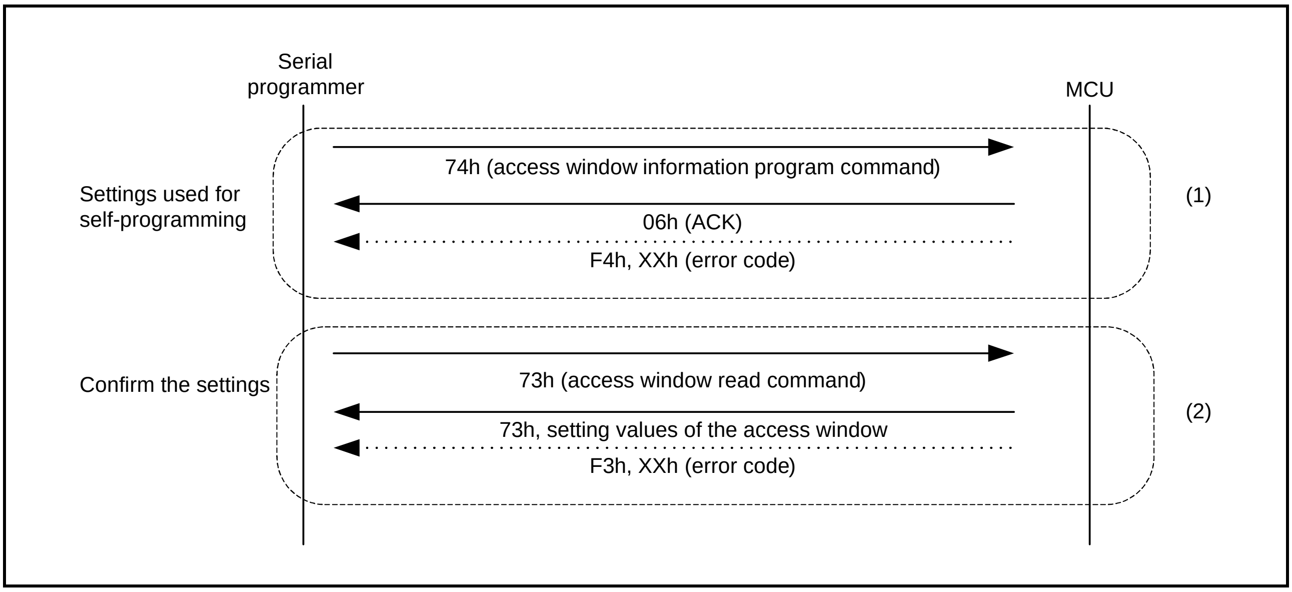
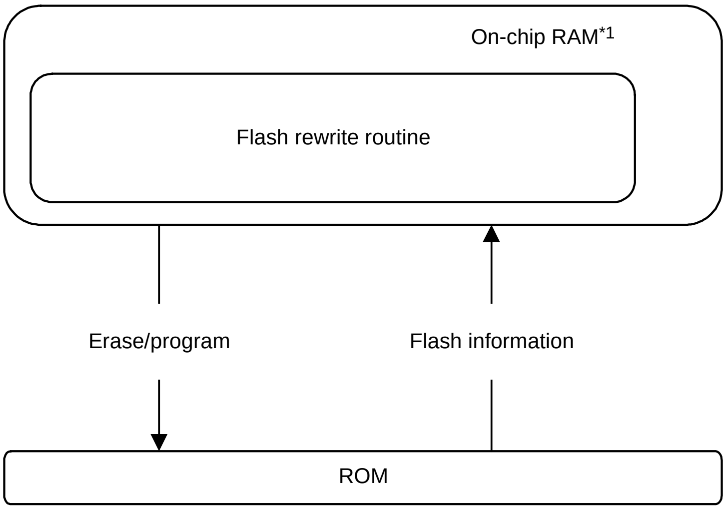


Figure 34.30 Procedure to Set the Access Window in the User Area

34.11 Rewriting by Self-Programming

34.11.1 Overview

The MCU supports rewriting of the flash memory by the user program. The ROM can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.



Note 1. The ROM cannot be rewritten by executing the flash rewrite routine on the ROM.

Figure 34.31 Self-Programming Overview

34.12 Usage Notes

(1) Access the Block Where Erase Operation is Forcibly Stopped
When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.

(2) Processing After Forced Stop of Erase Operation
When an erase operation is forcibly stopped, issue a block erase command again to the same block.

(3) Additional Programming Disabled
The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

(4) Reset during Program/Erase
If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 35, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.
The IWDT reset and software reset can be used regardless of tRESW.

(5) Non-maskable Interrupt Disabled during Program/Erase
When a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, or voltage monitoring 2 interrupt) occurs during a program/erase operation, the vectors are fetched from the ROM, and undefined data is read. Therefore, do not generate a non-maskable interrupt during a program/erase operation on the ROM.
(The description in (5) applies only to the ROM.)

(6) Location of Interrupt Vectors during a Program/Erase Operation
When an interrupt occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, set the destination for fetching interrupt vectors to an area other than the ROM with the CPU interrupt table register (INTB).

(7) Abnormal Termination during Program/Erase
When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (9), erase the area again.

(8) Actions Prohibited during Program/Erase
To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.

(9) FCLK during Program/Erase
For programming/erasure by self-programming, set the frequency of the FlashIF clock (FCLK), and specify an integer FCLK frequency (MHz) in FISR.PCKA[4:0] bits. Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz). If the FCLK is equal to or less than 4 MHz, only 1, 2, 3, or 4 MHz can be used.

34.13 Usage Notes in Boot Mode

(1) Notes on Communication Errors in Boot Mode

When communication with the MCU cannot be performed properly, reset and start up in boot mode again.

(2) Notes on Power Supply Voltage in Boot Mode (SCI)

When the bit rate exceeds 500 kbps in boot mode (SCI), use a voltage that is 3.0 or higher.

(3) Notes on Option-Setting Memory in Boot Mode

The settings of option function select register 0 (OFS0), option function select register 1 (OFS1), and endian select register (MDE) are disabled in boot mode.

(4) Notes on Switching the Start-Up Area

Switch the start-up area by self-programming.

35. Electrical Characteristics

35.1 Absolute Maximum Ratings

Table 35.1 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	Port 4	V_{in}	-0.3 to AVCC0+0.3	V
	Except for port 4 and ports for 5 V tolerant*1		–0.3 to VCC+0.3	V
	Ports for 5 V tolerant*1		–0.3 to +6.5	V
Reference power supply voltage		VREFH0	–0.3 to AVCC0+0.3	V
Analog power supply voltage		AVCC0	–0.3 to +6.5	V
Analog input voltage	When AN000 to AN007 used	V_{AN}	–0.3 to AVCC0+0.3	V
	When AN016 and AN017 used		–0.3 to VCC+0.3	
Operating temperature*2		T_{opr}	–40 to +85 –40 to +105	°C
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.
To preclude any malfunctions due to noise interference, insert capacitors of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the VREFH0 and VREFL0 pins. Place capacitors of about 0.1 µF as close as possible to every power supply pin and use the shortest and heaviest possible traces.
Connect the VCL pin to a VSS pin via a 4.7 µF capacitor. The capacitor must be placed close to the pin.
Do not input signals or an I/O pull-up power supply to ports other than 5-V tolerant ports while the device is not powered.
The current injection that results from input of such a signal or I/O pull-up may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.
Even if –0.3 to +6.5 V is input to 5-V tolerant ports, it will not cause problems such as damage to the MCU.

Note 1. Ports B1 and B2 are 5 V tolerant.

Note 2. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, refer to section 1.2, List of Products.

Table 35.2 Recommended Operating Voltage Conditions

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltages	VCC*1, *2		2.7	—	5.5	V
	VSS		—	0	—	
Analog power supply voltages	AVCC0*1, *2		VCC	—	5.5	V
	VREFH0*1, *2		—	AVCC0	—	
	AVSS0, VREFL0		—	0	—	

Note 1. AVCC0/VREFH0 and VCC can be set individually within the operating range.

Note 2. When powering on the VCC and AVCC0/VREFH0 pins, power them on at the same time or the VCC pin first and then the AVCC0/VREFH0 pin.

35.2 DC Characteristics

Table 35.3 DC Characteristics (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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 B1 and B2 (5 V tolerant)		VCC × 0.8	—	5.8		
	Ports 00 to 02, 10, 11 Ports 22 to 24 Ports 30 to 33, 36, 37 Ports 70 to 76 Ports 91 to 94 Ports A2 to A5 Ports B0, B3 to B7 Ports D3 to D7 Port E2 Port RES#		VCC × 0.8	—	VCC + 0.3		
	Ports 40 to 47		AVCC0 × 0.8	—	AVCC0 + 0.3		
	RIIC input pin (except for SMBus)	V_{IL}	–0.3	—	VCC × 0.3		
	Ports 40 to 47		–0.3	—	AVCC0 × 0.2		
	Other than RIIC input pin or ports 40 to 47		–0.3	—	VCC × 0.2		
	RIIC input pin (except for SMBus)	ΔV_T	VCC × 0.05	—	—		
	Ports 40 to 47		AVCC0 × 0.1	—	—		
	Other than RIIC input pin or ports 40 to 47		VCC × 0.1	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL (external clock input)		VCC × 0.8	—	VCC + 0.3		
	RIIC input pin (SMBus)		2.1	—	VCC + 0.3		
	MD	V_{IL}	–0.3	—	VCC × 0.1		
	EXTAL (external clock input)		–0.3	—	VCC × 0.2		
	RIIC input pin (SMBus)		–0.3	—	0.8		

Table 35.4 DC Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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, port E2	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	Port 4	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V, AVCC0
	Ports except for 5-V tolerant ports and port 4		—	—	0.2		V_{in} = 0 V, VCC
	Ports for 5 V tolerant		—	—	1.0		V_{in} = 0 V, 5.8 V
Input capacitance	All input pins	C_{in}	—	4	15	pF	V_{in} = 0 mV, f = 1 MHz, T_a = 25°C
Input pull-up resistor	All ports (except for port E2)	R_U	10	20	50	kΩ	V_{in} = 0 V

Table 35.5 DC Characteristics (3)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item					Symbol	Typ.	Max.	Unit	Test Conditions
Supply current*1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 40 MHz	I_{CC}	4.6	—	mA	
				ICLK = 32 MHz		3.9	—		
				ICLK = 16 MHz		2.8	—		
				ICLK = 8 MHz		2.2	—		
			All peripheral operation: Normal	ICLK = 40 MHz*3		15.0	—		
				ICLK = 32 MHz*4		12.4	—		
				ICLK = 16 MHz*4		7.2	—		
				ICLK = 8 MHz*4		4.6	—		
			All peripheral operation: Max.	ICLK = 40 MHz*3		—	33.0		
				ICLK = 32 MHz*4		—	24.5		
		Sleep mode	No peripheral operation*2	ICLK = 40 MHz		2.7	—		
				ICLK = 32 MHz		2.3	—		
				ICLK = 16 MHz		1.9	—		
				ICLK = 8 MHz		1.6	—		
			All peripheral operation: Normal	ICLK = 40 MHz*3		6.8	—		
				ICLK = 32 MHz*4		5.7	—		
				ICLK = 16 MHz*4		3.6	—		
				ICLK = 8 MHz*4		2.5	—		
		Deep sleep mode	No peripheral operation*2	ICLK = 40 MHz		1.7	—		
				ICLK = 32 MHz		1.5	—		
				ICLK = 16 MHz		1.3	—		
				ICLK = 8 MHz		1.3	—		
			All peripheral operation: Normal	ICLK = 40 MHz*3		5.3	—		
				ICLK = 32 MHz*4		4.4	—		
				ICLK = 16 MHz*4		2.8	—		
				ICLK = 8 MHz*4		2.0	—		
	Middle-speed operating modes	Normal operating mode	No peripheral operation*6	ICLK = 12 MHz	I_{CC}	2.6	—	mA	
				ICLK = 8 MHz		1.9	—		
				ICLK = 1 MHz		1.3	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		5.5	—		
				ICLK = 8 MHz		4.2	—		
				ICLK = 1 MHz		1.6	—		
			All peripheral operation: Max.*7	ICLK = 12 MHz		—	11.0		

Item					Symbol	Typ.	Max.	Unit	Test Conditions
Supply current *1	Middle-speed operating modes	Sleep mode	No peripheral operation*6	ICLK = 12 MHz	I_{CC}	2.0	—	mA	
				ICLK = 8 MHz		1.4	—		
				ICLK = 1 MHz		1.2	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		2.8	—		
				ICLK = 8 MHz		2.3	—		
				ICLK = 1 MHz		1.3	—		
		Deep sleep mode	No peripheral operation*6	ICLK = 12 MHz		1.5	—		
				ICLK = 8 MHz		1.2	—		
				ICLK = 1 MHz		1.1	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		2.8	—		
				ICLK = 8 MHz		2.3	—		
				ICLK = 1 MHz		1.1	—		

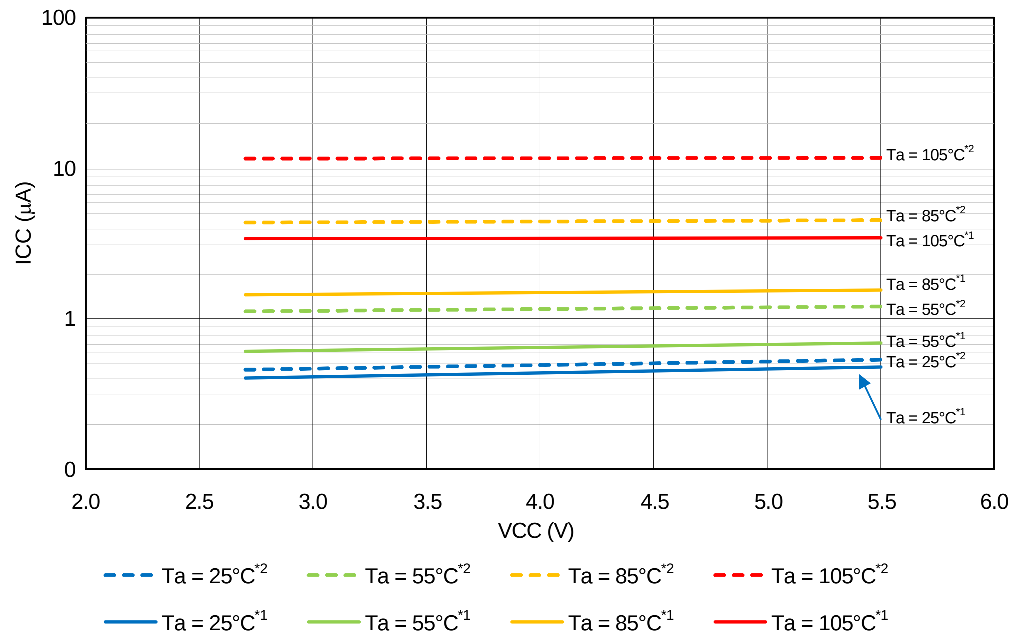
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Supply of the clock signal to peripheral modules is stopped in this state. The clock source is PLL. FCLK and PCLK are division by 64.
Note 3. The clock signal to peripheral modules is supplied in this state. The clock source is PLL. FCLK is division by 2. The frequency of PCLK is same as ICLK.
Note 4. The clock signal to peripheral modules is supplied in this state. The clock source is PLL. The frequencies of FCLK and PCLK are same as ICLK.
Note 5. Values when VCC = 5 V.
Note 6. Supply of the clock signal to peripheral modules is stopped in this state. The clock source is PLL. FCLK and PCLK are division by 64.
Note 7. Supply of the clock signal to peripheral modules is stopped in this state. The clock source is PLL. The frequencies of FCLK and PCLK are same as ICLK.

Table 35.6 DC Characteristics (4)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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	T_a = 25°C	I_{CC}	0.45	0.91	μA	
		T_a = 55°C		0.66	2.23		
		T_a = 85°C		1.50	9.14		
		T_a = 105°C		3.42	23.94		

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 = 5 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 35.1 Voltage Dependency in Software Standby Mode (Reference Data)

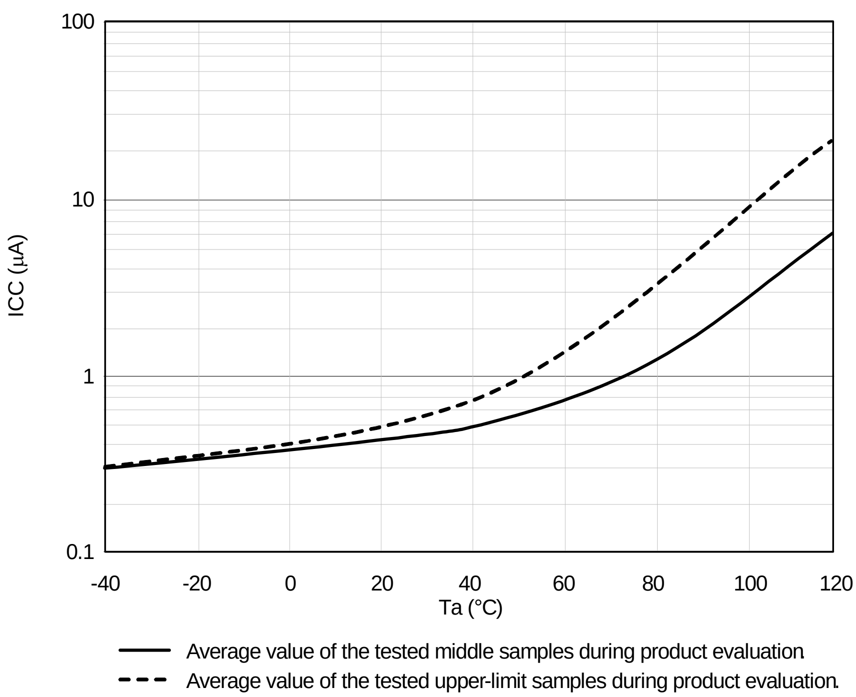


Figure 35.2 Temperature Dependency in Software Standby Mode (Reference Data)

Table 35.7 DC Characteristics (5)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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	—	300	mW	D-version product
Permissible total consumption power*1	Pd	—	125	mW	G-version product

Note: Please contact a Renesas Electronics sales office for information on the derating of the G-version product. 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 35.8 DC Characteristics (6)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.*2	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion (Sample-and-hold circuits in use)	I_{AVCC}	—	3.1	5.2	mA	
	During A/D conversion (Sample-and-hold circuits not in use)		—	0.9	1.8		
	During D/A conversion*1		—	0.4	0.9		
	Waiting for A/D and D/A conversion (all units)		—	—	0.4	μA	
Reference power supply current	During A/D conversion	I_{REFH0}	—	80	130	μA	
	Waiting for A/D conversion (all units)		—	—	60	nA	
Comparator C operating current*3	Comparator enabled (per channel)	I_{CMP}	—	40	60	μA	

Note 1. The value of the D/A converter is the value of the power supply current including the reference current.

Note 2. When VCC = AVCC0 = 5 V.

Note 3. Current consumed only by the comparator C module.

Table 35.9 DC Characteristics (7)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power-on VCC rising gradient	At normal startup	SrVCC	0.02	—	20	ms/V	
	Voltage monitoring 0 reset enabled at startup*1, *2		0.02	—	—		

Note 1. When OFS1.LVDAS = 0.

Note 2. Turn on the power supply voltage according to the normal startup rising gradient because the register settings set by OFS1 are not read in boot mode.

Table 35.10 DC Characteristics (8)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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 (2.7 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	Figure 35.3 $V_{r\ (VCC)}$ ≤ VCC × 0.2
		—	—	1	MHz	Figure 35.3 $V_{r\ (VCC)}$ ≤ VCC × 0.08
		—	—	10	MHz	Figure 35.3 $V_{r\ (VCC)}$ ≤ VCC × 0.06
Allowable voltage change rising/falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

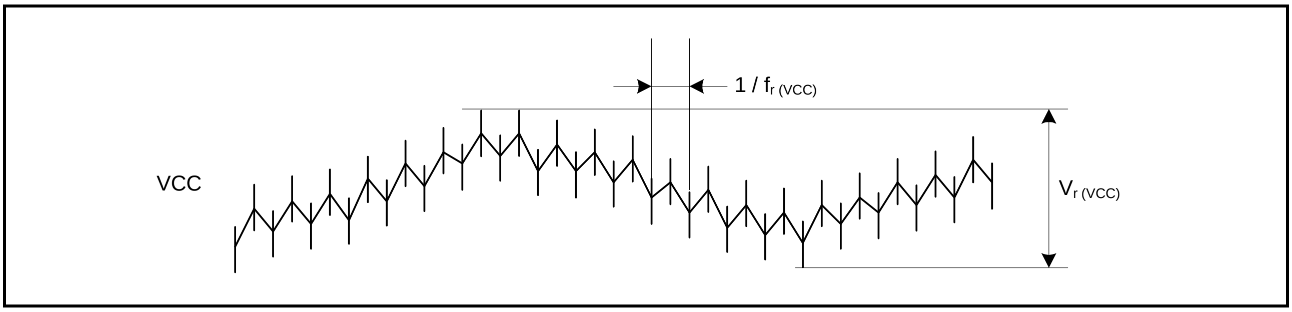


Figure 35.3 Ripple Waveform

Table 35.11 DC Characteristics (9)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Permissible error of VCL pin external capacitance	C_{VCL}	3.3	4.7	6.1	μF	

Note: The recommended capacitance is 4.7 μF. Variations in connected capacitors should be within the above range.

Table 35.12 Permissible Output Currents

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Max.	Unit
Permissible output low current	Large current ports (Ports 71 to 76, port B5, port D3)		I_{OL}	10.0	mA
	RIIC pins			6.0	
	Ports other than above	Normal output mode		4.0	
		High-drive output mode		8.0	
Permissible output low current	Total of large current ports		ΣI_{OL}	50	
	Total of all output pins			110	
Permissible output high current	Large current ports (Ports 71 to 76, port B5, port D3)		I_{OH}	–5.0	
	Ports other than above	Normal output mode		–4.0	
		High-drive output mode		–8.0	
Permissible output high current	Total of large current ports		ΣI_{OH}	–25	
	Total of all output pins			–35	

Note: Do not exceed the permissible total supply current.

Table 35.13 Output Values of Voltage

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Output low	Large current ports (Ports 71 to 76, port B5, port D3)		V_{OL}	—	0.8	V	I_{OL} = 10.0mA
	RIIC pins	Standard mode		—	0.4		I_{OL} = 3.0mA
		Fast mode		—	0.6		I_{OL} = 6.0mA
	Ports other than above	Normal output mode		—	0.8		I_{OL} = 1.0mA
		High-drive output mode		—	0.8		I_{OL} = 2.0mA
Output high	Large current ports (Ports 71 to 76, port B5, port D3)		V_{OH}	VCC – 0.8	—	V	I_{OH} = –5.0mA
	Ports 40 to 47			AVCC0 – 0.8	—		I_{OH} = –2.0mA
	Ports other than above	Normal output mode		VCC – 0.8	—		I_{OH} = –2.0mA
		High-drive output mode		VCC – 0.8	—		I_{OH} = –4.0mA

35.2.1 Normal I/O Pin Output Characteristics (1)

Figure 35.4 to Figure 35.7 show the characteristics when normal output is selected by the drive capacity control register.

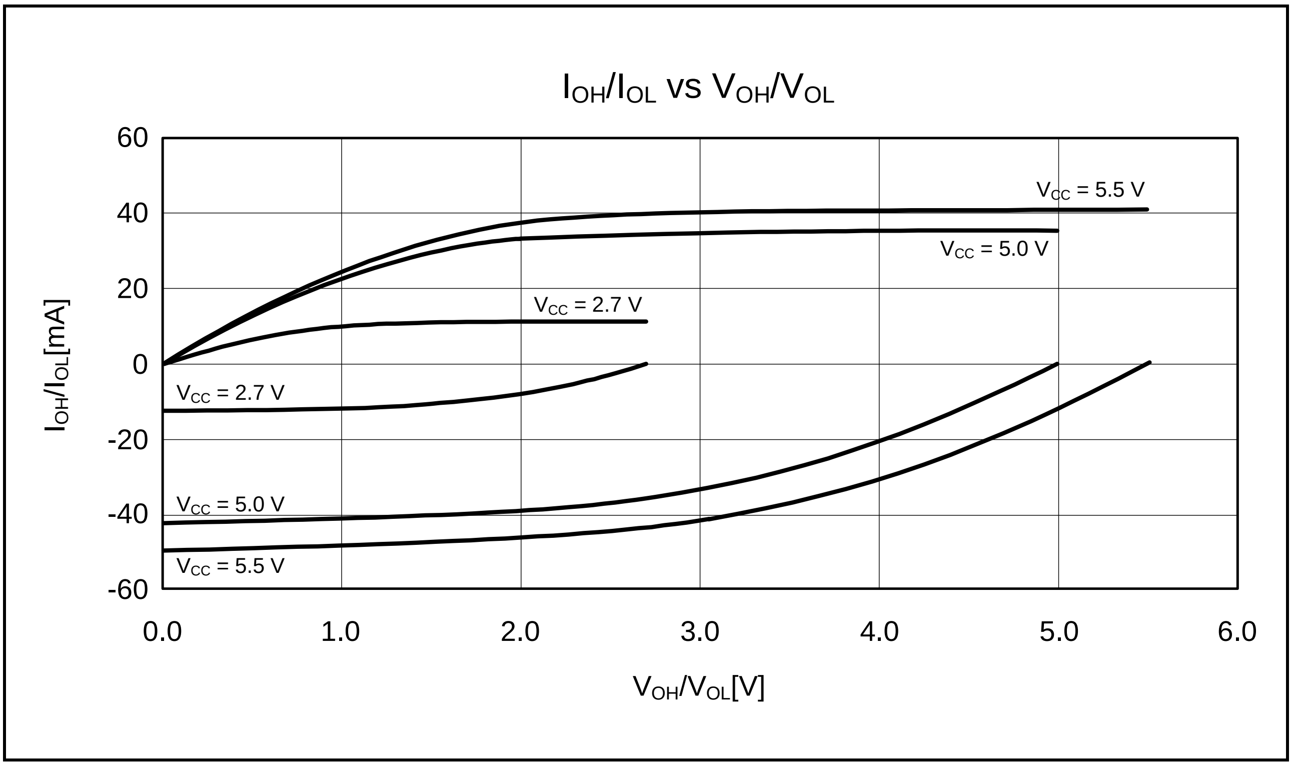


Figure 35.4 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C When Normal Output is Selected (Reference Data)

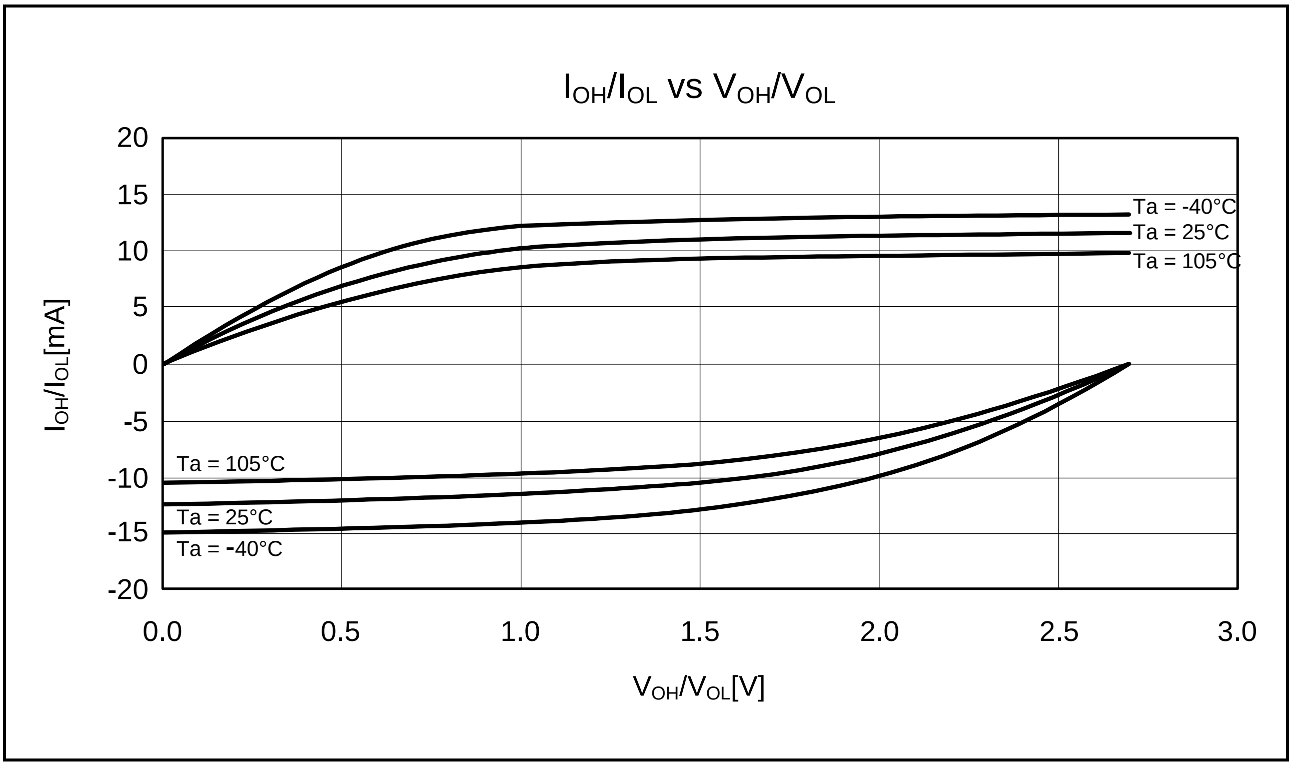


Figure 35.5 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V when Normal Output is Selected (Reference Data)

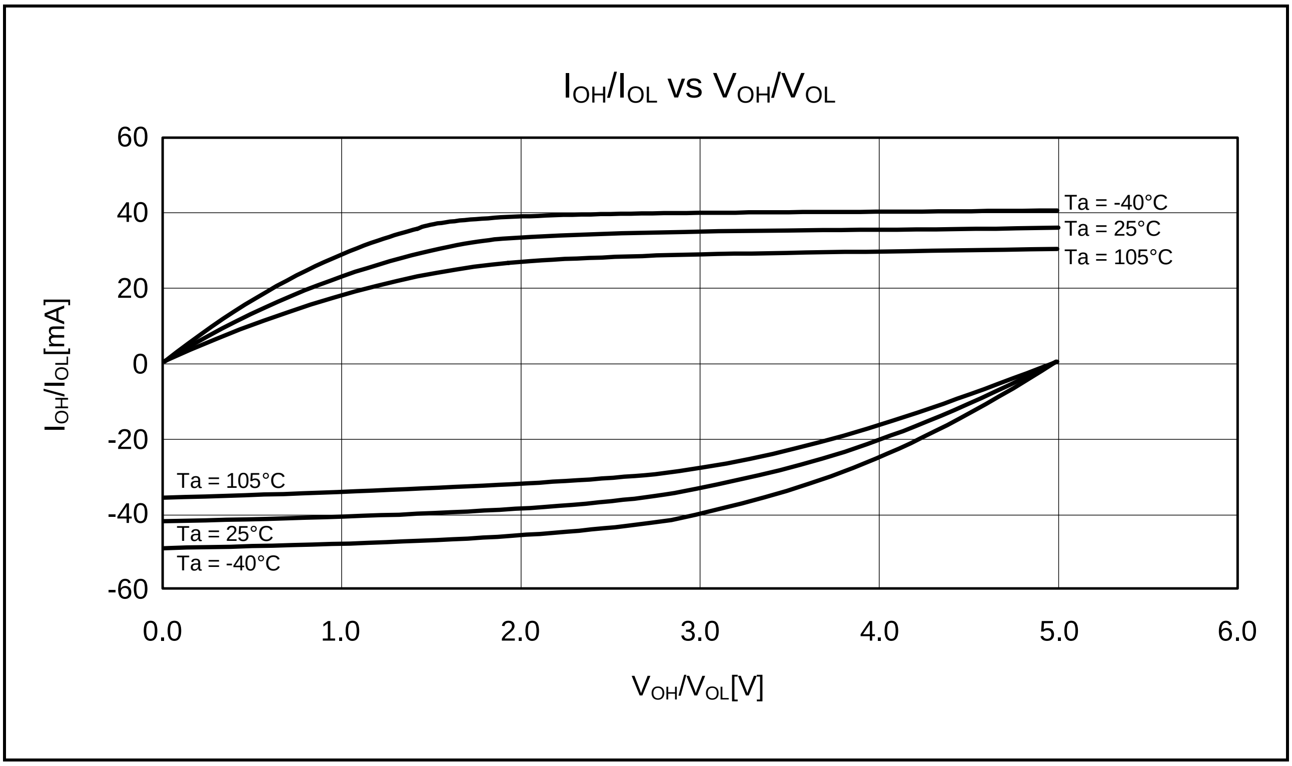


Figure 35.6 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.0 V when Normal Output is Selected (Reference Data)

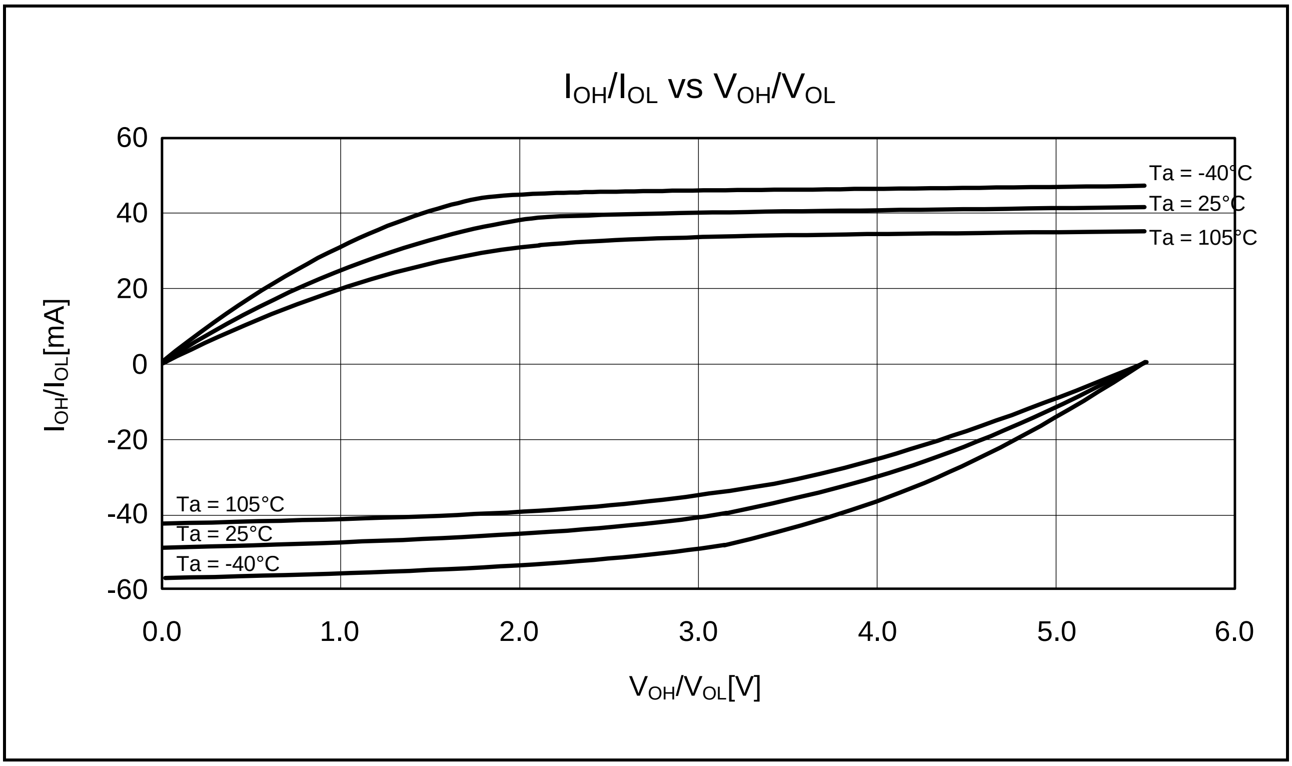


Figure 35.7 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.5 V when Normal Output is Selected (Reference Data)

35.2.2 Standard I/O Pin Output Characteristics (2)

Figure 35.8 to Figure 35.11 show the characteristics when high-drive output is selected by the drive capacity control register.

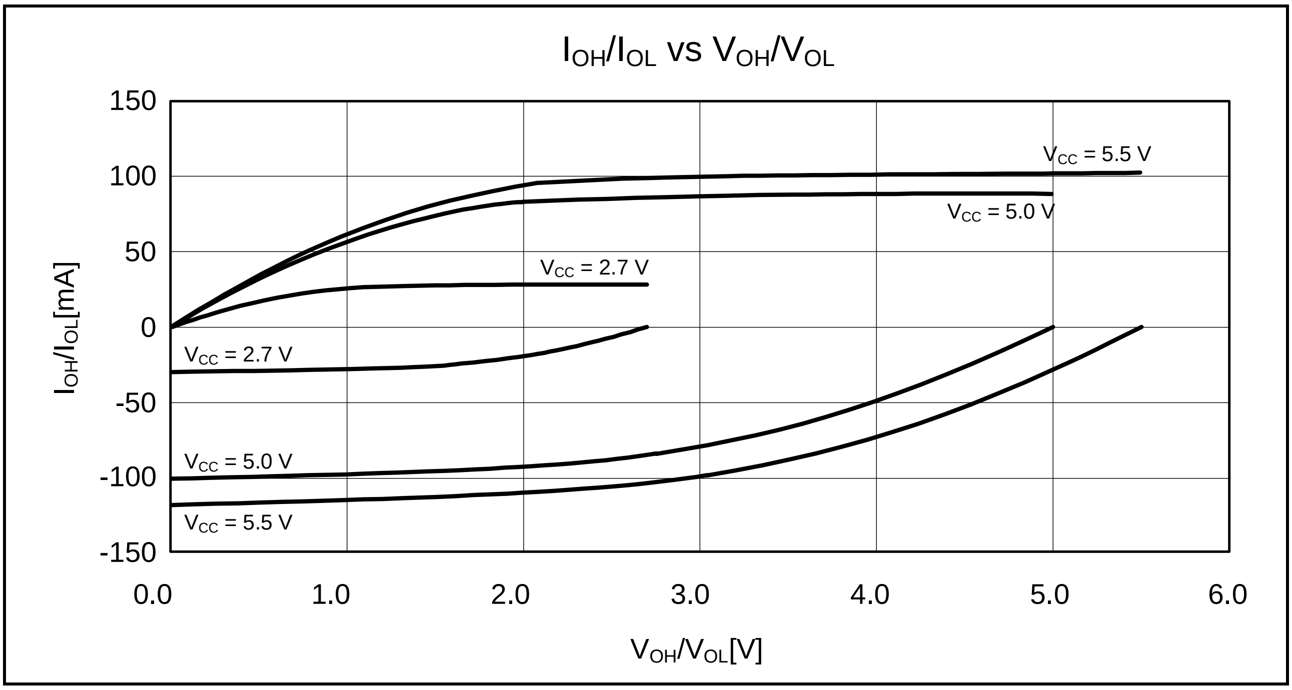


Figure 35.8 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics at T_a = 25°C When Normal Output is Selected (Reference Data)

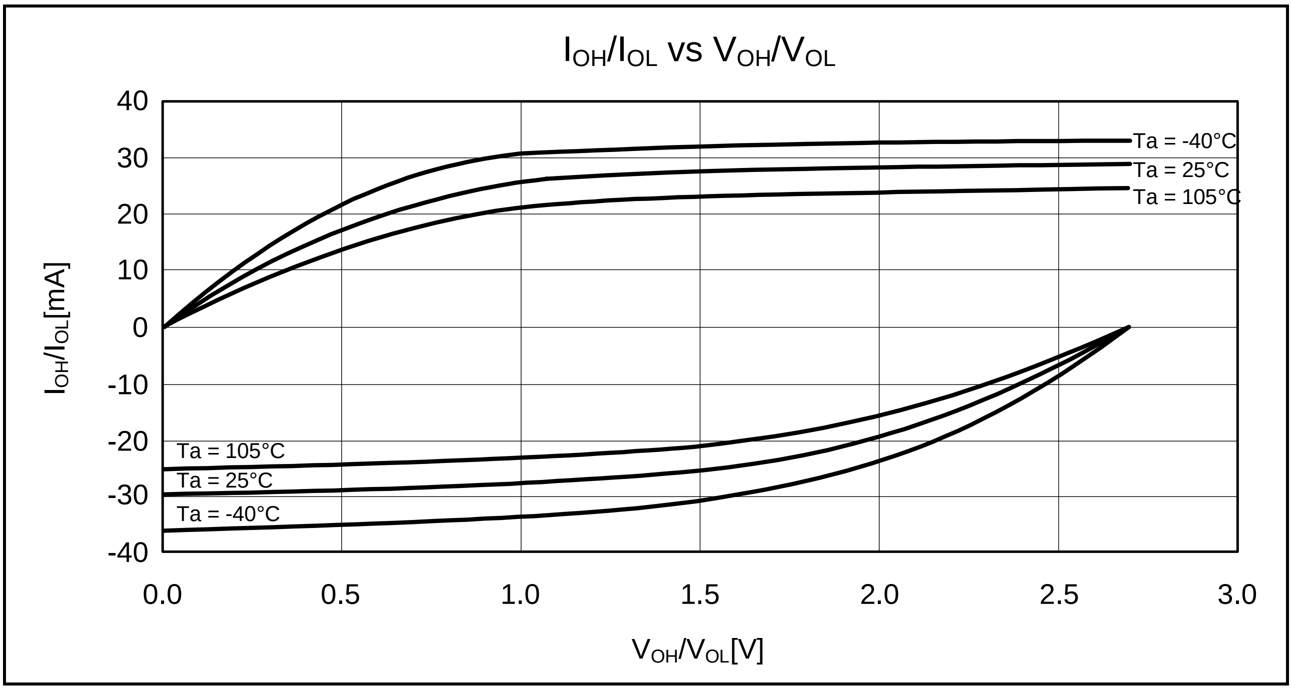


Figure 35.9 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 2.7 V when Normal Output is Selected (Reference Data)

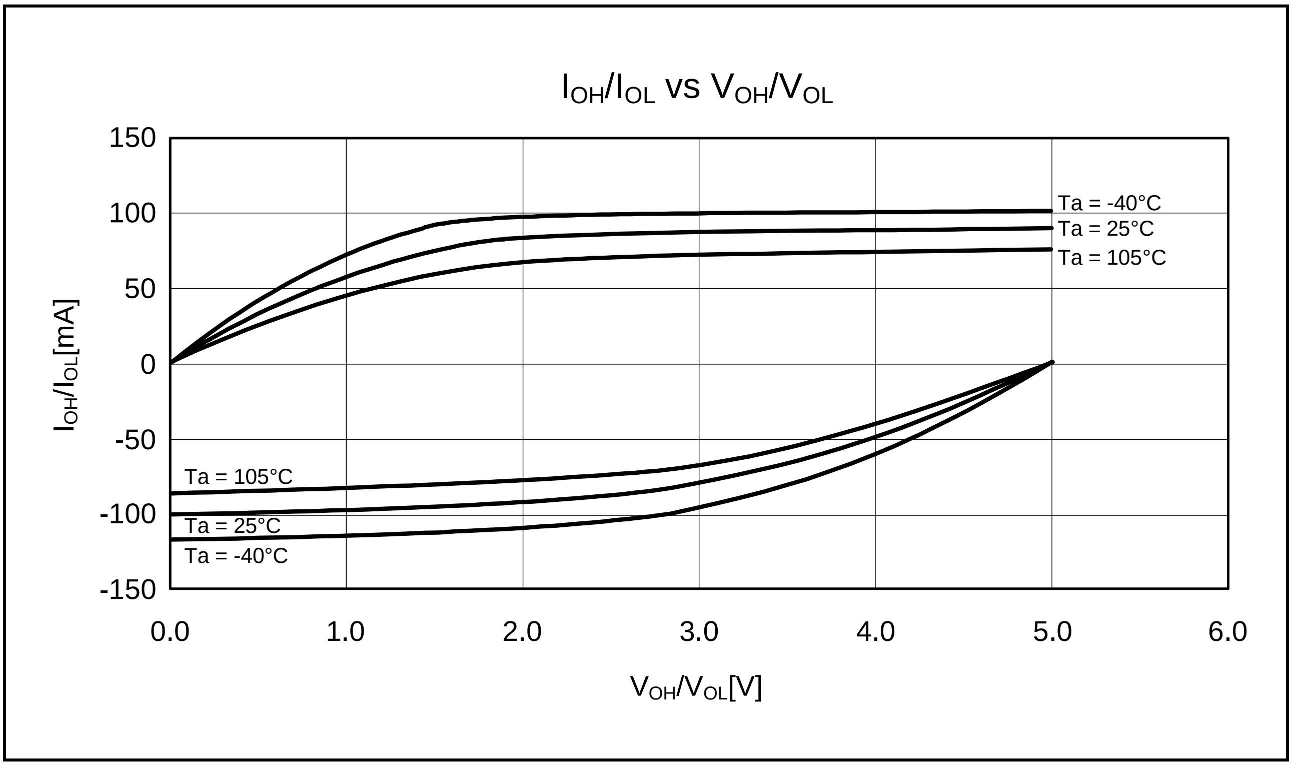


Figure 35.10 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.0 V when Normal Output is Selected (Reference Data)

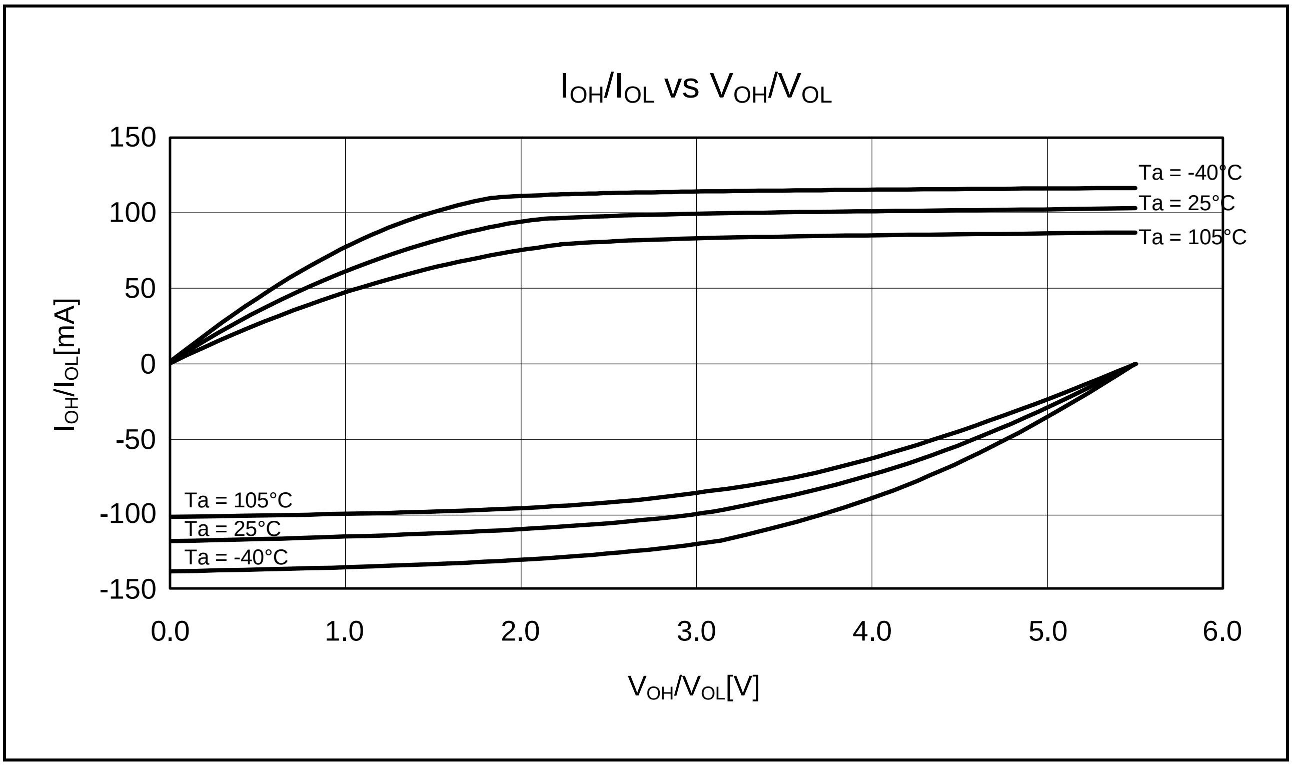


Figure 35.11 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics at VCC = 5.5 V when Normal Output is Selected (Reference Data)

35.2.3 Standard I/O Pin Output Characteristics (3)

Figure 35.12 to Figure 35.15 show the output characteristics of the large current ports (ports 71 to 76, port B5, port D3).

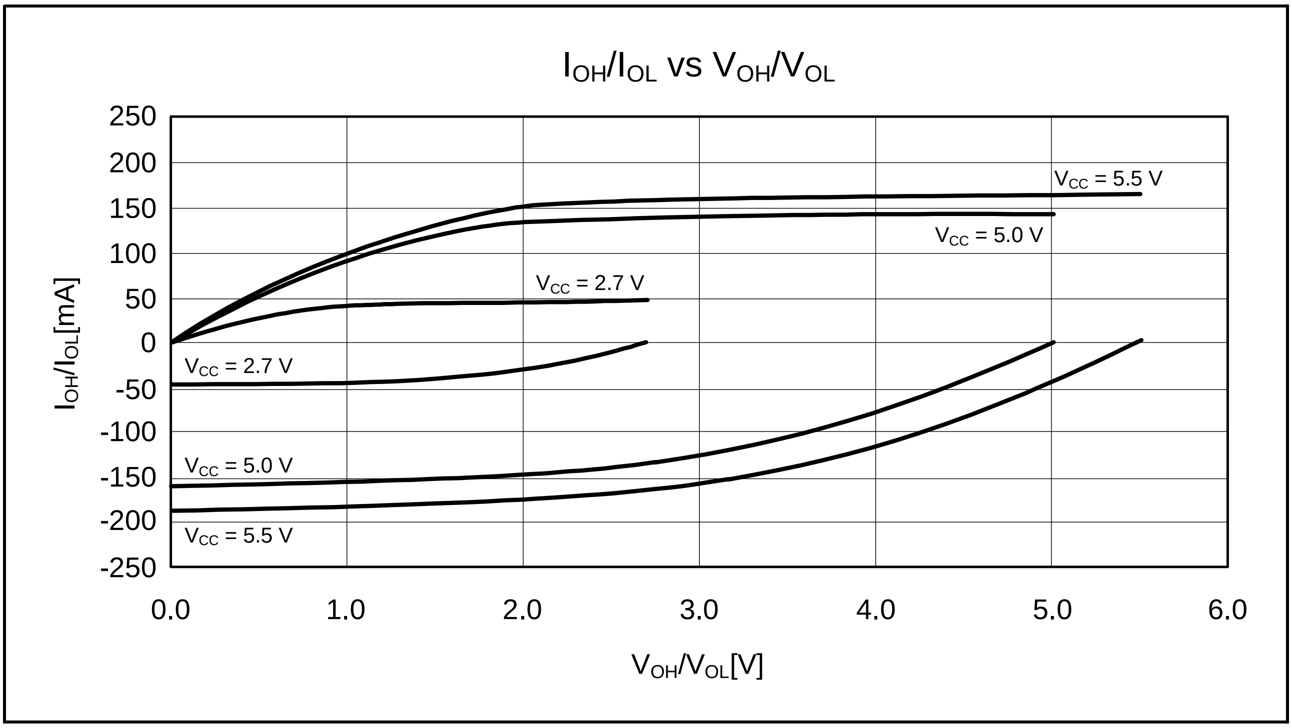


Figure 35.12 V_{OH}/V_{OL} and I_{OH}/I_{OL} Voltage Characteristics of Large Current Ports (Ports 71 to 76, Port B5, Port D3) at T_a = 25°C (Reference Data)

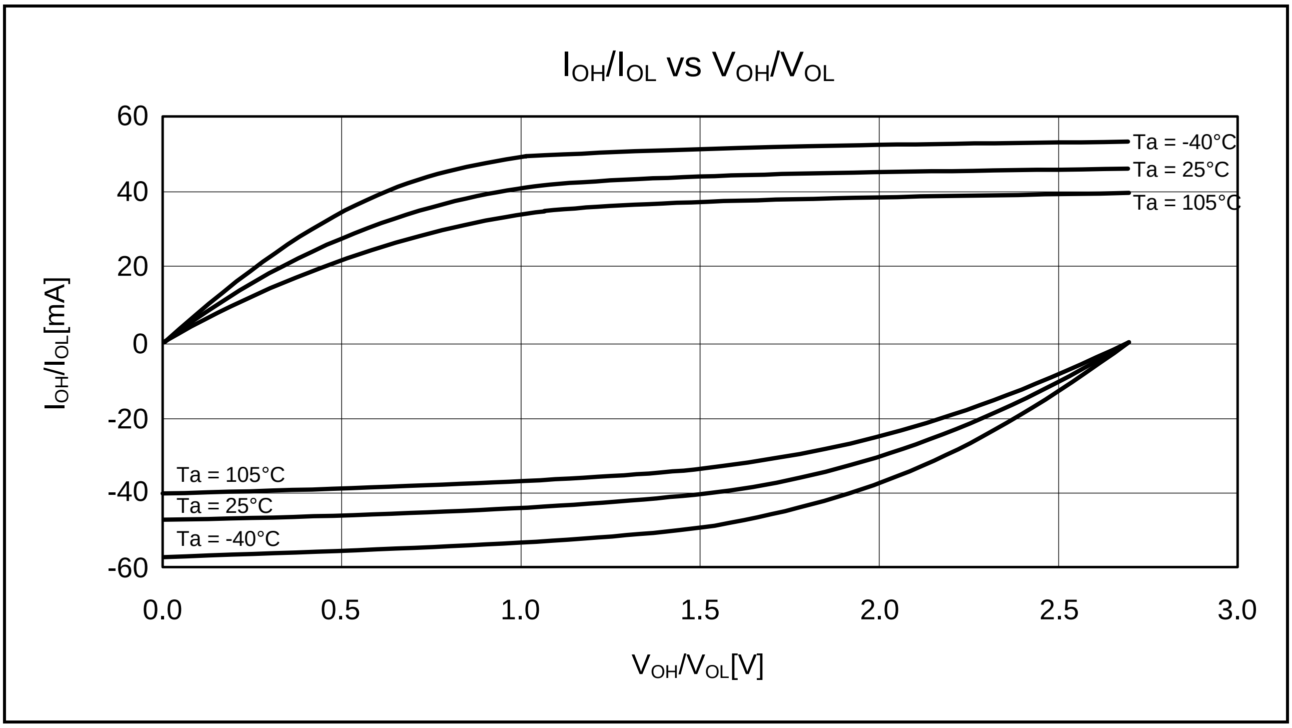


Figure 35.13 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics of Large Current Ports (Ports 71 to 76, Port B5, Port D3) at VCC = 2.7 V (Reference Data)

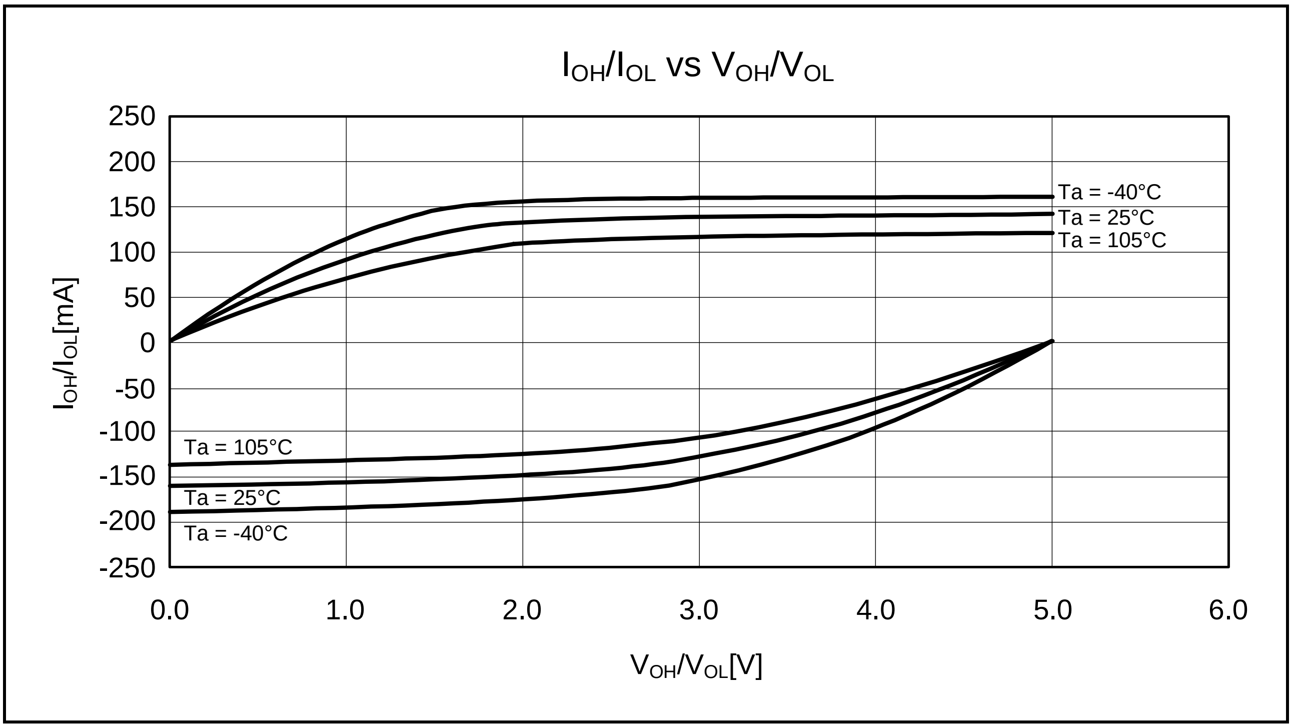


Figure 35.14 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics of Large Current Ports (Ports 71 to 76, Port B5, Port D3) at VCC = 5.0 V (Reference Data)

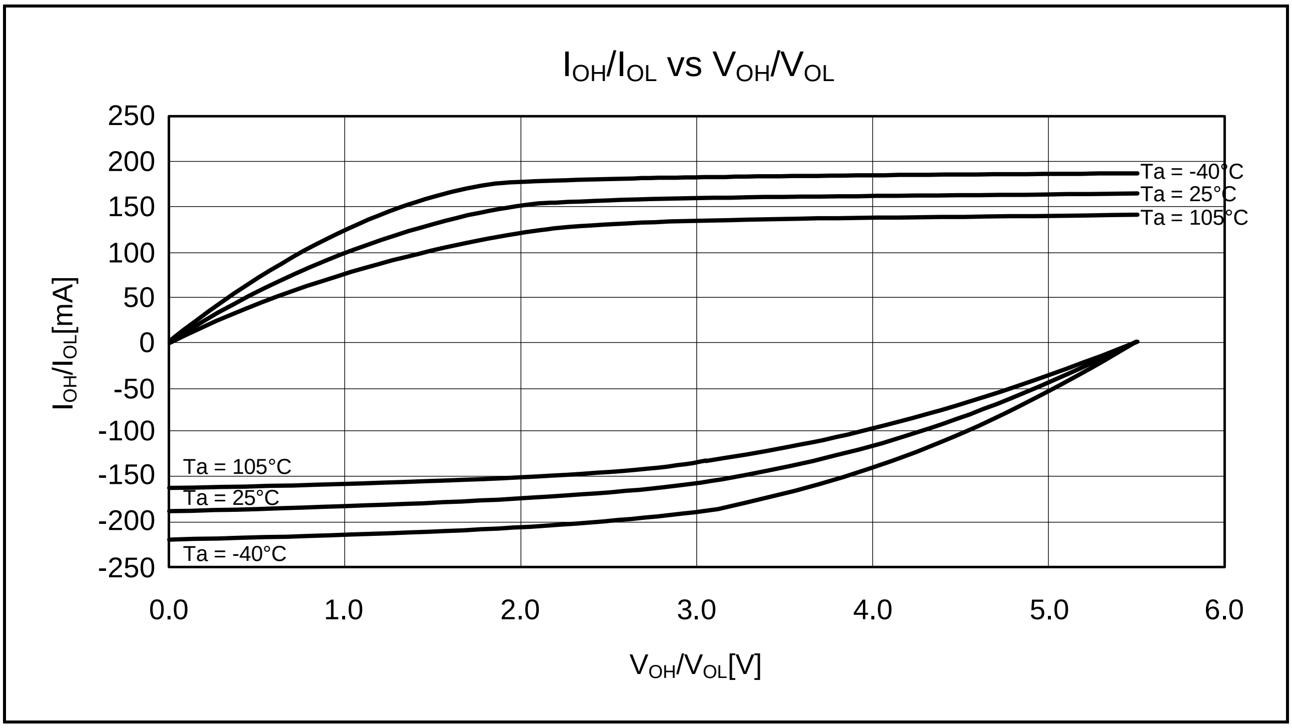


Figure 35.15 V_{OH}/V_{OL} and I_{OH}/I_{OL} Temperature Characteristics of Large Current Ports (Ports 71 to 76, Port B5, Port D3) at VCC = 5.5 V (Reference Data)

35.2.4 RIIC Pin Output Characteristics

Figure 35.16 to Figure 35.19 show the output characteristics of the RIIC pin.

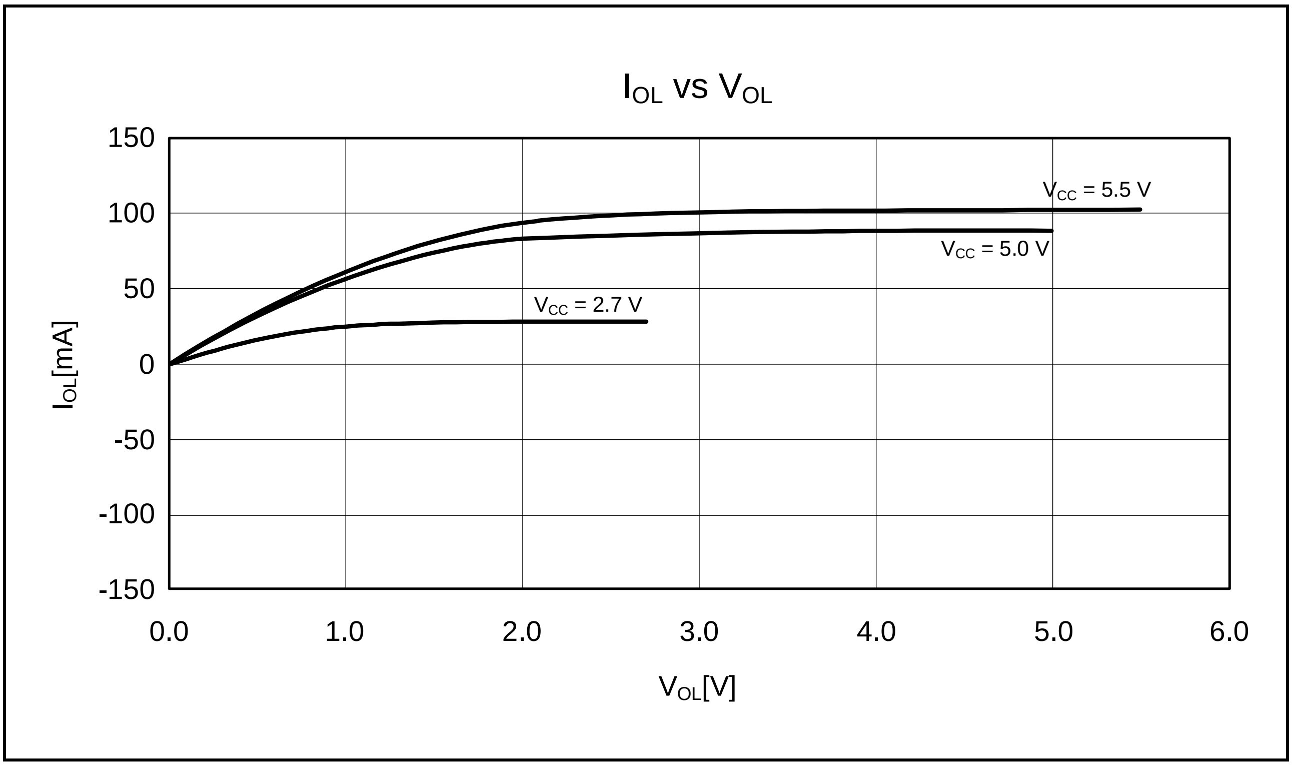


Figure 35.16 V_{OL} and I_{OL} Voltage Characteristics of RIIC Output Pin at T_a = 25°C (Reference Data)

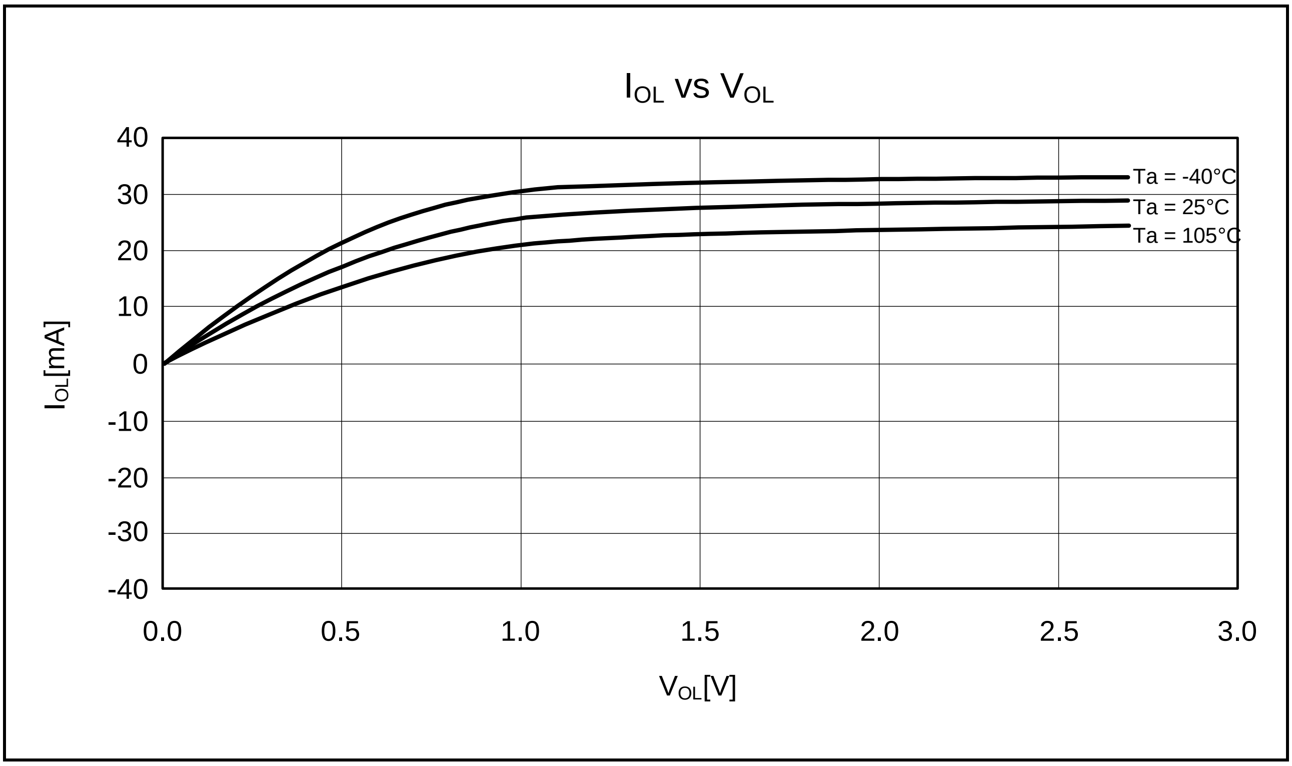


Figure 35.17 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 2.7 V (Reference Data)

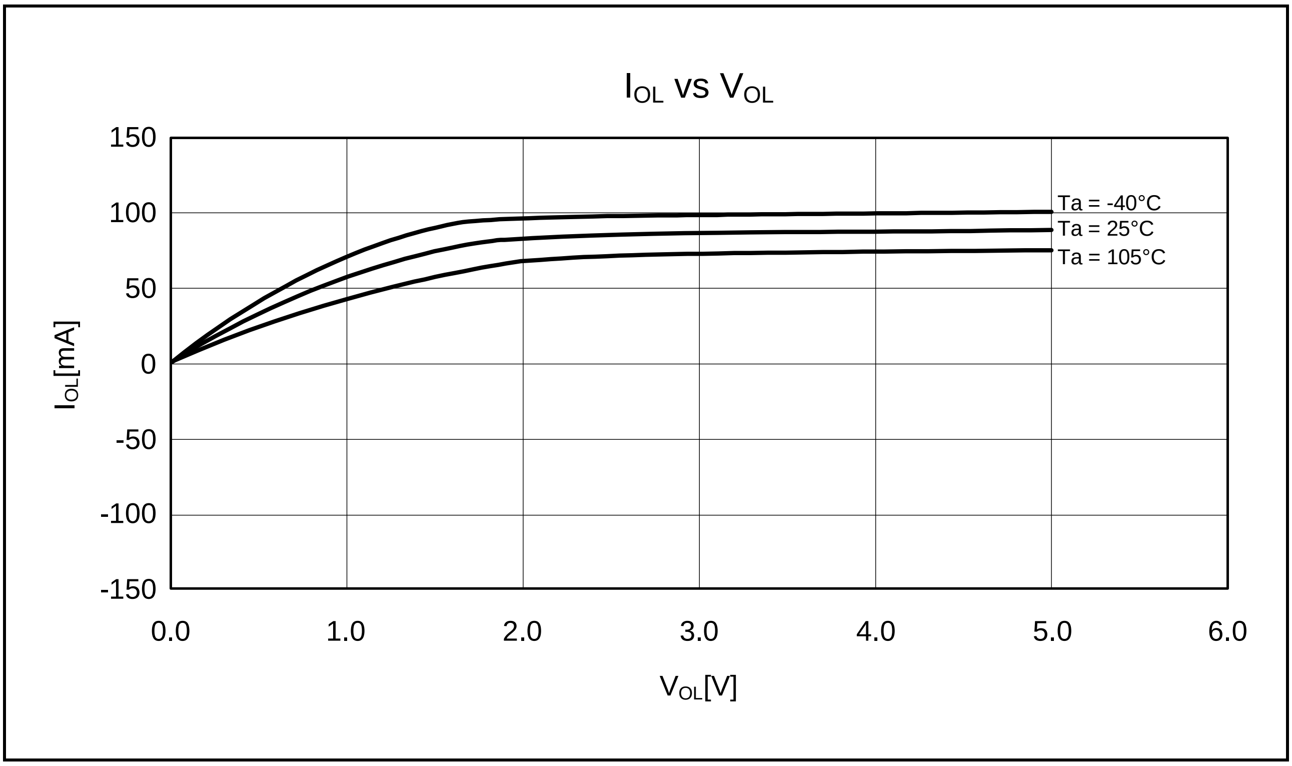


Figure 35.18 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 5.0 V (Reference Data)

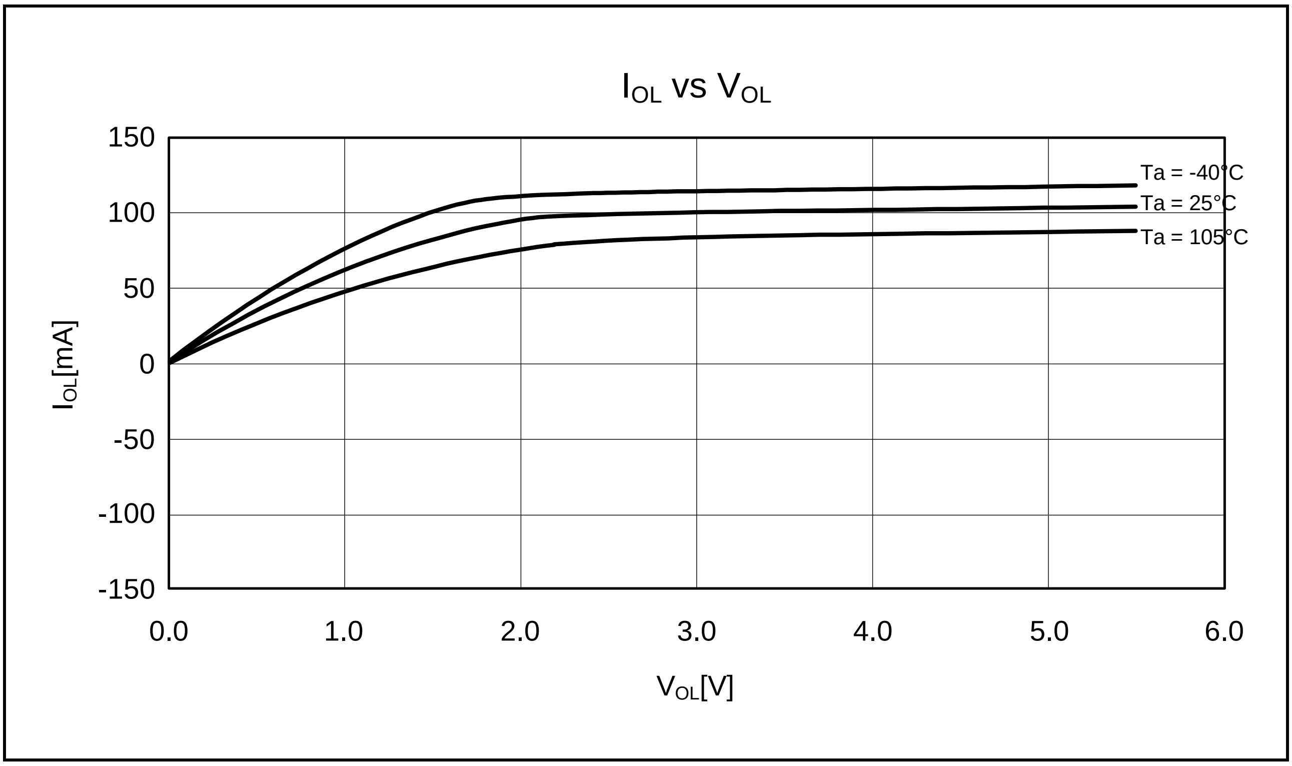


Figure 35.19 V_{OL} and I_{OL} Temperature Characteristics of RIIC Output Pin at VCC = 5.5 V (Reference Data)

35.3 AC Characteristics

35.3.1 Clock Timing

Table 35.14 Operating Frequency Value (High-Speed Operating Mode)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	min.	typ.	max.	Unit
Maximum operating frequency	System clock (ICLK)	f_{max}	—	—	40	MHz
	FlashIF clock (FCLK)*1, *2		—	—	32	
	Peripheral module clock (PCLKA)		—	—	40	
	Peripheral module clock (PCLKB)		—	—	40	
	Peripheral module clock (PCLKD)		—	—	40	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.

Table 35.15 Operating Frequency Value (Middle-Speed Operating Mode)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	min.	typ.	max.	Unit
Maximum operating frequency	System clock (ICLK)	f_{max}	—	—	12	MHz
	FlashIF clock (FCLK)*1, *2		—	—	12	
	Peripheral module clock (PCLKA)		—	—	12	
	Peripheral module clock (PCLKB)		—	—	12	
	Peripheral module clock (PCLKD)		—	—	12	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.

Table 35.16 Clock Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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_{Xcyc}	50	—	—	ns	Figure 35.20
EXTAL external clock input high pulse width	t_{XH}	20	—	—	ns	
EXTAL external clock input low pulse width	t_{XL}	20	—	—	ns	
EXTAL external clock rise time	t_{Xr}	—	—	5	ns	
EXTAL external clock fall time	t_{Xf}	—	—	5	ns	
EXTAL external clock input wait time*1	t_{EXWT}	0.5	—	—	µs	
Main clock oscillator oscillation frequency*2	f_{MAIN}	1	—	20	MHz	
Main clock oscillation stabilization time (crystal)*2	$t_{MAINOSC}$	—	3	—	ms	Figure 35.21
Main clock oscillation stabilization time (ceramic resonator)*2	$t_{MAINOSC}$	—	50	—	µs	
LOCO clock oscillation frequency	f_{LOCO}	3.44	4.0	4.56	MHz	
LOCO clock oscillation stabilization time	t_{LOCO}	—	—	0.5	µs	Figure 35.22
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	
IWDT-dedicated clock oscillation stabilization time	t_{ILOCO}	—	—	50	µs	Figure 35.23
HOCO clock oscillation frequency	f_{HOCO}	31.52	32	32.48	MHz	T_a = –40 to +85°C
		31.68	32	32.32		T_a = –20 to +85°C
		31.36	32	32.64		T_a = –40 to +105°C
HOCO clock oscillation stabilization time	t_{HOCO}	—	—	30	µs	Figure 35.25
PLL circuit oscillation frequency	f_{PLL}	24	—	40	MHz	
PLL clock oscillation stabilization time	t_{PLL}	—	—	50	µs	Figure 35.26
PLL free-running oscillation frequency	f_{PLLFR}	—	8	—	MHz	

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.

Note 2. Reference values when an 8-MHz resonator is used.
When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the resonator-manufacturer-recommended value.
After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCOVFSR.MOOVF flag to confirm that is has become 1, and then start using the main clock.

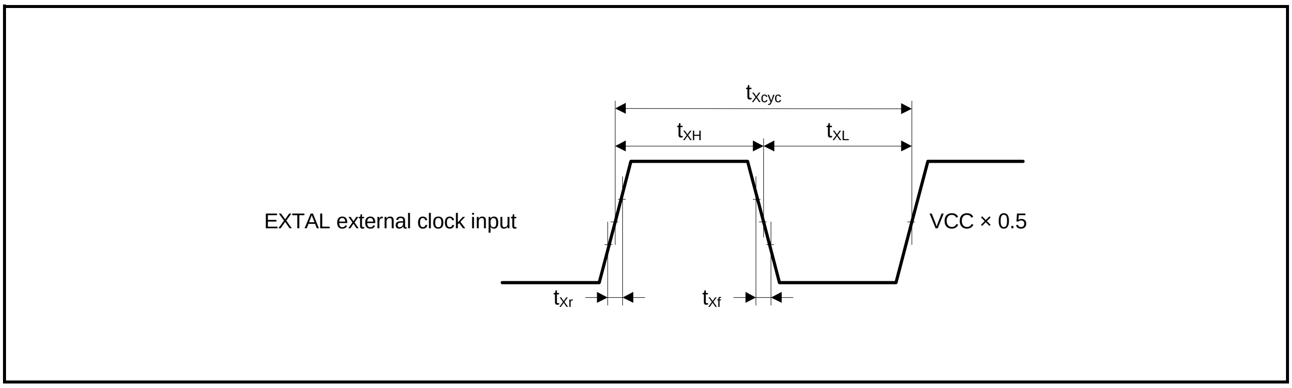


Figure 35.20 EXTAL External Clock Input Timing

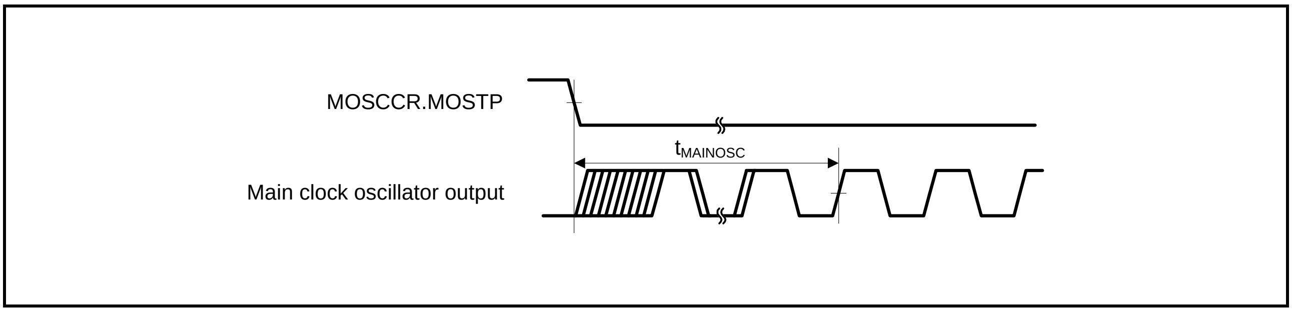


Figure 35.21 Main Clock Oscillation Start Timing

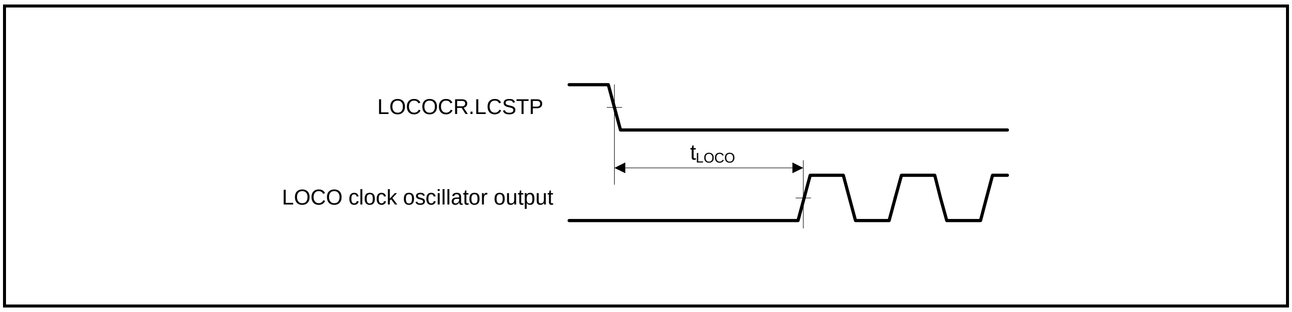


Figure 35.22 LOCO Clock Oscillation Start Timing

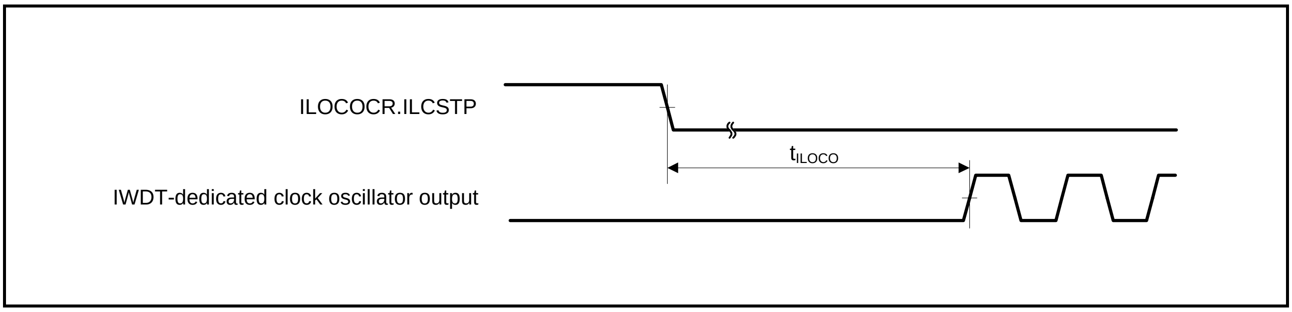


Figure 35.23 IWDT-Dedicated Clock Oscillation Start Timing

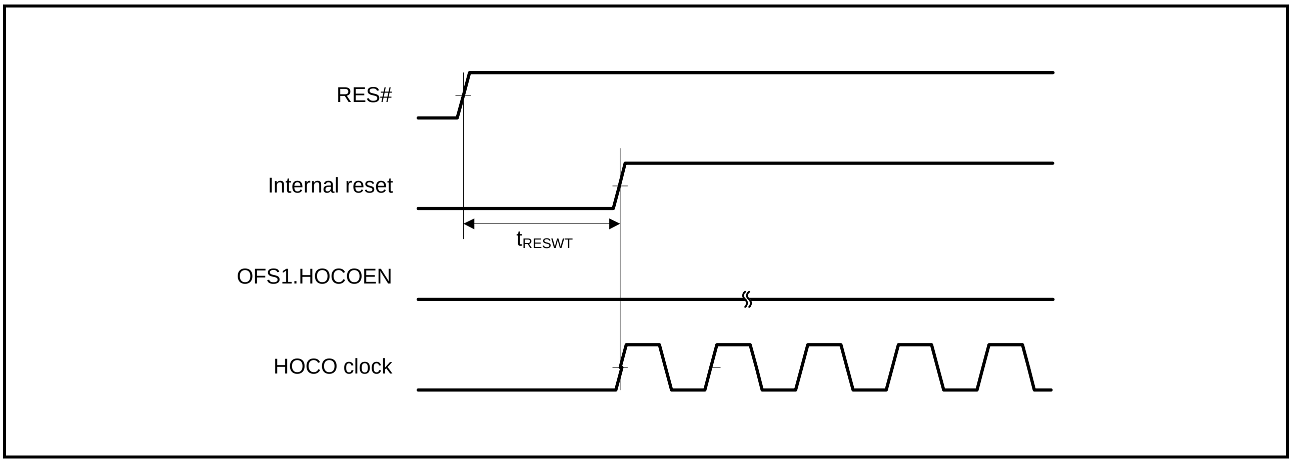


Figure 35.24 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)

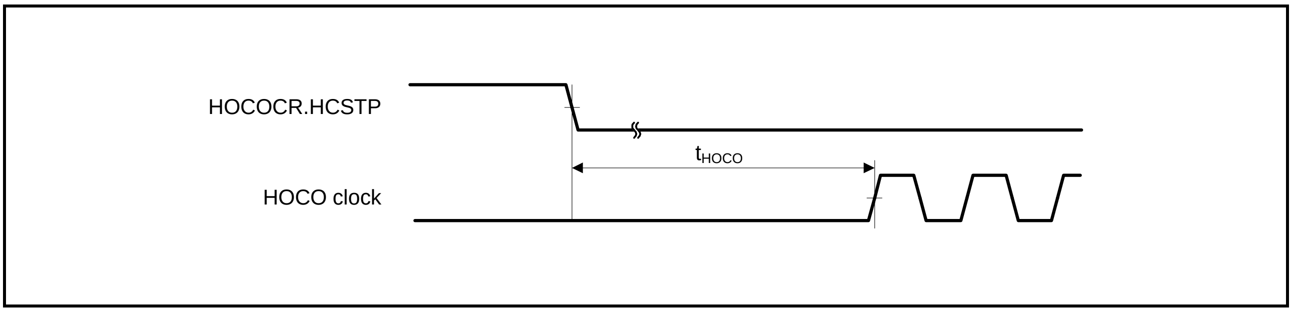


Figure 35.25 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)

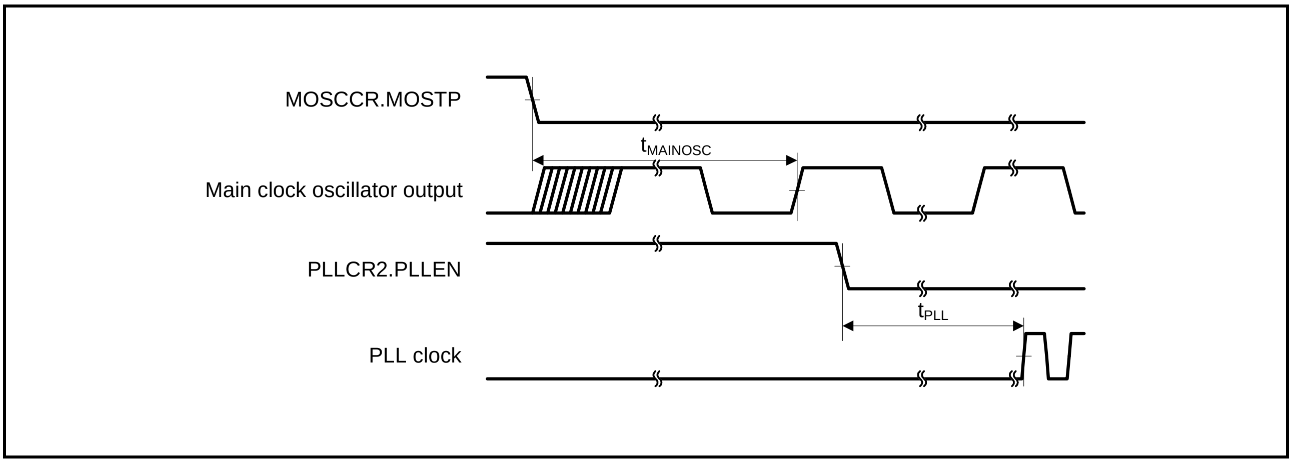


Figure 35.26 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)

35.3.2 Reset Timing

Table 35.17 Reset Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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	At power-on	t_{RESWP}	3	—	—	ms	Figure 35.27
	Other than above	t_{RESW}	30	—	—	μs	Figure 35.28
Wait time after RES# cancellation (at power-on)		t_{RESWT}	—	27.5	—	ms	Figure 35.27
Wait time after RES# cancellation (during powered-on state)		t_{RESWT}	—	114	—	μs	Figure 35.28
Independent watchdog timer reset period		t_{RESWIW}	—	1	—	IWDT clock cycle	Figure 35.29
Software reset period		t_{RESWSW}	—	1	—	ICLK cycle	
Wait time after independent watchdog timer reset cancellation*1		t_{RESW2}	—	300	—	μs	
Wait time after software reset cancellation		t_{RESW2}	—	168	—	μs	

Note 1. When IWDTCR.CKS[3:0] = 0000b.

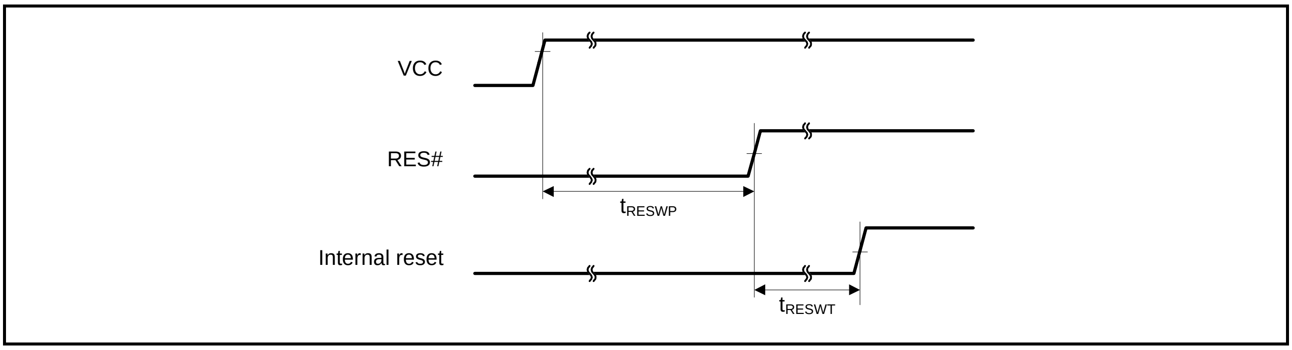


Figure 35.27 Reset Input Timing at Power-On

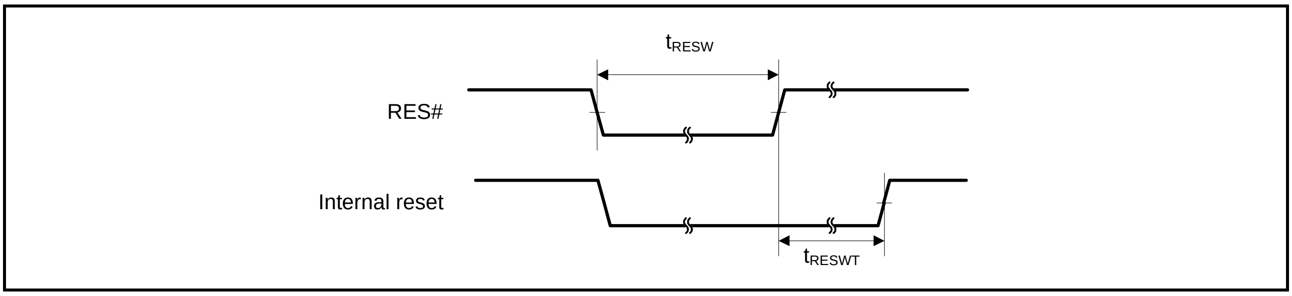


Figure 35.28 Reset Input Timing (1)

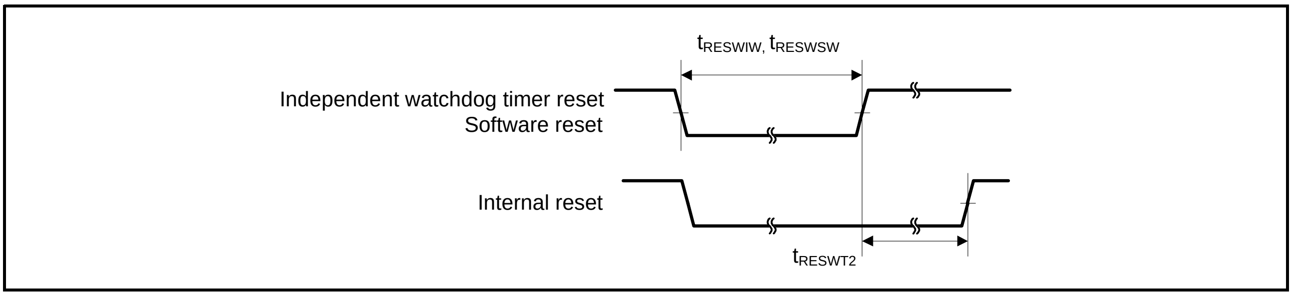


Figure 35.29 Reset Input Timing (2)

35.3.3 Timing of Recovery from Low Power Consumption Modes

Table 35.18 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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 from software standby mode*1	High-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 35.30
		External clock input to main clock oscillator	Main clock oscillator operating*3	t_{SBYEX}	—	35	50	μs	
			Main clock oscillator and PLL circuit operating*4	t_{SBYPE}	—	70	95	μs	
		LOCO clock oscillator operating		t_{SBYLO}	—	40	55	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. The above table applies when only the corresponding clock is operating.

Note 2. When the frequency of crystal is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 3. When the frequency of the external clock is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 4. When the frequency of PLL is 40 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Table 35.19 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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 from software standby mode*1	Middle-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 35.30
		External clock input to main clock oscillator	Main clock oscillator operating*3	t_{SBYEX}	—	3	4	μs	
			Main clock oscillator and PLL circuit operating*4	t_{SBYPE}	—	65	85	μs	
		LOCO clock oscillator operating		t_{SBYLO}	—	5	7	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. The above table applies when only the corresponding clock is operating.

Note 2. When the frequency of the crystal is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 3. When the frequency of the external clock is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 4. When the frequency of PLL is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

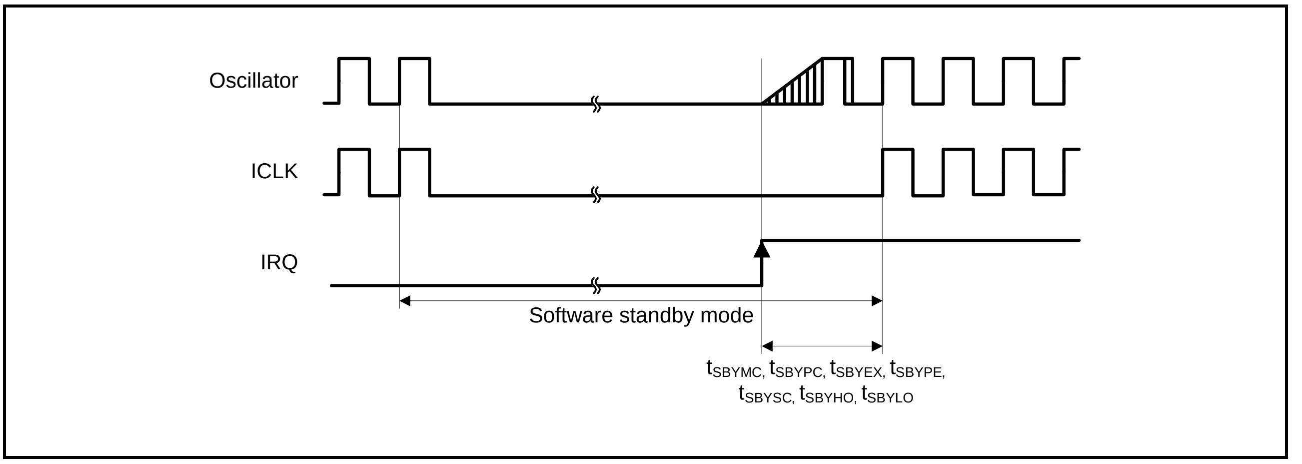


Figure 35.30 Software Standby Mode Recovery Timing

Table 35.20 Timing of Recovery from Low Power Consumption Modes (3)

Conditions: VCC = 2.7 V to AVCC0, AVCC0 = VREFH0 = 2.7 V 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 from deep sleep mode*1	High-speed mode*2	t_{DSLP}	—	2	3.5	µs	Figure 35.31
	Middle-speed mode*3	t_{DSLP}	—	3	4	µs	

Note 1. Oscillators continue oscillating in deep sleep mode.
Note 2. When the frequency of the system clock is 32 MHz.
Note 3. When the frequency of the system clock is 12 MHz.

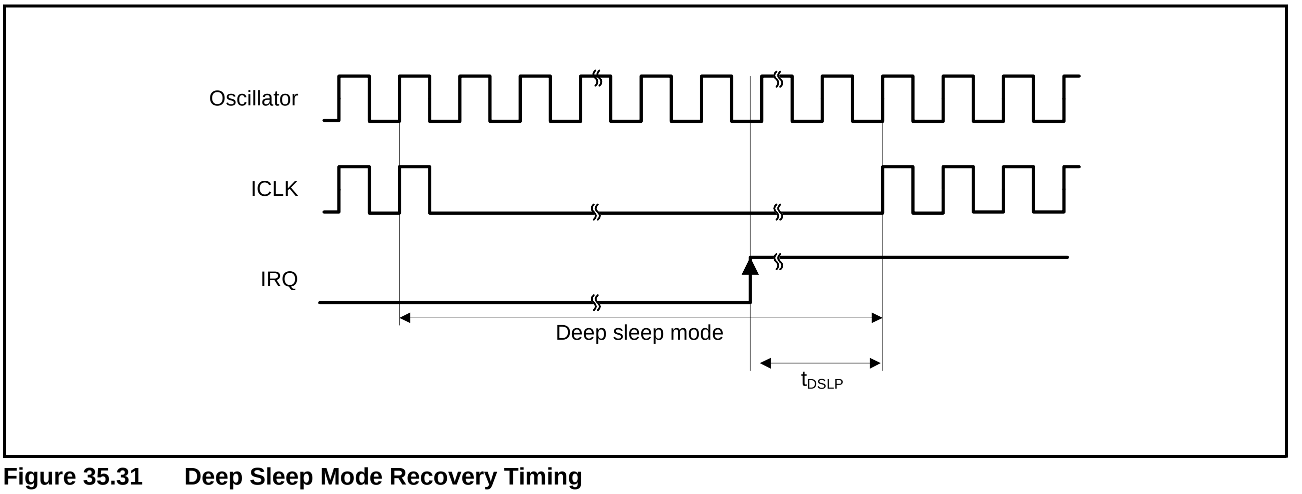


Figure 35.31 Deep Sleep Mode Recovery Timing

Table 35.21 Operating Mode Transition Time

Conditions: VCC = 2.7 V to AVCC0, AVCC0 = VREFH0 = 2.7 V to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Mode before Transition	Mode after Transition	ICLK Frequency	Transition Time			Unit
			Min.	Typ.	Max.	
High-speed operating mode	Middle-speed operating modes	8 MHz	—	10	—	µs
Middle-speed operating modes	High-speed operating mode	8 MHz	—	37.5	—	µs

Note: Values when the frequencies of PCLKB, PCLKD, and FCLK are not divided.

35.3.4 Control Signal Timing

Table 35.22 Control Signal Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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	NMI digital filter disabled (NMIFLTE.NFLTEN = 0)	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		NMI digital filter enabled (NMIFLTE.NFLTEN = 1)	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$*2	—	—			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled (IRQFLTE0.FLTENi = 0)	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		IRQ digital filter enabled (IRQFLTE0.FLTENi = 1)	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$*3	—	—			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in software standby mode.
Note 1. t_{Pcyc} indicates the cycle of PCLKB.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 5).

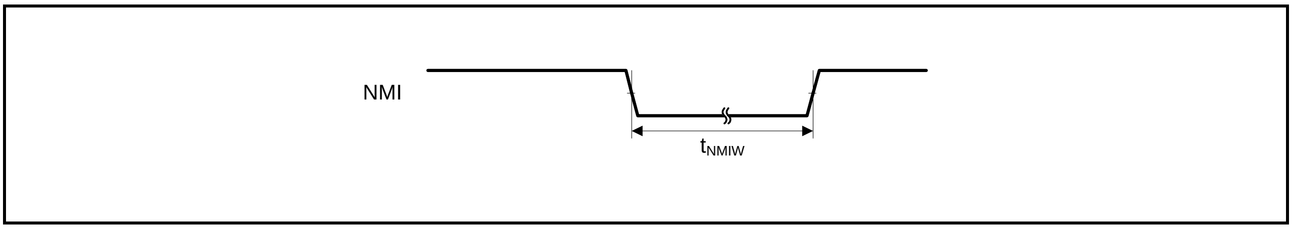


Figure 35.32 NMI Interrupt Input Timing

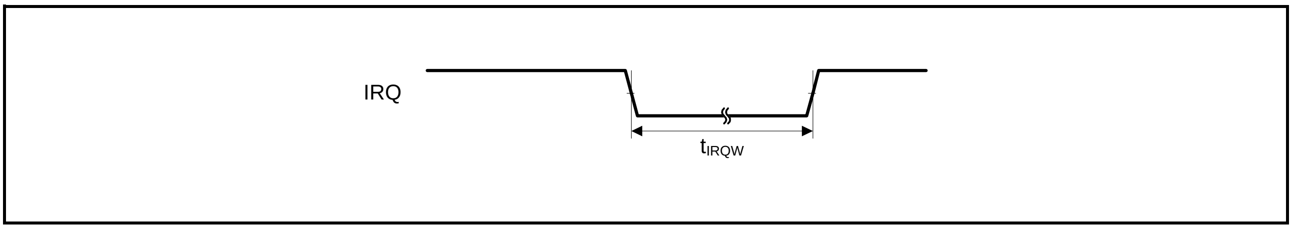


Figure 35.33 IRQ Interrupt Input Timing

35.3.5 Timing of On-Chip Peripheral Modules

Table 35.23 Timing of On-Chip Peripheral Modules (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC 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 35.34
MTU3	Input capture input pulse width		Single-edge setting	t_{TICW}	3	—	t_{PAcyc}	Figure 35.35
			Both-edge setting		5	—		
	Timer clock pulse width		Single-edge setting	t_{TCKWH}, t_{TCKWL}	3	—	t_{PAcyc}	Figure 35.36
			Both-edge setting		5	—		
			Phase counting mode		5	—		
POE3	POE# input pulse width			t_{POEW}	1.5	—	t_{Pcyc}	Figure 35.37
TMR	Timer clock pulse width		Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{Pcyc}	Figure 35.38
			Both-edge setting		2.5	—		
SCI	Input clock cycle		Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 35.39
			Clock synchronous		6	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20	ns	
	Output clock cycle		Asynchronous	t_{Scyc}	16	—	t_{Pcyc}	Figure 35.40
			Clock synchronous		4	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time			t_{SCKr}	—	20	ns	
	Output clock fall time			t_{SCKf}	—	20	ns	
	Transmit data delay time (master)	Clock synchronous		t_{TXD}	—	40	ns	
	Transmit data delay time (slave)	Clock synchronous	VCC = 4.0 V or above		—	40	ns	
			VCC = 2.7 V or above		—	65	ns	
	Receive data setup time (master)	Clock synchronous	VCC = 4.0 V or above	t_{RXS}	40	—	ns	
			VCC = 2.7 V or above		65	—	ns	
	Receive data setup time (slave)	Clock synchronous			40	—	ns	
	Receive data hold time	Clock synchronous		t_{RXH}	40	—	ns	
A/D converter	Trigger input pulse width			t_{TRGW}	1.5	—	t_{Pcyc}	Figure 35.41
CAC	CACREF input pulse width		$t_{Pcyc} \leq t_{cac}$ *2	t_{CACREF}	4.5 t_{cac} + 3 t_{Pcyc}	—	ns	
			$t_{Pcyc} > t_{cac}$ *2		5 t_{cac} + 6.5 t_{Pcyc}			

Note 1. t_{Pcyc}: PCLK cycle, t_{PAcyc}: PCLKA cycle

Note 2. t_{cac}: CAC count clock source cycle

Table 35.24 Timing of On-Chip Peripheral Modules (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C, C = 30pF

Item				Symbol	Min.	Max.	Unit	Test Conditions
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	4096	t_{Pcyc}*1	Figure 35.42
		Slave			8	4096		
	RSPCK clock high pulse width	Master	VCC = 4.0 V or above	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 5$	—	ns	
			VCC = 2.7 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 8$	—		
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock low pulse width	Master	VCC = 4.0 V or above	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 5$	—	ns	
			VCC = 2.7 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 8$	—		
		Slave			$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$	—		
	RSPCK clock rise/fall time	Output	VCC = 4.0 V or above	t_{SPCKr}, t_{SPCKf}	—	6	ns	
			VCC = 2.7 V or above		—	10		
		Input			—	0.1	µs/V	
	Data input setup time	Master	VCC = 4.0 V or above	t_{SU}	10	—	ns	Figure 35.43 to Figure 35.46
			VCC = 2.7 V or above		26	—		
		Slave			$25 - t_{Pcyc}$	—		
	Data input hold time	Master	RSPCK set to a division ratio other than PCLKB divided by 2	t_H	t_{Pcyc}	—	ns	
			RSPCK set to PCLKB divided by 2	t_{HF}	0	—		
		Slave		t_H	$20 + 2 \times t_{Pcyc}$	—		
	SSL setup time	Master		t_{LEAD}	$-30 + N^{*2} \times t_{SPcyc}$	—	ns	
		Slave			2	—	t_{Pcyc}	
	SSL hold time	Master		t_{LAG}	$-30 + N^{*3} \times t_{SPcyc}$	—	ns	
		Slave			2	—	t_{Pcyc}	
	Data output delay time	Master	VCC = 4.0 V or above	t_{OD}	—	10	ns	
			VCC = 2.7 V or above		—	14		
		Slave			—	$3 \times t_{Pcyc} + 65$		
	Data output hold time	Master	2.7 V or above	t_{OH}	0	—	ns	
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
		Slave			$4 \times t_{Pcyc}$	—		
	MOSI and MISO rise/fall time	Output		t_{Dr}, t_{Df}	—	10	ns	
		Input			—	1	µs	
	SSL rise/fall time	Output		t_{SSLr}, t_{SSLf}	—	10	ns	
		Input			—	1	µs	
	Slave access time			t_{SA}	—	6	t_{Pcyc}	Figure 35.45, Figure 35.46
	Slave output release time			t_{REL}	—	5	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLK cycle

Note 2. N: An integer from 1 to 8 that can be set by the RSPI clock delay register (SPCKD)

Note 3. N: An integer from 1 to 8 that can be set by the RSPI slave select negation delay register (SSLND)

Table 35.25 Timing of On-Chip Peripheral Modules (3)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C, C = 30pF

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	4	65536	t_{Pcyc}	Figure 35.42
	SCK clock cycle input (slave)			6	65536	t_{Pcyc}	
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time		t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time (master)	VCC = 4.0 V or above	t_{SU}	40	—	ns	Figure 35.43, Figure 35.44
		VCC = 2.7 V or above		65	—		
	Data input setup time (slave)			40	—		
	Data input hold time		t_H	40	—	ns	
	SS input setup time		t_{LEAD}	3	—	t_{SPcyc}	
	SS input hold time		t_{LAG}	3	—	t_{SPcyc}	
	Data output delay time (master)		t_{OD}	—	40	ns	
	Data output delay time (slave)	VCC = 4.0 V or above		—	40		
		VCC = 2.7 V or above		—	65		
	Data output hold time (master)	Master	t_{OH}	–10	—	ns	
		Slave		–10	—		
	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}	—	6	t_{Pcyc}	Figure 35.45, Figure 35.46
	Slave output release time		t_{REL}	—	6	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLK cycle

Table 35.26 Timing of On-Chip Peripheral Modules (4)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.*1, *2	Max.	Unit	Test Conditions
RIIC (Standard mode, SMBus)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 35.47
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	1000	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	1000	—	ns	
	STOP condition setup time	t_{STOS}	1000	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
RIIC (Fast mode)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	—	ns	Figure 35.47
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA rise time	t_{Sr}	—	300	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition setup time	t_{STAS}	300	—	ns	
	STOP condition setup time	t_{STOS}	300	—	ns	
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note 1. t_{IICcyc}: RIIC internal reference count clock (IICφ) cycle

Note 2. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.

Table 35.27 Timing of On-Chip Peripheral Modules (5)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.*2	Max.	Unit	Test Conditions
Simple I^2C (Standard mode)	SDA rise time	t_{Sr}	—	1000	ns	Figure 35.47
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{pcyc}*1	ns	
	Data setup time	t_{SDAS}	250	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
Simple I^2C (Fast mode)	SDA rise time	t_{Sr}	—	300	ns	Figure 35.47
	SDA fall time	t_{Sf}	—	300	ns	
	SDA spike pulse removal time	t_{SP}	0	4 × t_{pcyc}*1	ns	
	Data setup time	t_{SDAS}	100	—	ns	
	Data hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note 1. t_{Pcyc}: PCLK cycle

Note 2. C_b is the total capacitance of the bus lines.

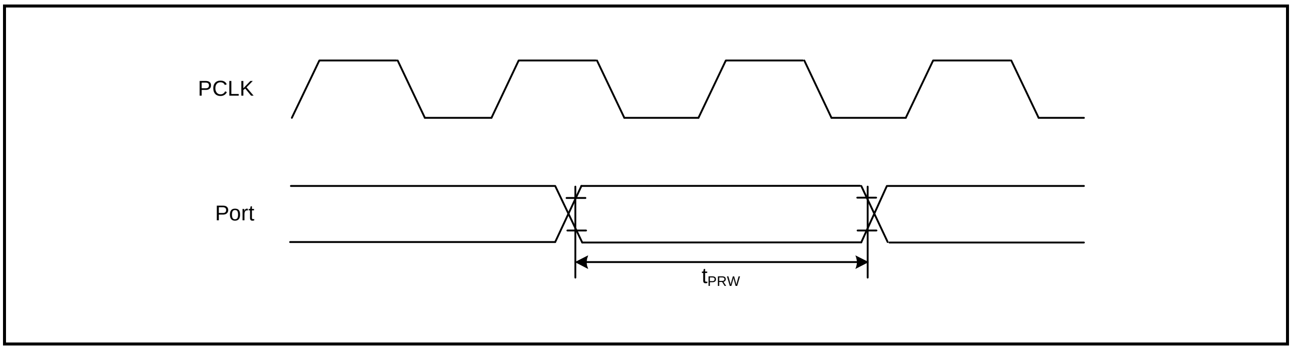


Figure 35.34 I/O Port Input Timing

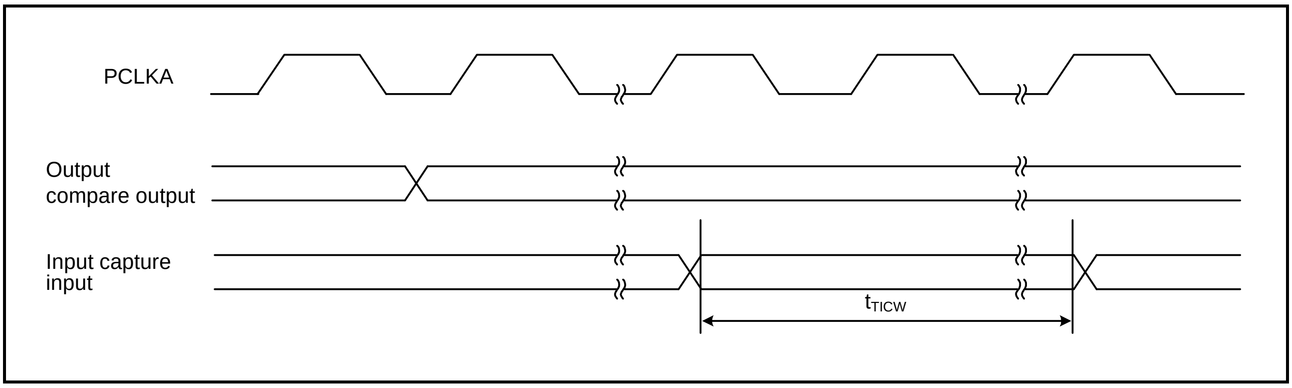


Figure 35.35 MTU3 Input/Output Timing

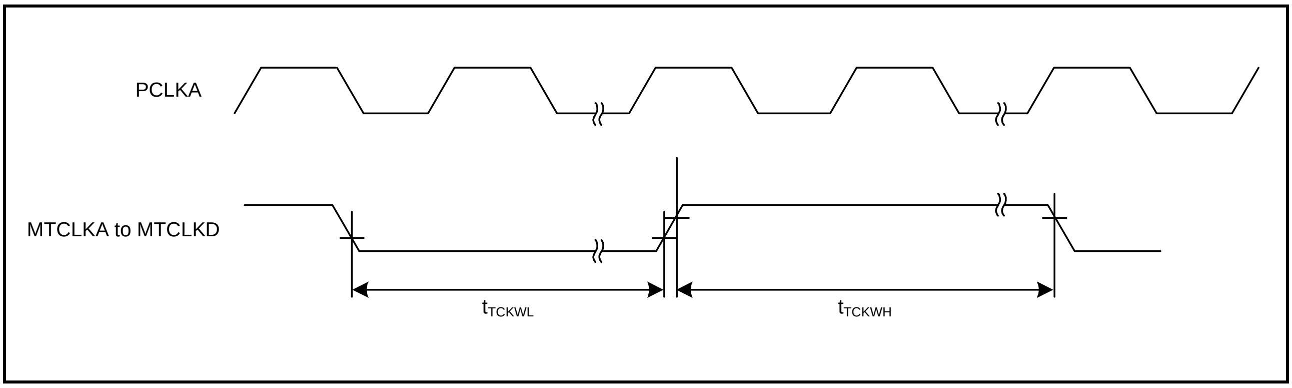


Figure 35.36 MTU3 Clock Input Timing

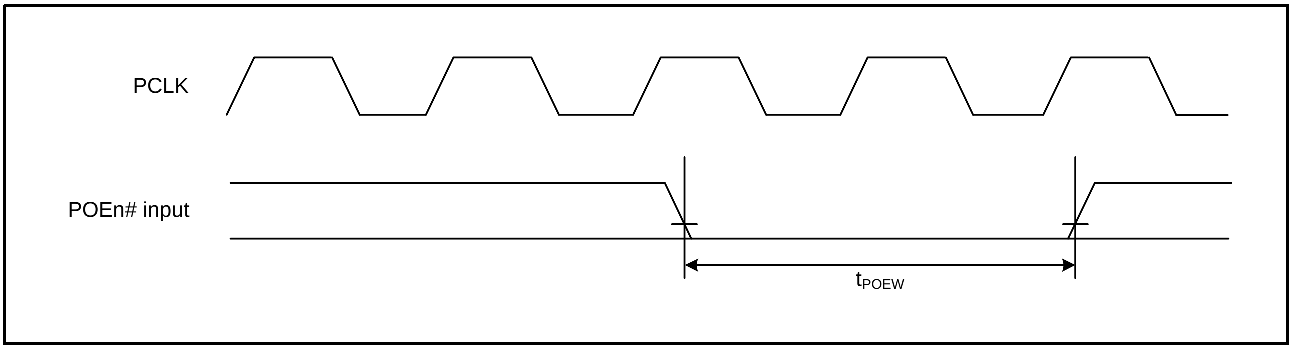


Figure 35.37 POE# Input Timing

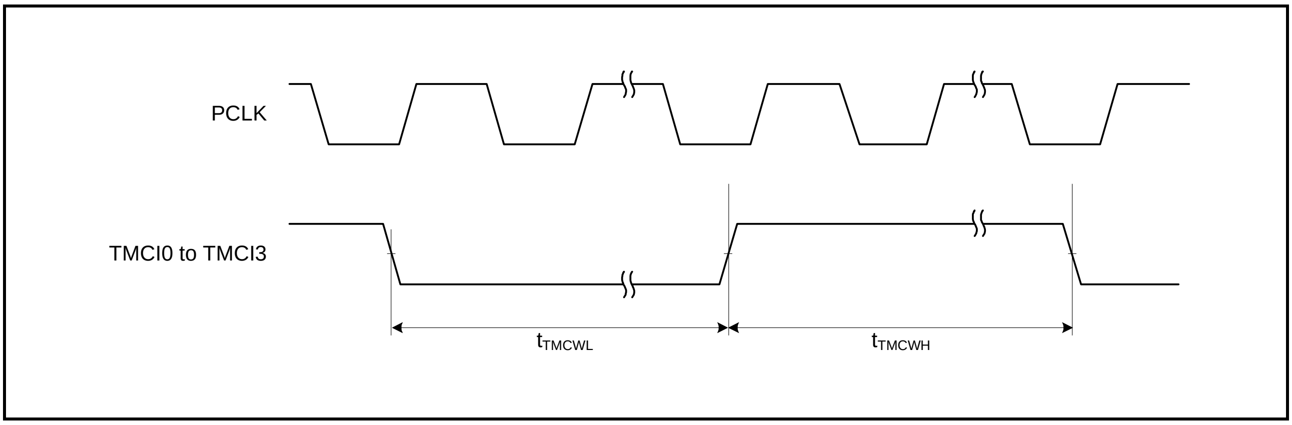


Figure 35.38 TMR Clock Input Timing

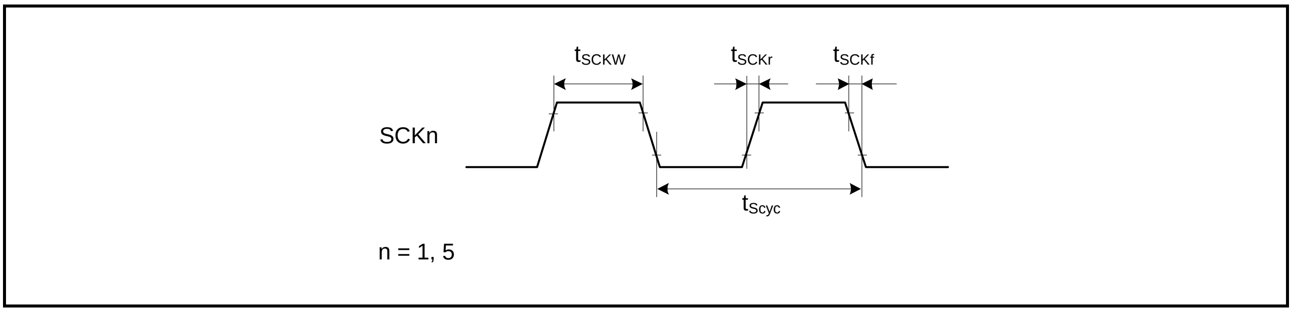


Figure 35.39 SCK Clock Input Timing

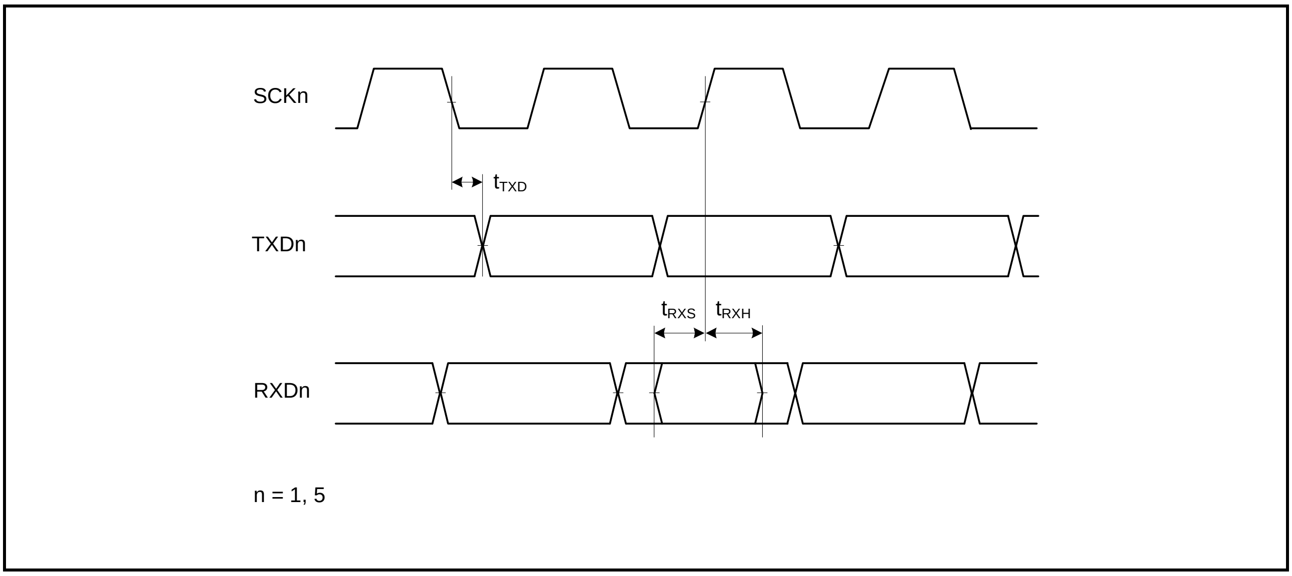


Figure 35.40 SCI Input/Output Timing: Clock Synchronous Mode

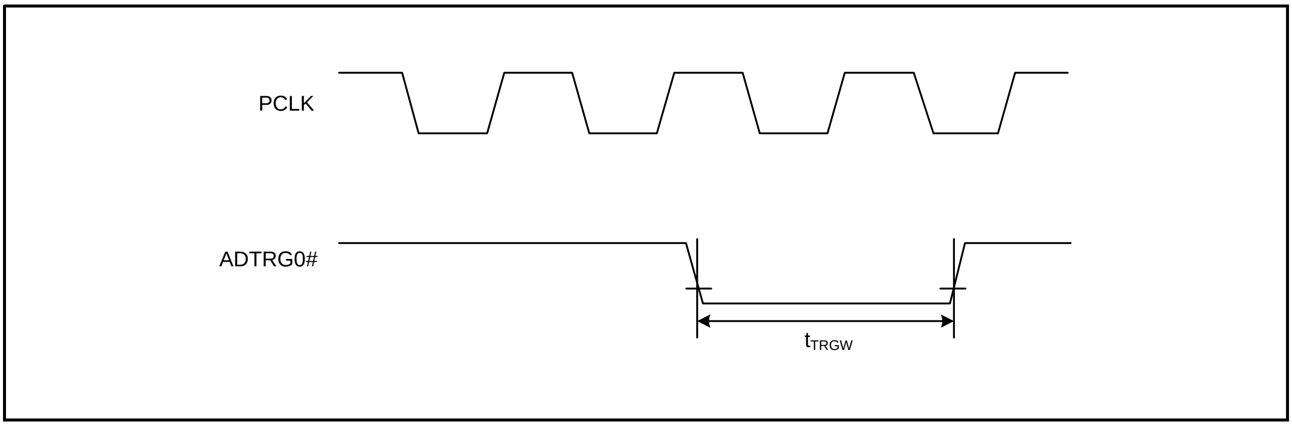


Figure 35.41 A/D Converter External Trigger Input Timing

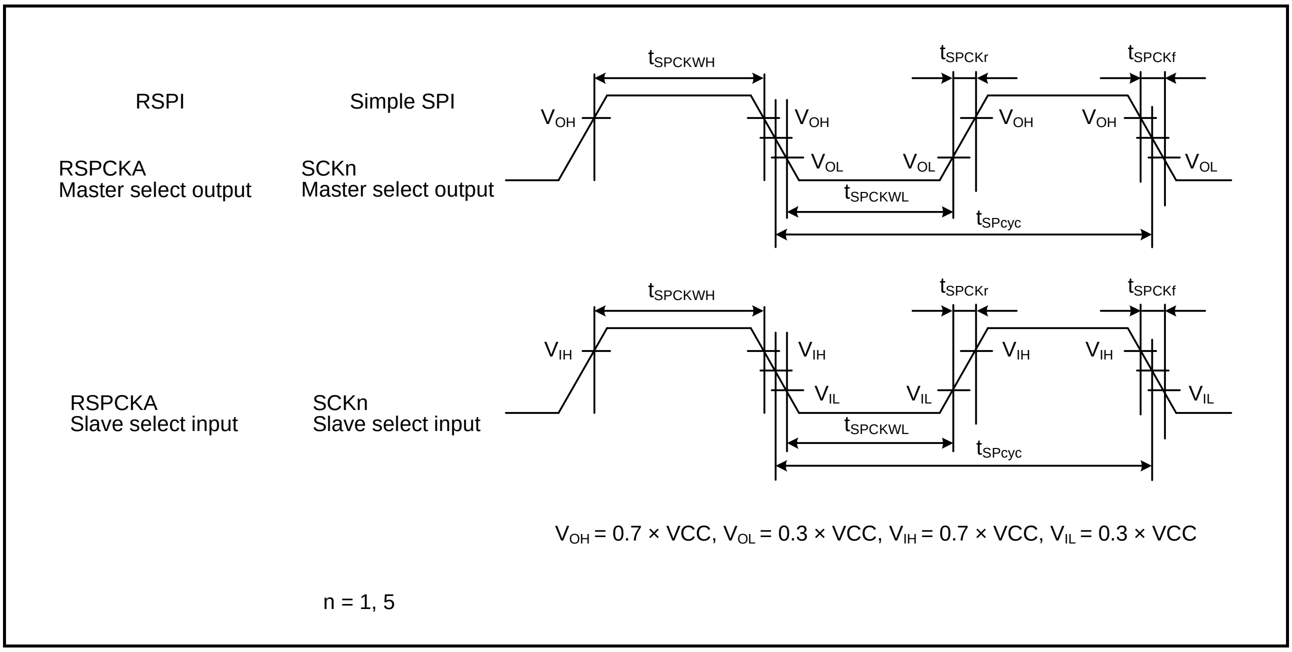


Figure 35.42 RSPI Clock Timing and Simple SPI Clock Timing

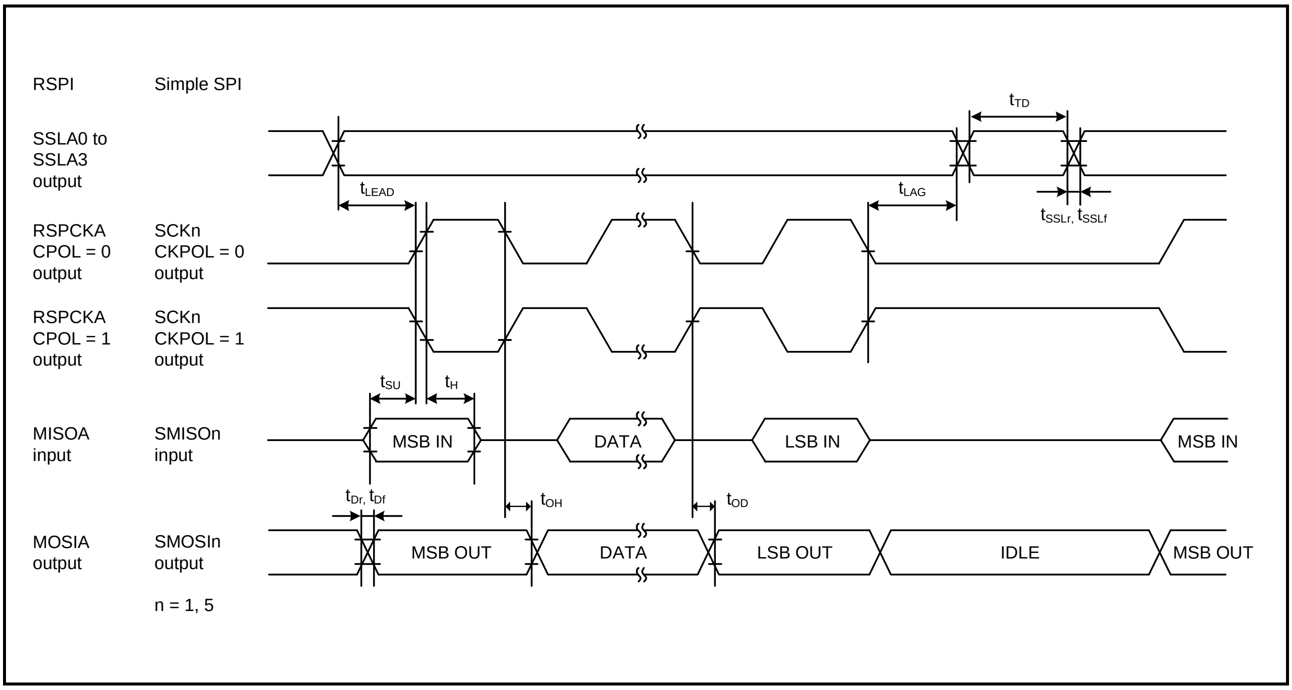


Figure 35.43 RSPI Timing (Master, CPHA = 0) and Simple SPI Clock Timing (Master, CKPH = 1)

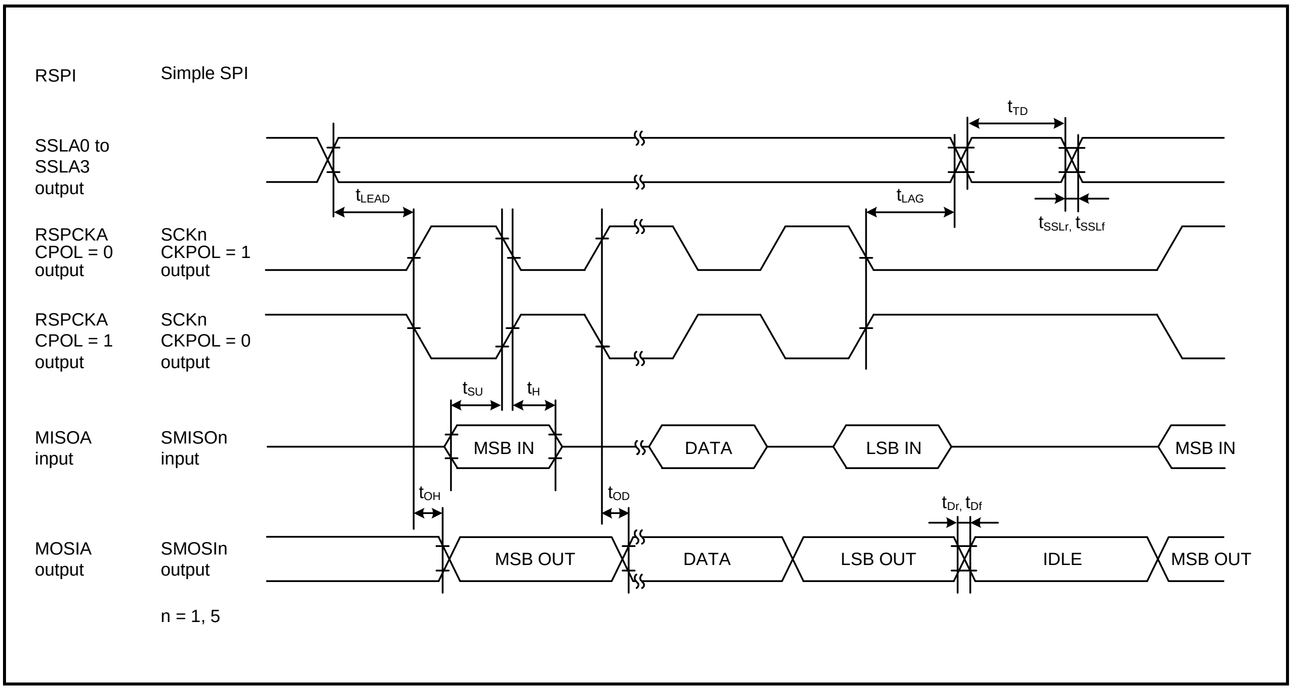


Figure 35.44 RSPI Timing (Master, CPHA = 1) and Simple SPI Clock Timing (Master, CKPH = 0)

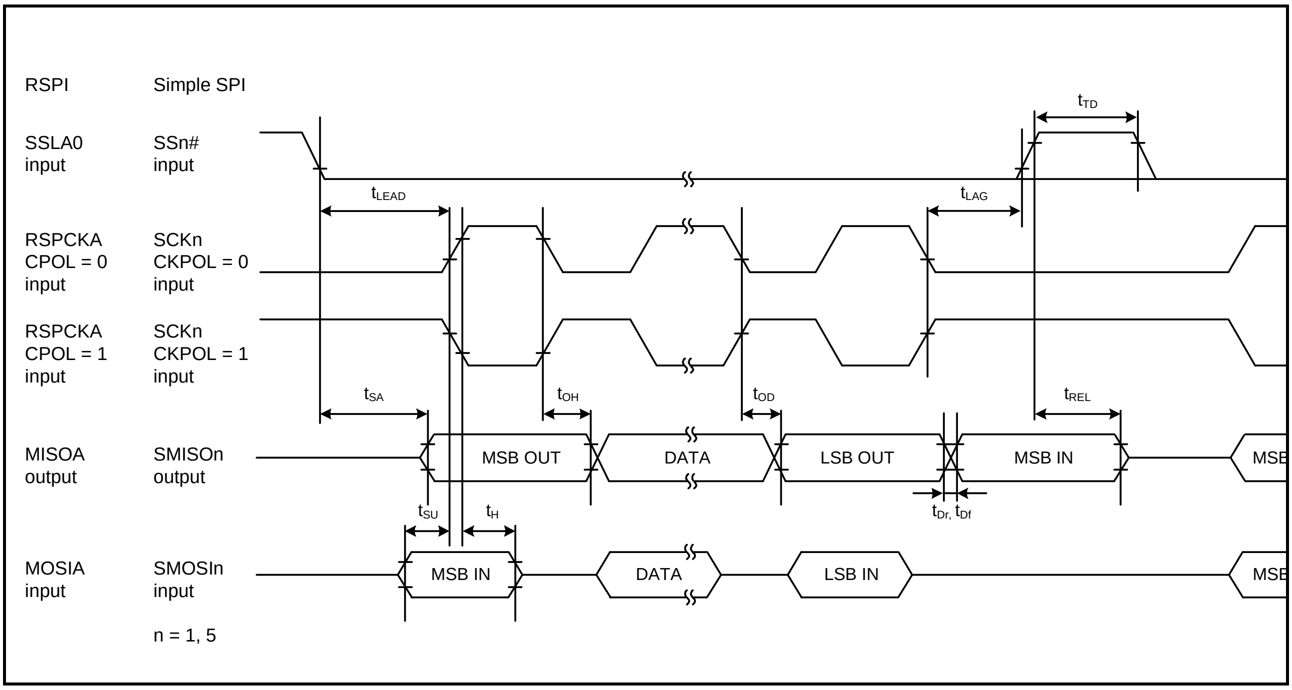


Figure 35.45 RSPI Timing (Slave, CPHA = 0) and Simple SPI Clock Timing (Slave, CKPH = 1)

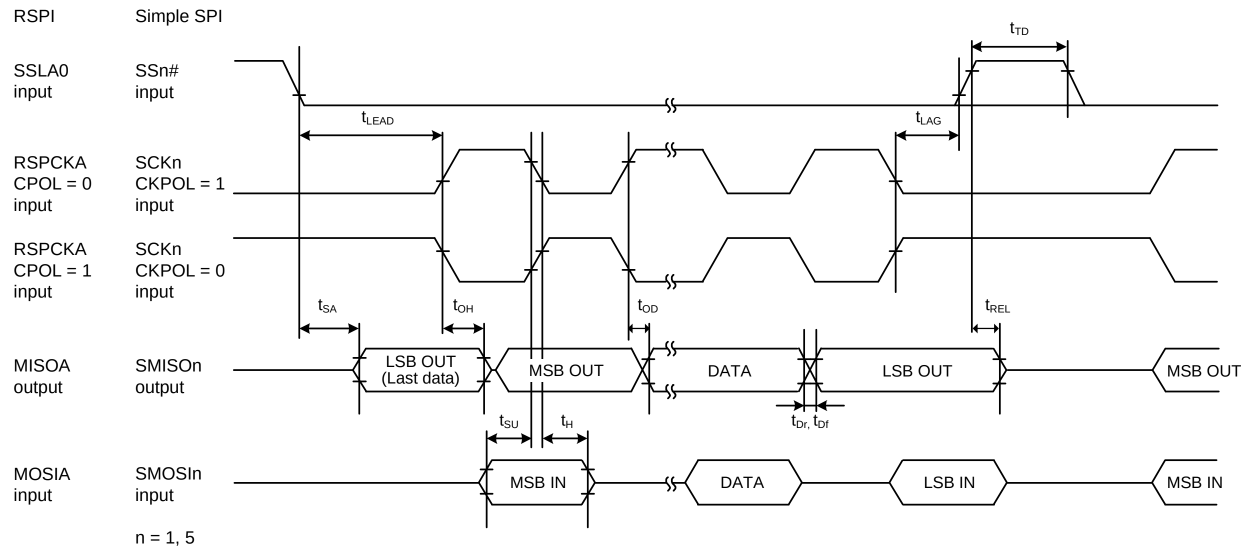
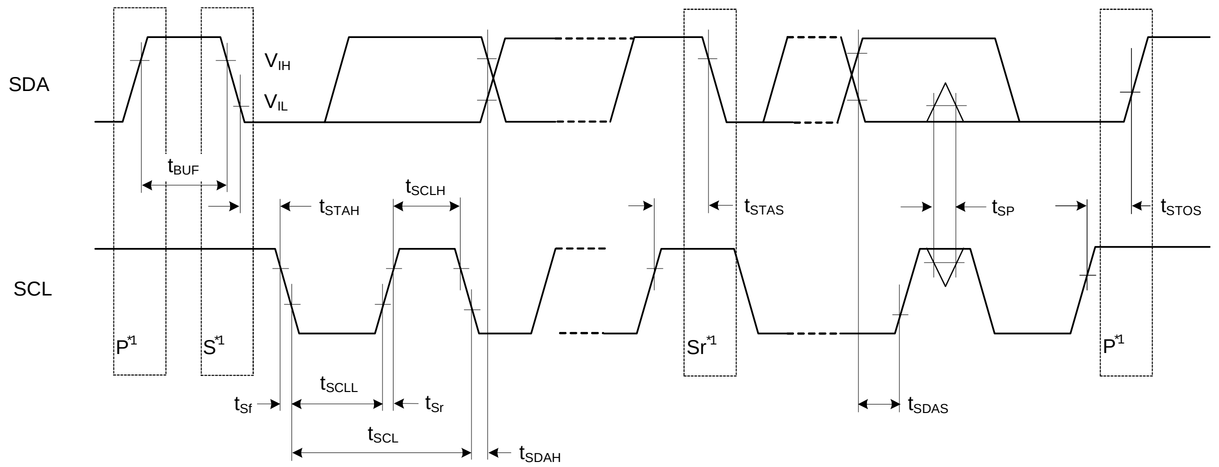


Figure 35.46 RSPI Timing (Slave, CPHA = 1) and Simple SPI Clock Timing (Slave, CKPH = 0)



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Test conditions
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3

Figure 35.47 RIIC Bus Interface Input/Output Timing and Simple I²C Bus Interface Input/Output Timing

35.4 A/D Conversion Characteristics

Table 35.28 A/D Conversion Characteristics (1)

Conditions: VCC = 4.5 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	40	MHz	
Resolution		—	—	12	Bit	
Conversion time*[1] (Operation at PCLKD = 40 MHz)	Permissible signal source impedance (Max.) = 1.0 kΩ Sample-and-hold circuit not in use	1.00	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 08h
		1.25	—	—	μs	Normal-precision channel ADSSTRn.SST[7:0] bits = 12h
	Permissible signal source impedance (Max.) = 1.0 kΩ / Sample-and-hold circuit in use	1.65	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 08h ADSHCR.SSTSH[7:0] bits = 0Dh AN000 to AN002 = 0.25 V to VREFH0 – 0.25 V
Analog input capacitance		—	—	12	pF	
Offset error		—	—	±6.5	LSB	
Full-scale error		—	—	±6.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	—	±8.0	LSB	
DNL differential nonlinearity error		—	±0.5	±1.5	LSB	
INL integral nonlinearity error		—	±2.0	±4.0	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.

Table 35.29 A/D Conversion Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	40	MHz	
Resolution		—	—	12	Bit	
Conversion time*[1] (Operation at PCLKD = 40 MHz)	Permissible signal source impedance (Max.) = 1.0 kΩ Sample-and-hold circuit not in use	1.15	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 0Eh
		1.30	—	—	μs	Normal-precision channel ADSSTRn.SST[7:0] bits = 14h
	Permissible signal source impedance (Max.) = 1.0 kΩ Sample-and-hold circuit in use	1.90	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 0Eh ADSHCR.SSTSH[7:0] bits = 11h AN000 to AN002 = 0.25 V to VREFH0 – 0.25 V
Analog input capacitance		—	—	12	pF	
Offset error		—	—	±6.5	LSB	
Full-scale error		—	—	±6.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	—	±8.0	LSB	
DNL differential nonlinearity error		—	±0.5	±1.5	LSB	
INL integral nonlinearity error		—	±2.0	±4.0	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.

Table 35.30 A/D Converter Channel Classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN007	AVCC0 = 2.7 to 5.5 V	Pins AN000 to AN007 cannot be used as digital outputs when the A/D converter is in use.
Normal-precision channel	AN016, AN017	VCC = AVCC0 = 2.7 to 5.5 V	
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.7 to 5.5 V	

Table 35.31 A/D Internal Reference Voltage Characteristics

Conditions: VCC = 2.7 V to AVCC0, AVCC0 = VREFH0 = 2.7 V to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Internal reference voltage input channel*[1]	1.36	1.43	1.50	V	

Note 1. The A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the A/D converter.

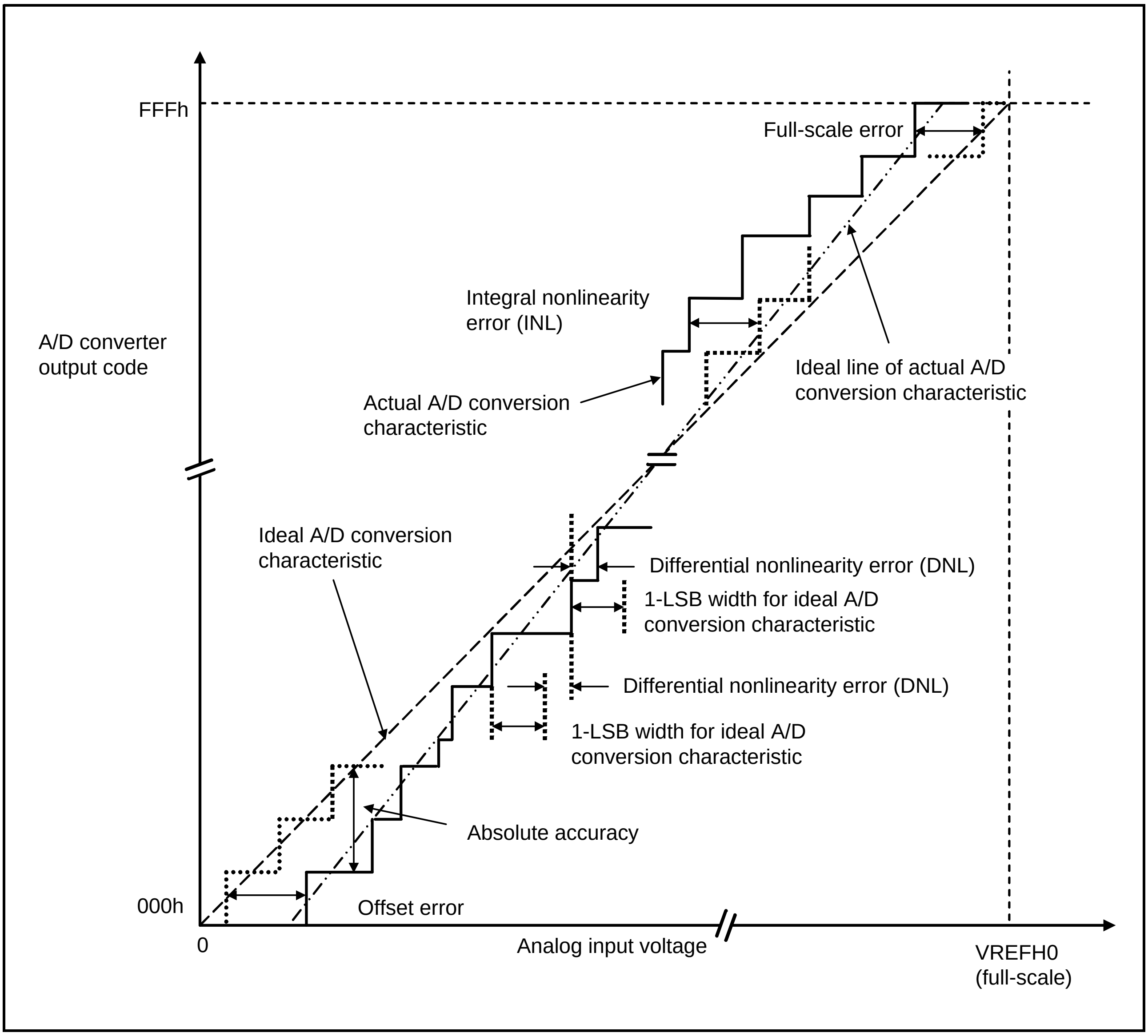


Figure 35.48 Illustration of A/D Converter Characteristic Terms

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 = 3.072 V), then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, ... are used as analog input voltages.

If analog input voltage is 6 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 actual 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.

35.5 Comparator Characteristics

Table 35.32 Comparator Characteristics

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Offset voltage	V_{cioff}	—	—	40	mV	
Reference input voltage range	V_{cref}	0	—	AVCC0	V	
Response time	t_{cr}	—	—	200	ns	VOD = 100 mV CMPCTL.CDFS = 0
	t_{cf}	—	—	200	ns	
Stabilization wait time for input selection	t_{cwait}	300	—	—	ns	
Operation stabilization wait time	t_{cmp}		—	1	μs	

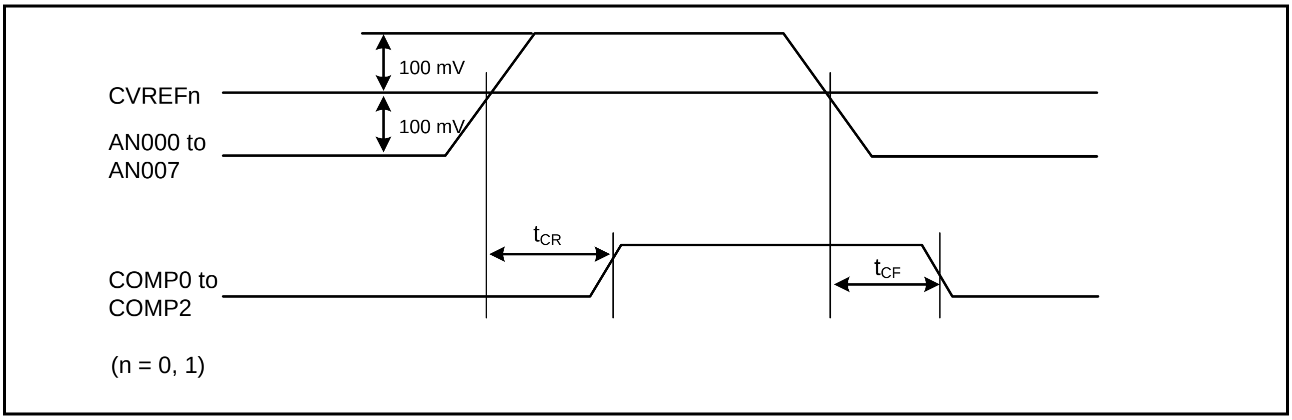


Figure 35.49 Comparator Response Time

35.6 D/A Conversion Characteristics

Table 35.33 D/A Conversion Characteristics

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	—	8	Bit	
Conversion time	t_{DCONV}	—	—	3.0	μs	
Absolute accuracy	—	—	±1.0	±3.0	LSB	

35.7 Power-On Reset Circuit and Voltage Detection Circuit Characteristics

Table 35.34 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, 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)	V_{POR}	1.35	1.50	1.65	V	Figure 35.50, Figure 35.51
	Voltage detection circuit (LVD0)*1	V_{det0_0}	3.67	3.84	3.97	V	
		V_{det0_2}	2.37	2.51	2.67		
	Voltage detection circuit (LVD1)*2	V_{det1_0}	4.12	4.29	4.42	V	Figure 35.53 At falling edge VCC
		V_{det1_1}	3.98	4.14	4.28		
		V_{det1_2}	3.86	4.02	4.16		
		V_{det1_3}	3.68	3.84	3.98		
		V_{det1_4}	2.99	3.10	3.29		
		V_{det1_5}	2.89	3.00	3.19		
		V_{det1_6}	2.79	2.90	3.09		
		V_{det1_7}	2.68	2.79	2.98		
		V_{det1_8}	2.57	2.68	2.87		
	Voltage detection circuit (LVD2)*1	V_{det2_0}*2	4.08	4.29	4.48		Figure 35.54 At falling edge VCC
		V_{det2_1}	3.95	4.14	4.35		
		V_{det2_2}	3.82	4.02	4.22		
		V_{det2_3}	3.62	3.84	4.02		

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD2), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. n in the symbol Vdet0_n denotes the value of the LVDS0[1:0] bits.

Note 2. n in the symbol Vdet1_n denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

Note 3. n in the symbol Vdet2_n denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.

Table 35.35 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, Ta = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Wait time after power-on reset cancellation	t_{POR}	—	28.4	—	ms	Figure 35.51
Wait time after voltage monitoring 0 reset cancellation	t_{LVD0}	—	568	—	μs	Figure 35.52
Wait time after voltage monitoring 1 reset cancellation	t_{LVD1}	—	100	—	μs	Figure 35.53
Wait time after voltage monitoring 2 reset cancellation	t_{LVD2}	—	100	—	μs	Figure 35.54
Response delay time	t_{det}	—	—	350	μs	Figure 35.50
Minimum VCC down time*1	t_{VOFF}	350	—	—	μs	Figure 35.50, VCC = 1.0 V or above
Power-on reset enable time	$t_{W(POR)}$	1	—	—	ms	Figure 35.51, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)	$Td_{(E-A)}$	—	—	300	μs	Figure 35.53, Figure 35.54
Hysteresis width (LVD1 and LVD2)	V_{LVH}	—	70	—	mV	Vdet1_0 to 4 selected
		—	60	—		Vdet1_5 to 8, LVD2 selected

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD1), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/LVD.

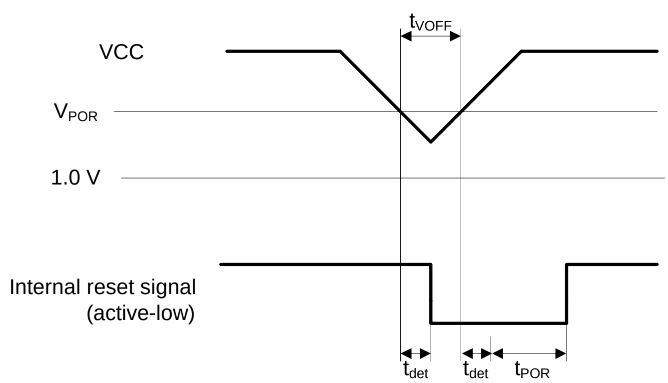
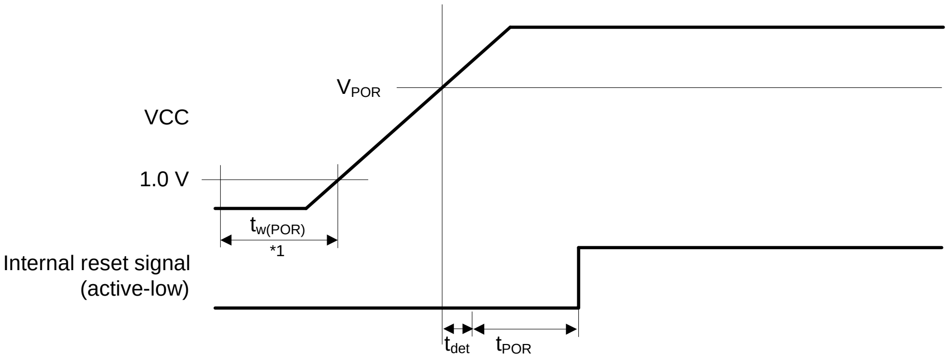


Figure 35.50 Voltage Detection Reset Timing



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 (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 35.51 Power-On Reset Timing

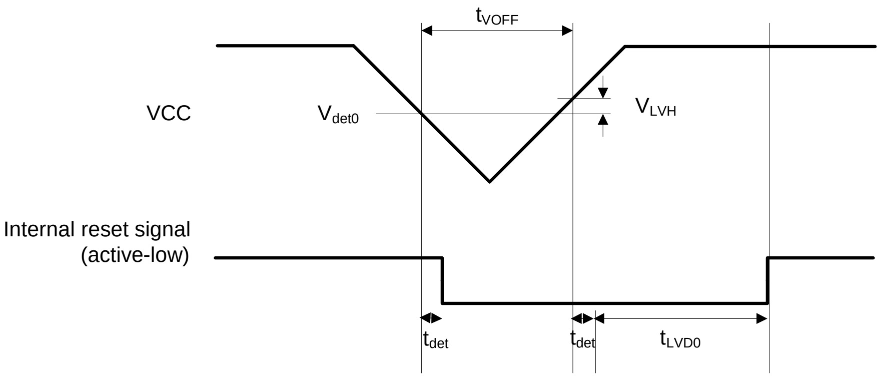


Figure 35.52 Voltage Detection Circuit Timing (Vdet0)

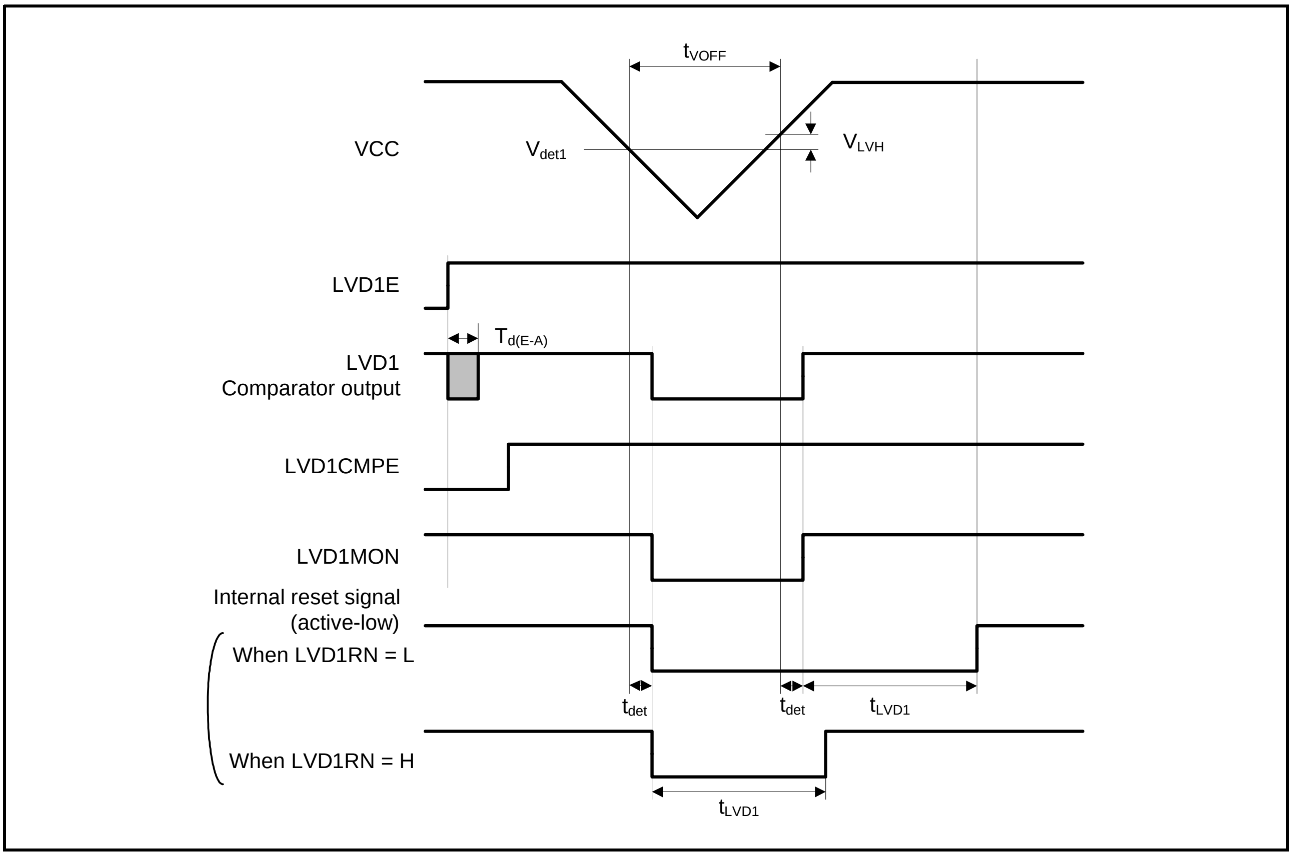


Figure 35.53 Voltage Detection Circuit Timing (V_{det1})

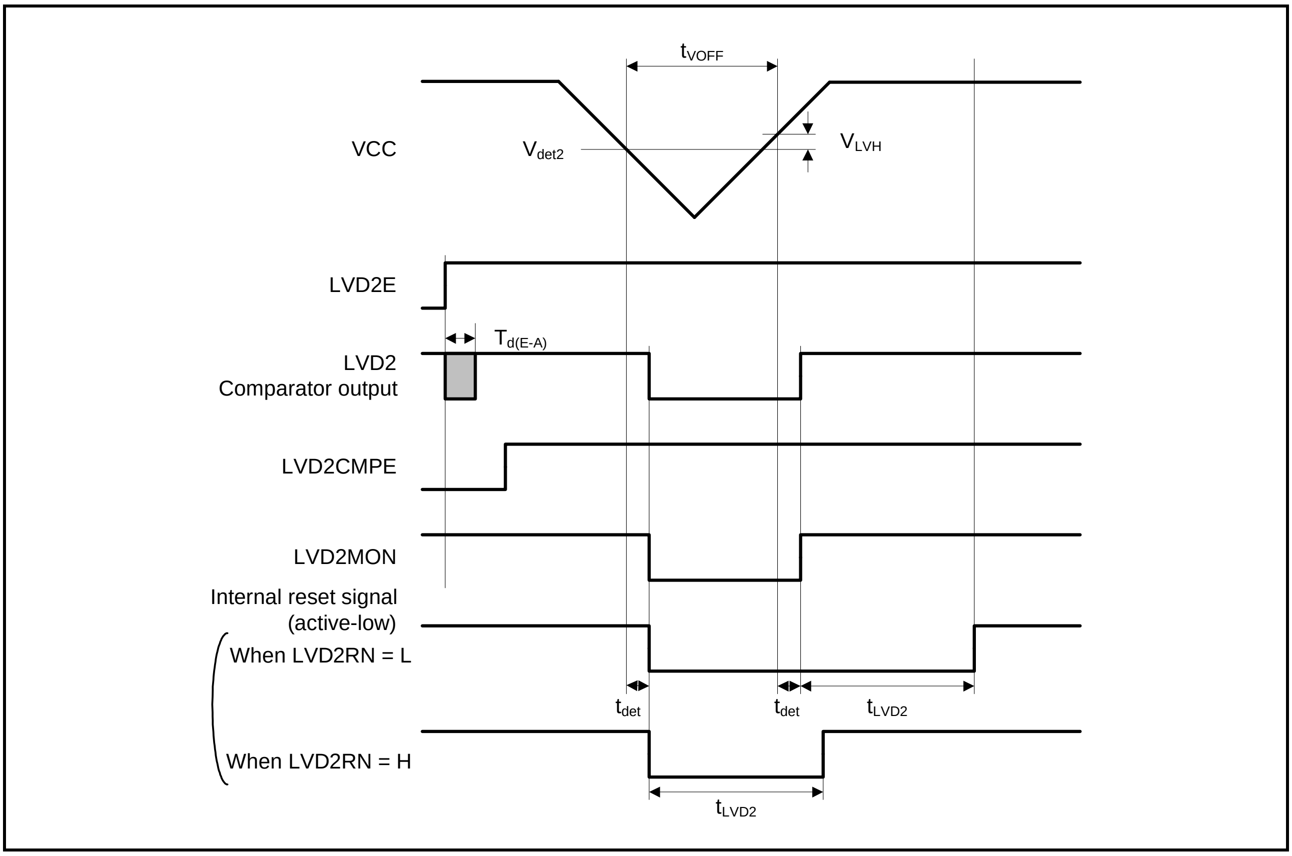


Figure 35.54 Voltage Detection Circuit Timing (V_{det2})

35.8 Oscillation Stop Detection Timing

Table 35.36 Oscillation Stop Detection Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, Ta = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 35.55

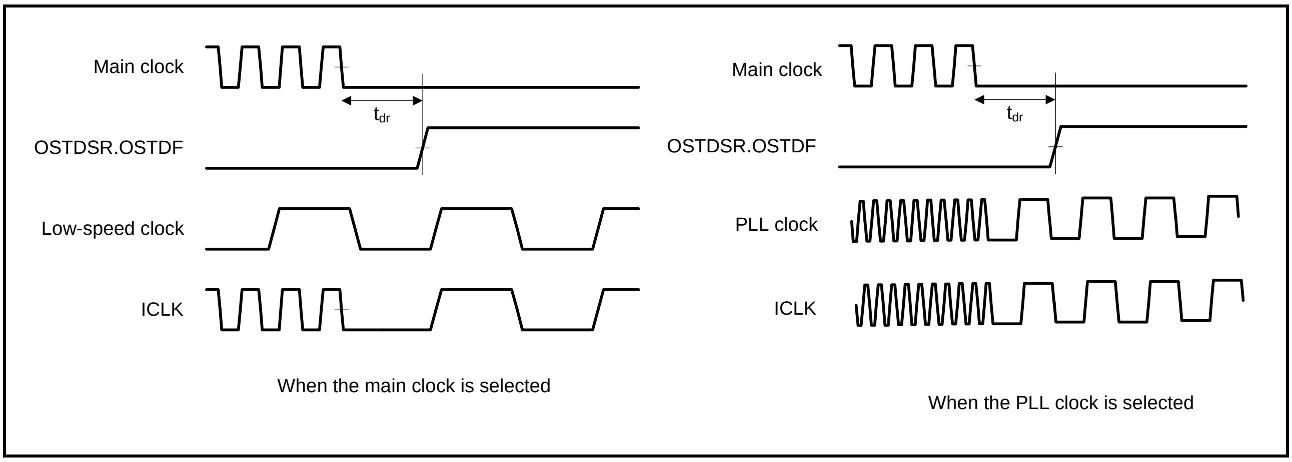


Figure 35.55 Oscillation Stop Detection Timing

35.9 ROM (Flash Memory for Code Storage) Characteristics

Table 35.37 ROM (Flash Memory for Code Storage) Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	1000	—	—	Times	
Data hold time	After 1000 times of N_{PEC}	t_{DRP}	20*2, *3	—	—	Year	T_a = +85°C

Note 1. Definition of reprogram/erase cycle: The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 4-byte programming is performed 256 times for different addresses in 1-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. Characteristic when using the flash memory programmer and the self-programming library provided from Renesas Electronics.

Note 3. This result is obtained from reliability testing.

Table 35.38 ROM (Flash Memory for Code Storage) Characteristics (2): High-Speed Operating Mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	112.0	967.0	—	52.3	490.5	μs
Erasure time	2-Kbyte	t_{E2K}	—	8.7	278.1	—	5.5	214.6	ms
	128-Kbyte (when block erase command used)		—	239.7	5111.4	—	25.9	734.3	ms
	128-Kbyte (when all-block erase command used)	t_{E128K}	—	234.5	4906.8	—	20.6	524.6	ms
Blank check time	8-byte	t_{BC8}	—	—	55.0	—	—	16.1	μs
	2-Kbyte	t_{BC2K}	—	—	1840.0	—	—	135.7	μs
Erase operation forcible stop time		t_{SED}	—	—	18.0	—	—	10.7	μs
Start-up area switching setting time		t_{SAS}	—	12.3	566.5	—	6.2	433.5	ms
Access window time		t_{AWS}	—	12.3	566.5	—	6.2	433.5	ms
ROM mode transition wait time 1		t_{DIS}	2.0	—	—	2.0	—	—	μs
ROM mode transition wait time 2		t_{MS}	5.0	—	—	5.0	—	—	μs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be ±3.5%.

Table 35.39 ROM (Flash Memory for Code Storage) Characteristics (3): Middle-Speed Operating Mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VREFH0 = VCC to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V, T_a = –40 to +105°C

Temperature range for the programming/erasure operation: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8-byte	t_{P8}	—	152.0	1367.0	—	97.9	936.0	μs
Erasure time	2-Kbyte	t_{E2K}	—	8.8	279.7	—	5.9	220.8	ms
	128-Kbyte (when block erase command used)		—	239.8	5114.7	—	55.5	1336.4	ms
	128-Kbyte (when all-block erase command used)	t_{E128K}	—	234.6	4908.5	—	50.3	1130.1	ms
Blank check time	8-byte	t_{BC8}	—	—	85.0	—	—	50.9	μs
	2-Kbyte	t_{BC2K}	—	—	1870.0	—	—	401.5	μs
Erase operation forcible stop time		t_{SED}	—	—	28.0	—	—	21.3	μs
Start-up area switching setting time		t_{SAS}	—	13.0	573.3	—	7.7	450.1	ms
Access window time		t_{AWS}	—	13.0	573.3	—	7.7	450.1	ms
ROM mode transition wait time 1		t_{DIS}	2.0	—	—	2.0	—	—	μs
ROM mode transition wait time 2		t_{MS}	3.0	—	—	3.0	—	—	μs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be ±3.5%.

35.10 Usage Notes

35.10.1 Connecting VCL Capacitor and Bypass Capacitors

This MCU integrates an internal voltage-down circuit, which is used for lowering the power supply voltage in the internal MCU to adjust automatically to the optimum level. A 4.7-µF capacitor needs to be connected between this internal voltage-down power supply (VCL pin) and VSS pin. Figure 35.56 to Figure 35.58 shows how to connect external capacitors. Place an external capacitor close to the pins. Do not apply the power supply voltage to the VCL pin.
Insert a multilayer ceramic capacitor as a bypass capacitor between each pair of the power supply pins. Implement a bypass capacitor to the MCU power supply pins as close as possible. Use a recommended value of 0.1 µF as the capacitance of the capacitors. For the capacitors related to crystal oscillation, see section 9, Clock Generation Circuit. For the capacitors related to analog modules, also see section 29, 12-Bit A/D Converter (S12ADE).
For notes on designing the printed circuit board, see the descriptions of the application note "Hardware Design Guide" (R01AN1411EJ). The latest version can be downloaded from Renesas Electronics Website.

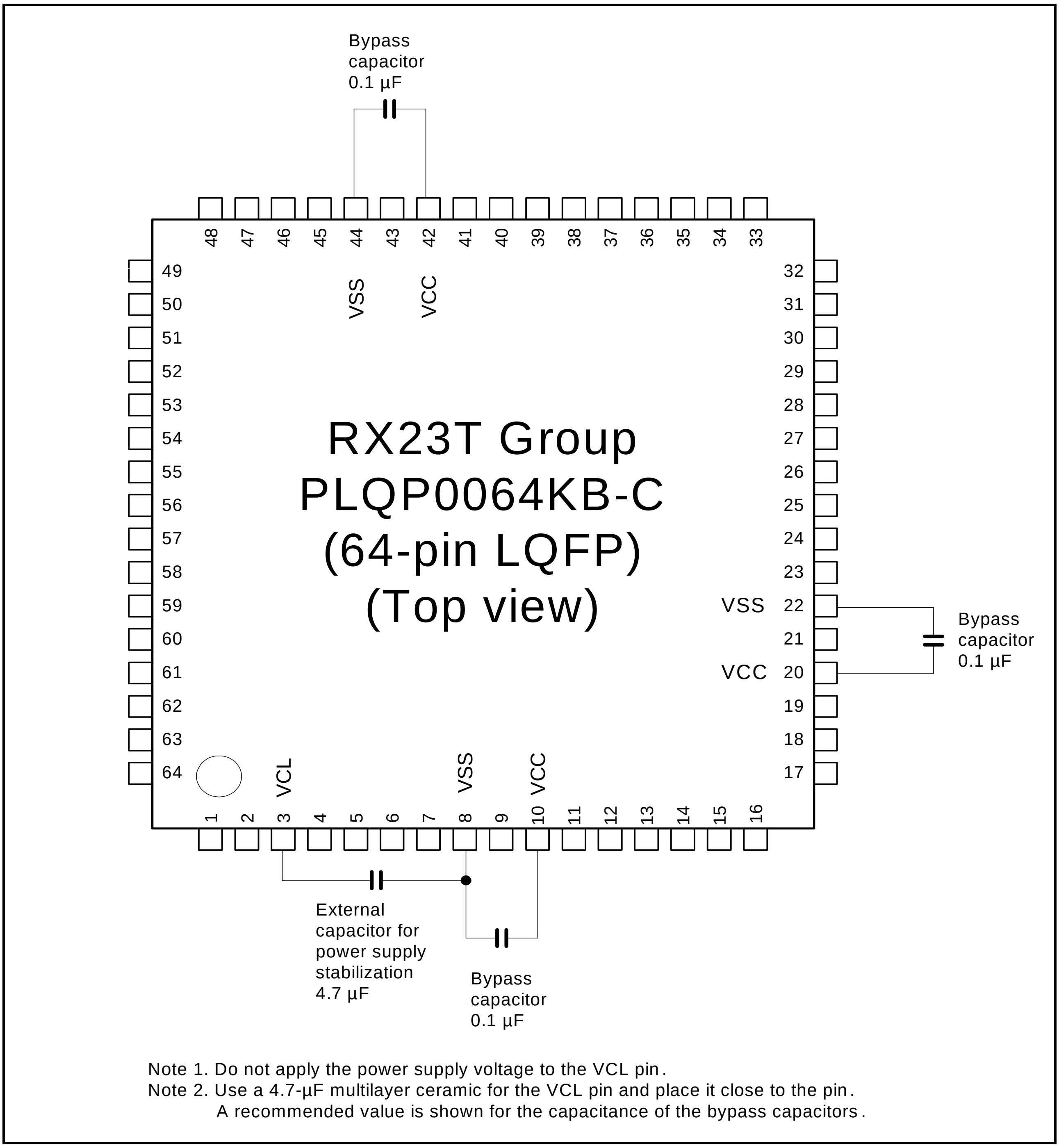
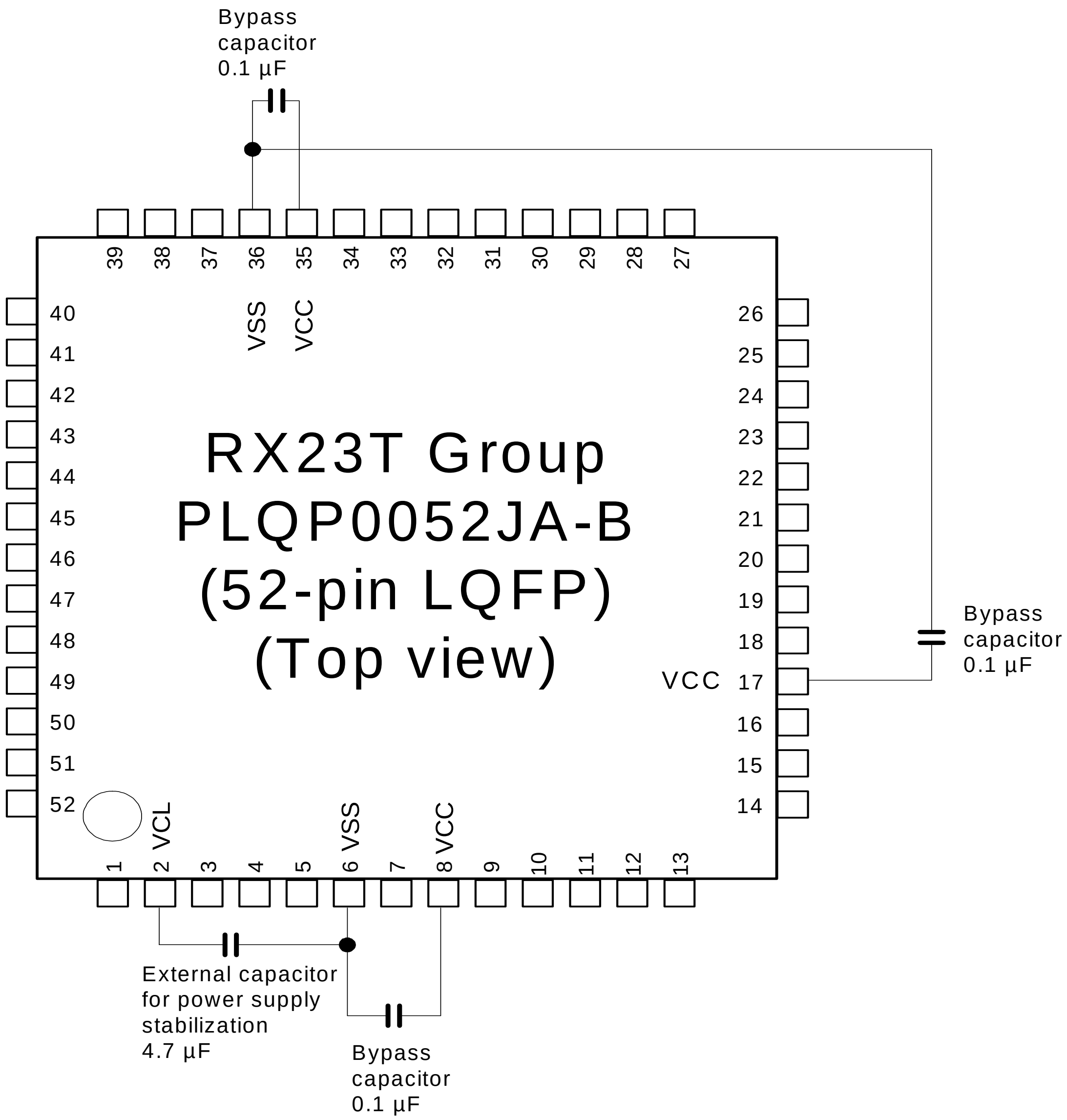
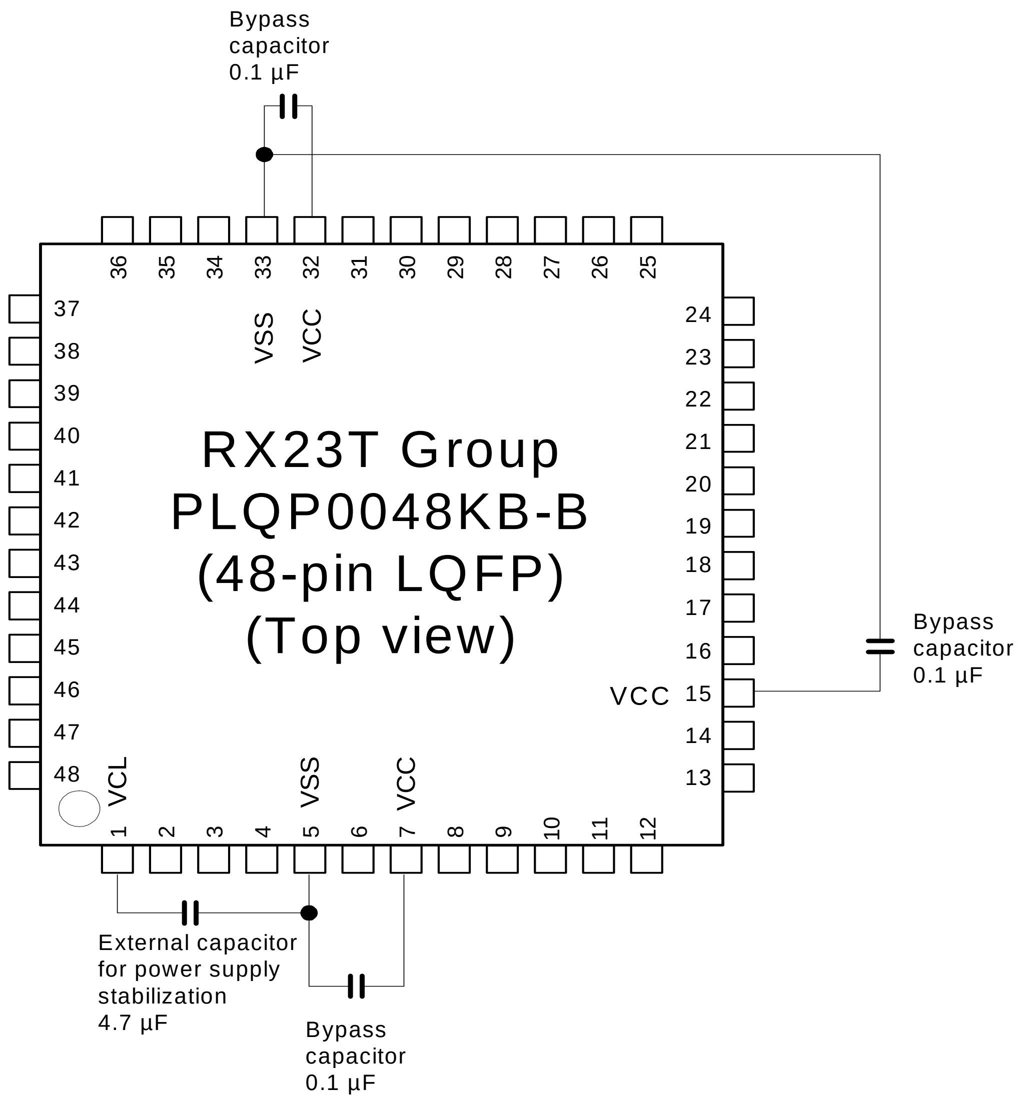


Figure 35.56 Connecting Capacitors (64 Pins)



Note 1. Do not apply the power supply voltage to the VCL pin.
Note 2. Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 35.57 Connecting Capacitors (52 Pins)



Note 1. Do not apply the power supply voltage to the VCL pin.
Note 2. Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 35.58 Connecting Capacitors (48 Pins)

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing Mode

Port Name (Pin Name)	Operating Mode According to Registers Setting	Reset	Software Standby Mode
P00, P01 (IRQ2, IRQ4)	All	Hi-Z	Keep-O*1
P02 (IRQ5)	All	Hi-Z	Keep-O*1
P10, P11 (IRQ0, IRQ1)	All	Hi-Z	Keep-O*1
P22, P23, P24 (IRQ2, IRQ4, IRQ3)	All	Hi-Z	Keep-O*1
P30 to P33, P36, P37	All	Hi-Z	Keep-O
P40 to P47	All	Hi-Z	Keep-O
P70 (IRQ5)	All	Hi-Z	Keep-O*1
P71 to P76	All	Hi-Z	Keep-O
P91, P92	All	Hi-Z	Keep-O
P93, P94 (IRQ0,IRQ1)	All	Hi-Z	Keep-O*1
PA2 (IRQ4)	All	Hi-Z	Keep-O*1
PA3 to PA5	All	Hi-Z	Keep-O
PB0, PB2, PB3, PB5, PB7	All	Hi-Z	Keep-O
PB1, PB4, PB6 (IRQ2, IRQ3, IRQ5)	All	Hi-Z	Keep-O*1
PD3, PD7	All	Hi-Z	Keep-O
PD4, PD5, PD6 (IRQ2, IRQ3, IRQ5)	All	Hi-Z	Keep-O*1
PE2 (NMI)	All	Hi-Z	Keep-O*1

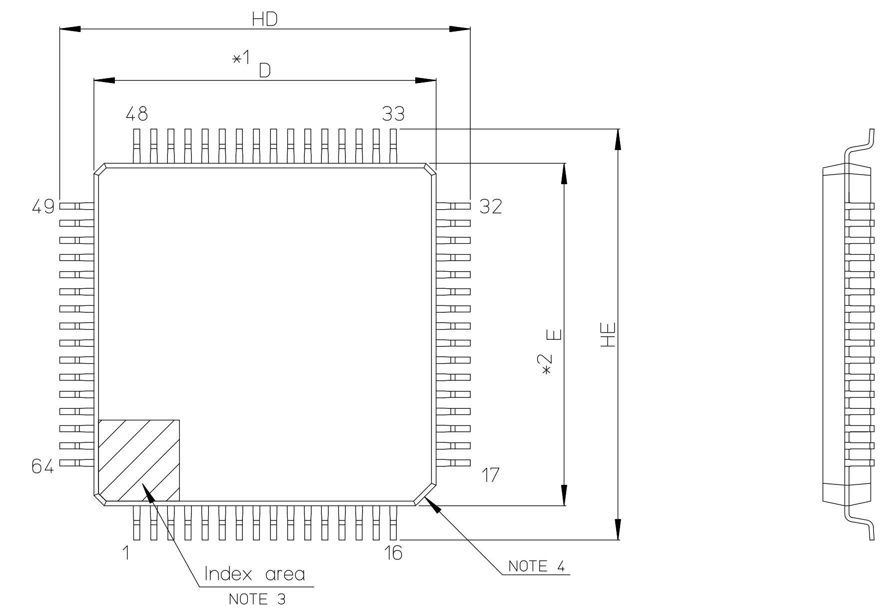
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Hi-Z: High-impedance

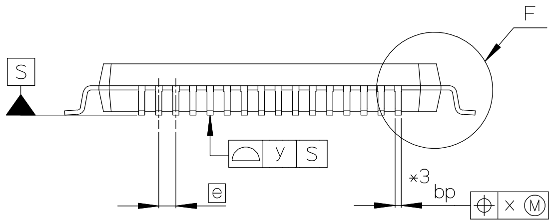
Note 1. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.

Appendix 2. Package Dimensions

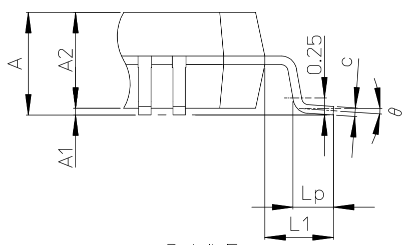
Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3g





NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

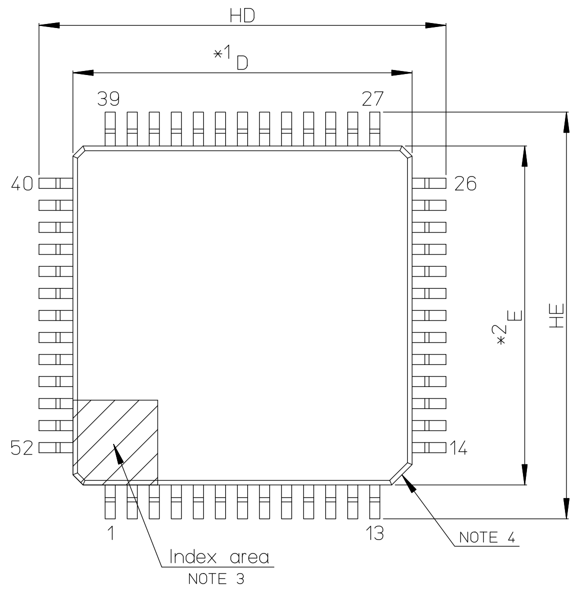


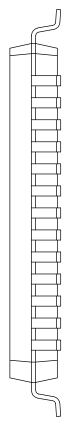
Detail F

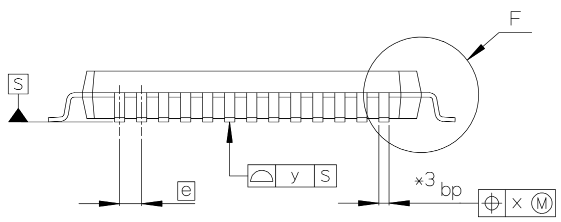
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A2	—	1.4	—
HD	11.8	12.0	12.2
HE	11.8	12.0	12.2
A	—	—	1.7
A1	0.05	—	0.15
bp	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Lp	0.45	0.6	0.75
L1	—	1.0	—

Figure A 64-Pin LFQFP (PLQP0064KB-C)

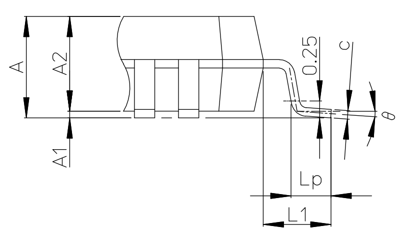
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP52-10x10-0.65	PLQP0052JA-B	—	0.3g







NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

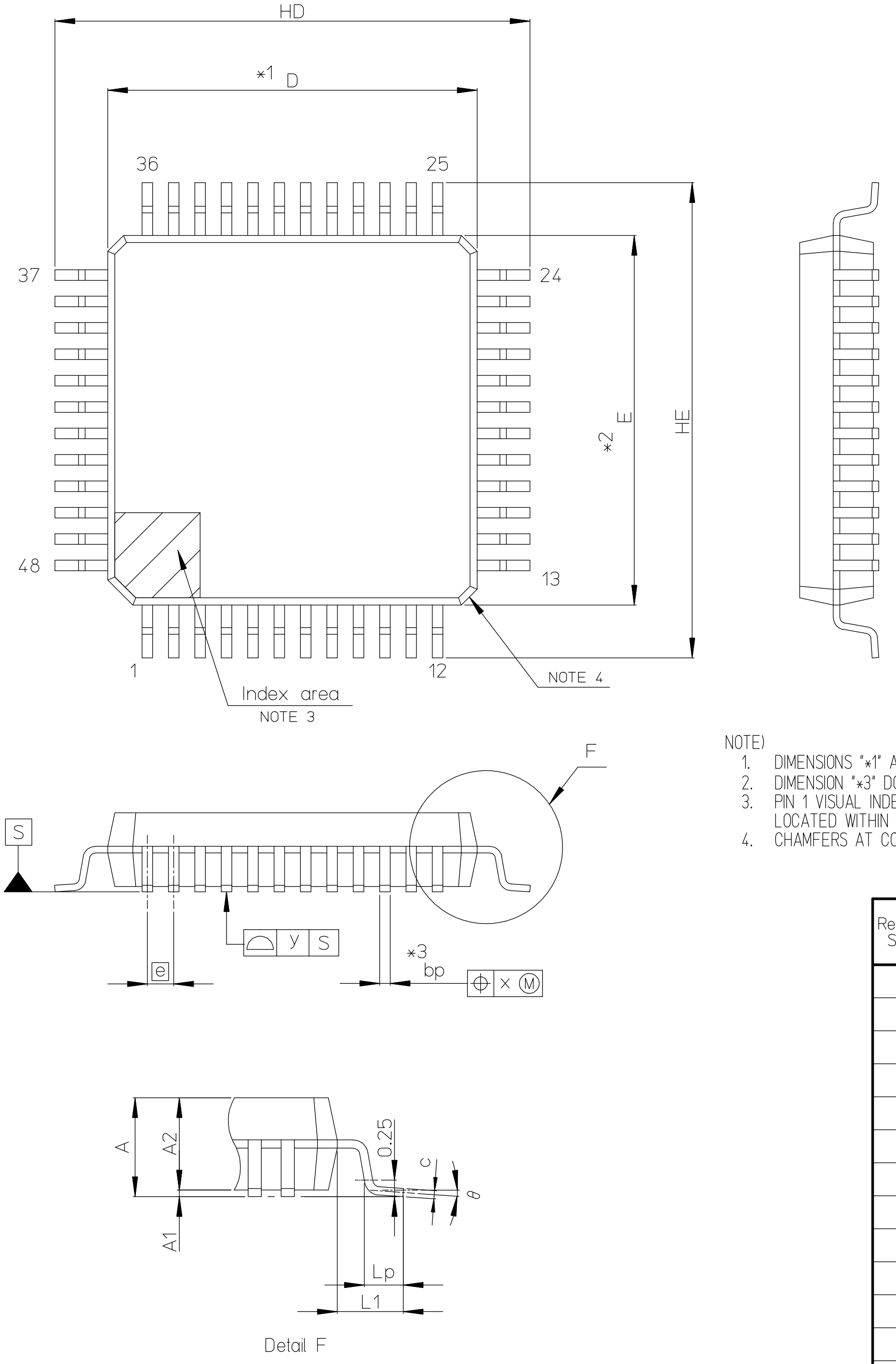


Detail F

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A2	—	1.4	—
HD	11.8	12.0	12.2
HE	11.8	12.0	12.2
A	—	—	1.7
A1	0.05	—	0.15
bp	0.27	0.32	0.37
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Lp	0.45	0.6	0.75
L1	—	1.0	—

Figure B 52-Pin LQFP (PLQP0052JA-B)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	——	0.2g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	——	1.4	——
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.17	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

Figure C 48 -Pin LFQFP (PLQP0048KB-B)

REVISION HISTORY	RX23T Group User's Manual: Hardware

Classifications
- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.00	Mar 31, 2015	—	First edition, issued	
1.10	Jan 13, 2016	Features		
		34	Features changed	
		1. Overview		
		35, 36	Table 1.1 Outline of Specifications (1/3) (2/3) changed	
		39	Table 1.3 List of Products: D Version (Ta = –40 to +85°C) changed	
		39	Table 1.4 List of Products: G Version (Ta = –40 to +105°C) changed	
		40	Figure 1.1 How to Read the Product Part Number changed	
		43	Table 1.5 Pin Functions (2/2) changed	
		44	Figure 1.3 Pin Assignments of the 64-Pin LFQFP changed	
		45	Figure 1.4 Pin Assignments of the 52-Pin LQFP changed	
		46	Figure 1.5 Pin Assignments of the 48-Pin LFQFP changed	
		47	Table 1.6 List of Pins and Pin Functions (64-Pin LFQFP) (1/2) changed	
		49	Table 1.7 List of Pins and Pin Functions (52-Pin LQFP) changed	
		50	Table 1.8 List of Pins and Pin Functions (48-Pin LFQFP) changed	
		4. Address Space		
		84	Figure 4.1 Memory Map in Each Operating Mode changed	
		5. I/O Registers		
		87	Table 5.1 List of I/O Registers (Address Order) (1 / 16) Address: 0008 0036h High-Speed On-Chip Oscillator Control Register (HOCOCR), Address: 0008 00A5h High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) added	
		97	Table 5.1 List of I/O Registers (Address Order) (11 / 16) Address: 0008 C087h Open Drain Control Register 1 (ODR1) added	
		7. Option-Setting Memory		
		118	7.2.2 Option Function Select Register 1 (OFS1) HOCOEN (HOCO Oscillation Enable) bit description added	
		9. Clock Generation Circuit		
		137	Table 9.1 Specifications of Clock Generation Circuit changed	
		138	Figure 9.1 Block Diagram of Clock Generation Circuit changed	
		142	9.2.2 System Clock Control Register 3 (SCKCR3) CKSEL[2:0] (Clock Source Select) bit description changed	
		148	9.2.8 High-Speed On-Chip Oscillator Control Register (HOCOCR) added	
		149	9.2.9 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) added	
		150, 151	9.2.10 Oscillation Stabilization Flag Register (OSCOVFSR) HCOVF (HOCO Clock Oscillation Stabilization Flag) added	
		154	9.2.13 Main Clock Oscillator Wait Control Register (MOSCWTCR) MSTS[4:0] (Main Clock Oscillator Wait Time) bit description changed	
		161	9.4.2 Oscillation Stop Detection Interrupts changed	
		162	9.6 Internal Clock changed	
		162	9.6.4 CAC Clock changed	
		165	9.7.4 Notes on Resonator Connection Pins added	
		10. Clock Frequency Accuracy Measurement Circuit (CAC)		
		166	Table 10.1 CAC Specifications changed	
		166	Figure 10.1 CAC Block Diagram changed	
		168	10.2.2 CAC Control Register 1 (CACR1) changed	
		169	10.2.3 CAC Control Register 2 (CACR2) changed	
		11. Low Power Consumption		
		177	Table 11.2 Operating Conditions of Each Power Consumption Mode changed, High-speed on-chip oscillator added	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Jan 13, 2016	183	11.2.4 Module Stop Control Register C (MSTPCRC) changed	
		191	11.6.2.1 Entry to Deep Sleep Mode, Note 1 added	
		12. Register Write Protection Function		
		197	Table 12.1 Association between PRCR Bits and Registers to be Protected changed	
		198	12.1.1 Protect Register (PRCR) changed	
		14. Interrupt Controller (ICUb)		
		239	14.4.8 External Pin Interrupts changed	
		240	14.5 Non-maskable Interrupt Operation changed	
		15. Buses		
		244	Table 15.2 Addresses Assigned for Each Bus changed	
		17. Data Transfer Controller (DTCa)		
		285	17.2.8 DTC Vector Base Register (DTCVBR) changed	
		286	17.2.10 DTC Module Start Register (DTCST) changed	
		306	17.8 Low Power Consumption Function changed	
		18. I/O Ports		
		308, 309	18.1 Overview, Table 18.1 Specifications of I/O Ports changed	
		309	Table 18.2 Port Functions changed	
		310, 311	Figure 18.1 I/O Port Configuration (1), Figure 18.2 I/O Port Configuration (2) changed	
		312	Figure 18.3 I/O Port Configuration (3) added	
		313	18.3.1 Port Direction Register (PDR) changed	
		318	18.3.6 Open Drain Control Register 1 (ODR1) changed	
		323	Table 18.6 Unused Pin Configuration changed	
		20. Multi-Function Timer Pulse Unit (MTU3c)		
		342	Table 20.1 MTU Specifications changed	
		345	Figure 20.1 Block Diagram of MTU (MTU0 to MTU5) changed	
		346	Table 20.3 Pin Configuration of the MTU changed	
		348	Table 20.4 CCLR[2:0] (MTU0, MTU3, MTU4), Table 20.5 CCLR[2:0] (MTU1 and MTU2) moved from 20.2.2 Timer Control Register 2 (TCR2)	
		419	20.3.4 Cascaded Operation changed	
		429 to 431	Phase Counting Mode, 20.3.6.1 16-Bit Phase Counting Mode changed	
		440	20.3.6.2 Cascade Connection 32-Bit Phase Counting Mode added	
		465	(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode changed	
		479	(4) Complementary PWM Mode Output Protection Functions changed	
		486	20.3.10 Synchronous Operation of MTU0 to MTU4 changed	
		491	20.3.15 A/D Conversion Start Request Frame Synchronization Signal changed	
		492	Table 20.63 MTU Interrupt Sources priority column added	
		493	(1) Input Capture/Compare Match Interrupt, (2) Overflow Interrupt changed	
		513	Figure 20.134 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode changed	
		518	20.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode changed	
		520	20.6.27 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode changed	
		21. Port Output Enable 3 (POE3b)		
		555	21.2.4 Input Level Control/Status Register 6 (ICSR6) OSTSTF (OSTST High-Impedance Flag) flag description changed	
		22. 8-Bit Timer (TMR)		
		All	Terms changed: Frequency dividing clock → internal clock, external reset → external counter reset, external clock → external count clock, counter clock → count clock counter external reset → external counter reset TCNT input clock → TCNT count clock	
		586	22.2.4 Timer Control Register (TCR) Note changed	

Rev.	Date	Description: Page	Description: Summary	Classification
1.10	Jan 13, 2016	587	22.2.5 Timer Counter Control Register (TCCR) Note changed	
		588	Table 22.5 Clock Input to TCNT and Count Condition Note changed	
		23. Compare Match Timer (CMT)		
		609	Table 23.2 CMT Interrupt Sources, CMI2 and CMI3 added	
		24. Independent Watchdog Timer (IWDTa)		
		611	24.1 Overview changed	
		616	Figure 24.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period changed	
		620	24.3.1.1 Register Start Mode changed	
		625	24.3.3 Refresh Operation [Sample refreshing timings] changed	
		628	Figure 24.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b) changed	
		25. Serial Communications Interface (SCIg)		
		642 to 644	(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0) RE Bit (Receive Enable), RIE Bit (Receive Interrupt Enable) bit descriptions changed	
		645, 646	(2) Smart Card Interface Mode (SCMR.SMIF = 1) RIE Bit (Receive Interrupt Enable) bit description changed	
		647 to 649	(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0) b6 and b7 changed, RDRF Flag (Receive Data Full Flag), TDRE Flag (Transmit Data Empty Flag) flag descriptions added	TN-RX*-A138A/E
		650, 651	(2) Smart Card Interface Mode (SCMR.SMIF = 1) changed, RDRF Flag (Receive Data Full Flag), TDRE Flag (Transmit Data Empty Flag) flag descriptions added	TN-RX*-A138A/E
		652	25.2.10 Smart Card Mode Register (SCMR) changed	
		657	Table 25.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) changed	
		663, 664	25.2.13 Serial Extended Mode Register (SEMR) changed, ACS0 (Asynchronous Mode Clock Source Select) bit description changed	
		679	25.3.6 SCI Initialization (Asynchronous Mode) changed	
		688	25.4 Multi-Processor Communications Function changed	
		730	Table 25.27 Interrupt Sources changed	
		26. I^{2}C-bus Interface (RIICa)		
		All	Terms changed: transferred frames → transferred bytes, address frame → address byte standard → specification [Sm], [Fm], [W], [R] → (Sm), (Fm), (write), (read), chip → MCU	
		762, 763	26.2.9 I^{2}C-bus Status Register 1 (ICSR1) HOA (Host Address Detection Flag) flag description changed	
		27. Serial Peripheral Interface (RSPIa)		
		All	Terms changed: RSPCK → RSPCKA, MOSI → MOSIA, SSL0 → SSLA0	
		824 to 826	27.2.4 RSPI Status Register (SPSR) b5 and b7 changed SPTEF (Transmit Buffer Empty Flag), SPRF (Receive Buffer Full Flag) flag descriptions added	
		827 to 829	27.2.5 RSPI Data Register (SPDR) changed	
		833, 834	27.2.9 RSPI Data Control Register (SPDCR) SPFC[1:0] (Number of Frames Specification), SPRDTD (RSPI Receive/Transmit Data Select) bit descriptions changed	
		840, 841	27.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7) SPB[3:0] (RSPI Data Length Setting) bit description changed	
		843	Table 27.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode changed	
		862	27.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0) changed Figure 27.24 Operation Example of SPCR.TXMD = 0 changed	
		863	27.3.6.2 Transmit Operations Only (SPCR.TXMD = 1) changed Figure 27.25 Operation Example of SPCR.TXMD = 1 changed	
		864, 865	27.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts changed Figure 27.26 Operation Example of SPTI and SPRI Interrupts changed	

Rev.	Date	Description: Page	Description: Summary	Classification
1.10	Jan 13, 2016	866	27.3.8 Error Detection changed	
		867, 868	27.3.8.1 Overrun Error changed Figure 27.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1), Figure 27.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0) changed	
		871	27.3.9.1 Initialization by Clearing the SPE Bit changed	
		872	(1) Starting a Serial Transfer, (2) Terminating a Serial Transfer changed	
		879	Figure 27.36 Flowchart in Master Mode (Transmission) changed	
		880	Figure 27.37 Flowchart in Master Mode (Reception) changed	
		881	Figure 27.38 Flowchart for Master Mode (Error Processing) changed	
		882	(2) Terminating a Serial Transfer changed	
		884	Figure 27.40 Flowchart in Slave Mode (Transmission), Figure 27.41 Flowchart in Slave Mode (Reception) changed	
		885	(c) Flow of Error Processing, Figure 27.42 Flowchart for Slave Mode (Error Processing) changed	
		886	(1) Starting a Serial Transfer, (2) Terminating a Serial Transfer changed	
		890	(2) Terminating a Serial Transfer changed	
		894	Table 27.13 Interrupt Sources of RSPI changed	
		895	27.4.4 Notes on the SPRF and SPTEF flags added	
		29. 12-Bit A/D Converter (S12ADE)		
		914	29.2.4 A/D Channel Select Register A0 (ADANSA0) changed	
		915	29.2.5 A/D Channel Select Register A1 (ADANSA1) changed	
		916	29.2.6 A/D Channel Select Register B0 (ADANSB0) changed	
		917	29.2.7 A/D Channel Select Register B1 (ADANSB1) changed	
		918	29.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0) changed	
		919	29.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1) changed	
		921, 922	29.2.11 A/D Control Extended Register (ADCER) DIAGLD (Self-Diagnosis Mode Select), DIAGM (Self-Diagnosis Enable) bit descriptions changed	
		962, 963	29.7.9 Voltage Range of Analog Power Supply Pins • Analog input voltage range changed, Figure 29.28 Relationship Between Voltage Applied to Analog Input Pins and Output Code added	
		30. D/A Converter for Generating Comparator C Reference Voltage (DA)		
		968	30.4.4 Setting the D/A Converter added	
		31. Comparator C (CMPC)		
		All	Terms changed: Comparator detection result → comparison result	
		972	31.2.1 Comparator Control Register (CMPCTL) CEG[1:0] (Comparator Edge Select) bit description changed	
		974	31.2.3 Comparator Reference Voltage Select Register (CMPSEL1) changed	
		976	31.3.1 Comparator Operation Example changed	
		979	Figure 31.6 Comparator Operation Termination Flowchart changed	
		980	31.4.3 Comparator C Operation in Software Standby Mode changed 31.4.4 Setting the D/A Converter for Generating Comparator C Reference Voltage changed	
		33. RAM		
		987	Table 33.1 Specifications of RAM changed	
		34. Flash Memory		
		1029	Table 34.14 Error Information changed	
		1031	34.9.5.4 Block Information Inquiry changed	
		1044	34.10 Serial Programmer Operation in Boot Mode (SCI) changed	
		35. Electrical Characteristics		
		1058	Table 35.3 DC Characteristics (1) changed	
		1073	Table 35.14 Operating Frequency Value (High-Speed Operating Mode), Table 35.15 Operating Frequency Value (Middle-Speed Operating Mode) changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Jan 13, 2016	1074	Table 35.16 Clock Timing, Figure 35.20 EXTAL External Clock Input Timing changed	
		1075, 1076	Figure 35.24 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0), Figure 35.25 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit) added	
		1095	Table 35.32 Comparator Characteristics changed	
		1096	Table 35.33 D/A Conversion Characteristics symbol added	
		1098	Table 35.35 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2) changed	
		1099	Figure 35.52 Voltage Detection Circuit Timing (Vdet0) changed	
		1105	Figure 35.56 Connecting Capacitors (64 Pins) changed	
		1106	Figure 35.57 Connecting Capacitors (52 Pins) changed	
		1107	Figure 35.58 Connecting Capacitors (48 Pins) changed	
		Appendix 1. Port States in Each Processing Mode		
		1108	Table 1.1 Port States in Each Processing Mode changed	
		Appendix 2. Package Dimensions		
		1109	Figure A 64-Pin LFQFP (PLQP0064KB-C) changed	
		1110	Figure B 52-Pin LQFP (PLQP0052JA-B) changed	
		1111	Figure C 48 -Pin LFQFP (PLQP0048KB-B) changed	

RX23T Group User’s Manual: Hardware

Publication Date:	Rev.1.00	Mar 31, 2015
	Rev.1.10	Jan 13, 2016
Published by:	Renesas Electronics Corporation	

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

RX23T Group

Renesas Electronics Corporation

R01UH0520EJ0110