User's Manual

16

RL78/G1G

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev. 1.40 Apr 2024

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

"Standard":	Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
"High Quality":	Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

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

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/G1G and design and develop application systems and programs for these devices.
The target products are as follows.

- 30-pin: R5F11EAx (x = 8, A)
- 32-pin: R5F11EBx (x = 8, A)
- 44-pin: R5F11EFx (x = 8, A)

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

The RL78/G1G manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G1G User's Manual Hardware (This Manual)	**RL78 Family User's Manual Software**
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupts	• Explanation of each instruction
• Other on-chip peripheral functions	
• Electrical specifications	

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark “<R>” shows major revised points. The revised points can be easily searched by copying an “<R>” in the PDF file and specifying it in the “Find what:” field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the RL78/G1G Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual Software (R01US0015E)**.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representations: $\overline{\times\times\times}$ (overscore over pin and signal name)

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numerical representations: Binary.................×××× or ××××B

Decimal...............××××

Hexadecimal.......××××H

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Devices

Document Name	Document No.
RL78/G1G User’s Manual Hardware	This manual
RL78 Family User’s Manual Software	R01US0015E

Documents Related to Flash Memory Programming (User’s Manual)

Document Name		Document No.
PG-FP5 Flash Memory Programmer User’s Manual		—
	RL78, 78K, V850, RX100, RX200, RX600 (Except RX64x), R8C, SH	R20UT2923E
	Common	R20UT2922E
	Setup Manual	R20UT0930E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
Renesas Microcontrollers RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	R50ZZ0003E
Semiconductor Reliability Handbook	R51ZZ0001E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

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

CONTENTS

6. TIMER ARRAY UNIT 167

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 2.7 to 5.5 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high-speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator) to low-speed (1.0 μs: @1 MHz operation with high-speed on-chip oscillator)
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 1.5 KB

Code flash memory

- Code flash memory: 8 to 16 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (flash shield window function)

High-speed on-chip oscillator

- Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 4 MHz, and 1 MHz
- High accuracy: ±2.0%

Operating ambient temperature

- T_A = -40 to +85°C

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 6 levels)

Event link controller (ELC)

- Event signals of 18 to 19 types can be linked to the specified peripheral function.

Serial interfaces

- Simplified SPI (CSI[Note 1]): 1 channel
- UART: 2 channels
- Simplified I^{2}C: 1 channel

Timer

- 16-bit timer: 7 channels
  (Timer Array Unit (TAU): 4 channels, Timer RJ: 1 channel, Timer RD: 2 channels)
- 12-bit interval timer: 1 channel
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 2.7 to 5.5 V)
- Analog input: 8 to 12 channels
- Internal reference voltage (1.45 V) and temperature sensor[Note 2]

Comparator

- 2 channels
- The voltage from a dedicated 8-bit DAC (resolution of 256 with V_{DD}/AV_{REFP} or V_{SS}/AV_{REFM} as the internally generated reference voltage) can be selected as the reference voltage.

Programmable gain amplifier

I/O port

- I/O port: 26 to 40
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 2.5/3 V device
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

Others

- On-chip BCD (binary-coded decimal) correction circuit

Note 1. Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

Note 2. Selectable only in HS (high-speed main) mode.

Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

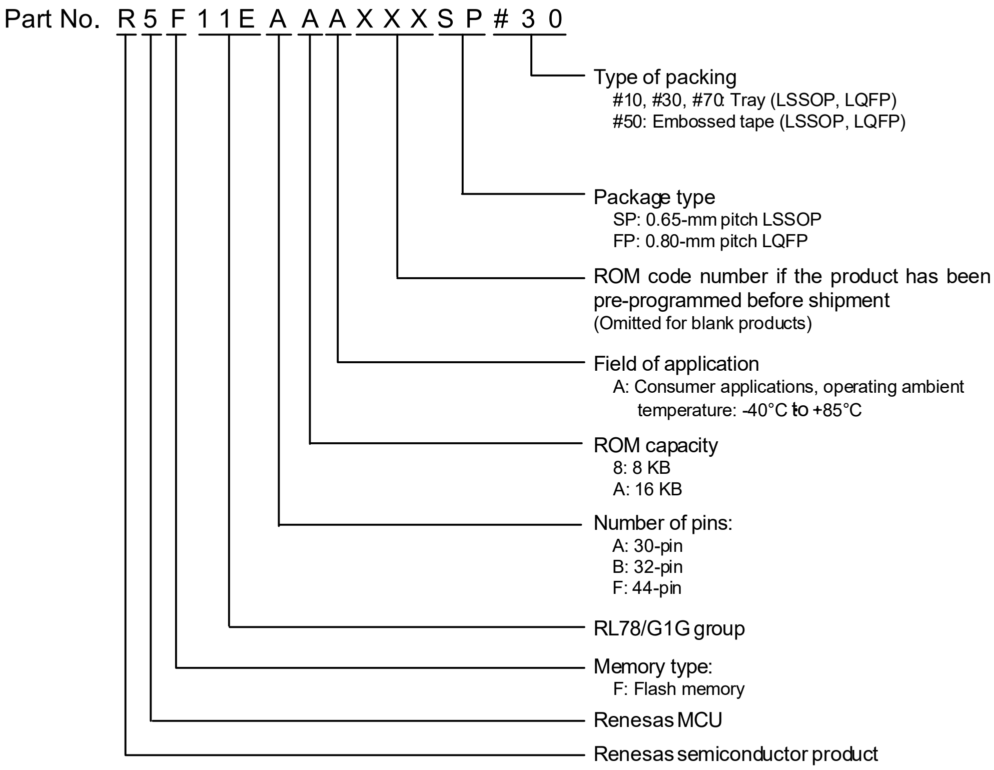
○ ROM, RAM capacities

Flash ROM	RAM	30 pins	32 pins	44 pins
16 KB	1.5 KB Note	R5F11EAAASP	R5F11EBAAFP	R5F11EFAAFP
8 KB		R5F11EA8ASP	R5F11EB8AFP	R5F11EF8AFP

Note This is 630 bytes when the self-programming function is used. (For details, see **CHAPTER 3 CPU ARCHITECTURE**).

1.2 List of Part Numbers

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G1G



<R>

Type of packing
#10, #30, #70: Tray (LSSOP, LQFP)
#50: Embossed tape (LSSOP, LQFP)

Package type
SP: 0.65-mm pitch LSSOP
FP: 0.80-mm pitch LQFP

ROM code number if the product has been pre-programmed before shipment
(Omitted for blank products)

Field of application
A: Consumer applications, operating ambient temperature: -40°C to +85°C

ROM capacity
8: 8 KB
A: 16 KB

Number of pins:
A: 30-pin
B: 32-pin
F: 44-pin

RL78/G1G group

Memory type:
F: Flash memory

Renesas MCU

Renesas semiconductor product

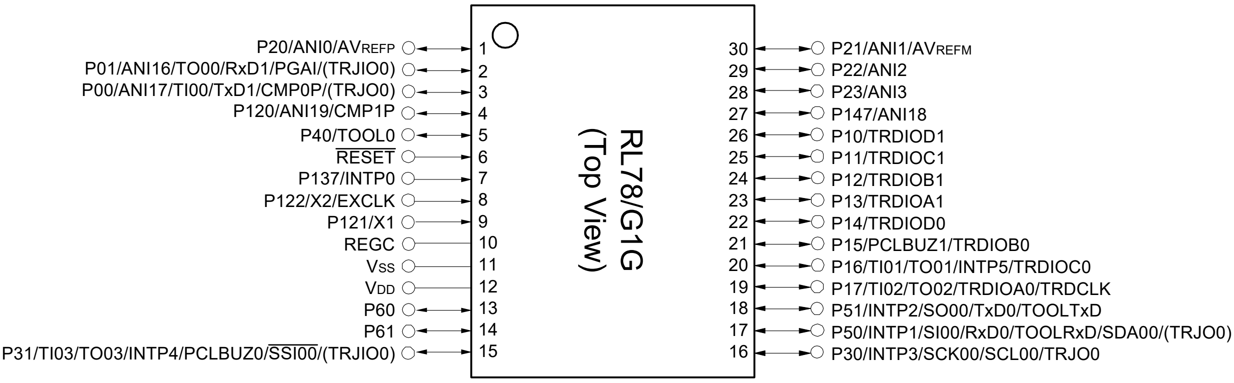
Table 1 - 1 Orderable Part Numbers

Pin Count	Package	Part Number
44 pins	44-pin plastic LQFP (10 × 10 mm)	R5F11EFAAFP#10, R5F11EFAAFP#30, R5F11EFAAFP#50, R5F11EFAAFP#70
		R5F11EF8AFP#10, R5F11EF8AFP#30, R5F11EF8AFP#50, R5F11EF8AFP#70
32 pins	32-pin plastic LQFP (7 × 7 mm)	R5F11EBAAFP#10, R5F11EBAAFP#30, R5F11EBAAFP#50, R5F11EBAAFP#70
		R5F11EB8AFP#10, R5F11EB8AFP#30, R5F11EB8AFP#50, R5F11EB8AFP#70
30 pins	30-pin plastic LSSOP (7.62 mm (300))	R5F11EAAASP#10, R5F11EAAASP#30, R5F11EAAASP#50, R5F11EAAASP#70
		R5F11EA8ASP#10, R5F11EA8ASP#30, R5F11EA8ASP#50, R5F11EA8ASP#70

1.3 Pin Configuration (Top View)

1.3.1 30-pin products

- 30-pin plastic LSSOP (7.62 mm (300), 0.65 mm pitch)



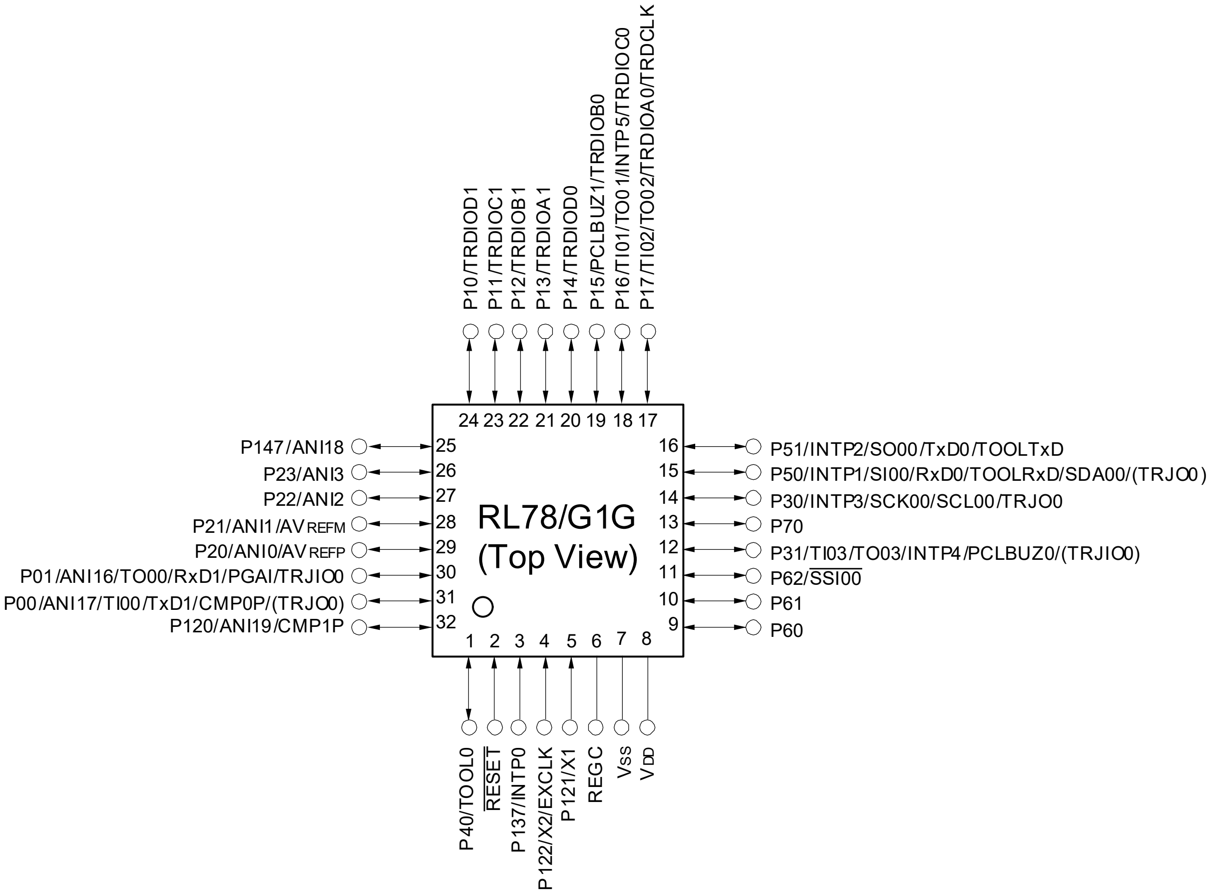
Caution Connect the REGC pin to Vss pin via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

1.3.2 32-pin products

- 32-pin plastic LQFP (7 x 7 mm, 0.8 mm pitch)



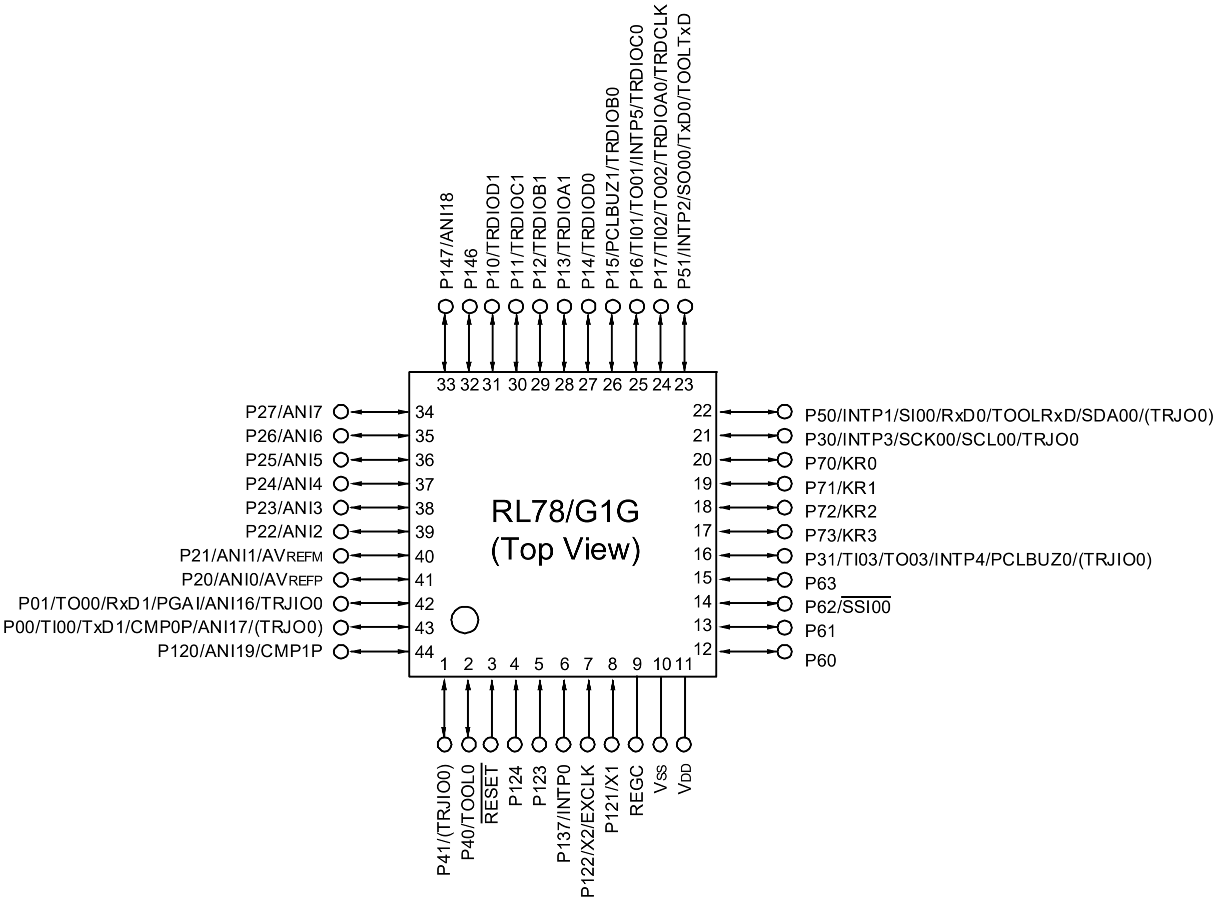
Caution **Connect the REGC pin to Vss pin via a capacitor (0.47 to 1 μF).**

Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

1.3.3 44-pin products

- 44-pin plastic LQFP (10 x 10 mm, 0.8 mm pitch)



Caution **Connect the REGC pin to Vss pin via a capacitor (0.47 to 1 μF).**

Remark 1. For pin identification, see **1.4 Pin Identification**.

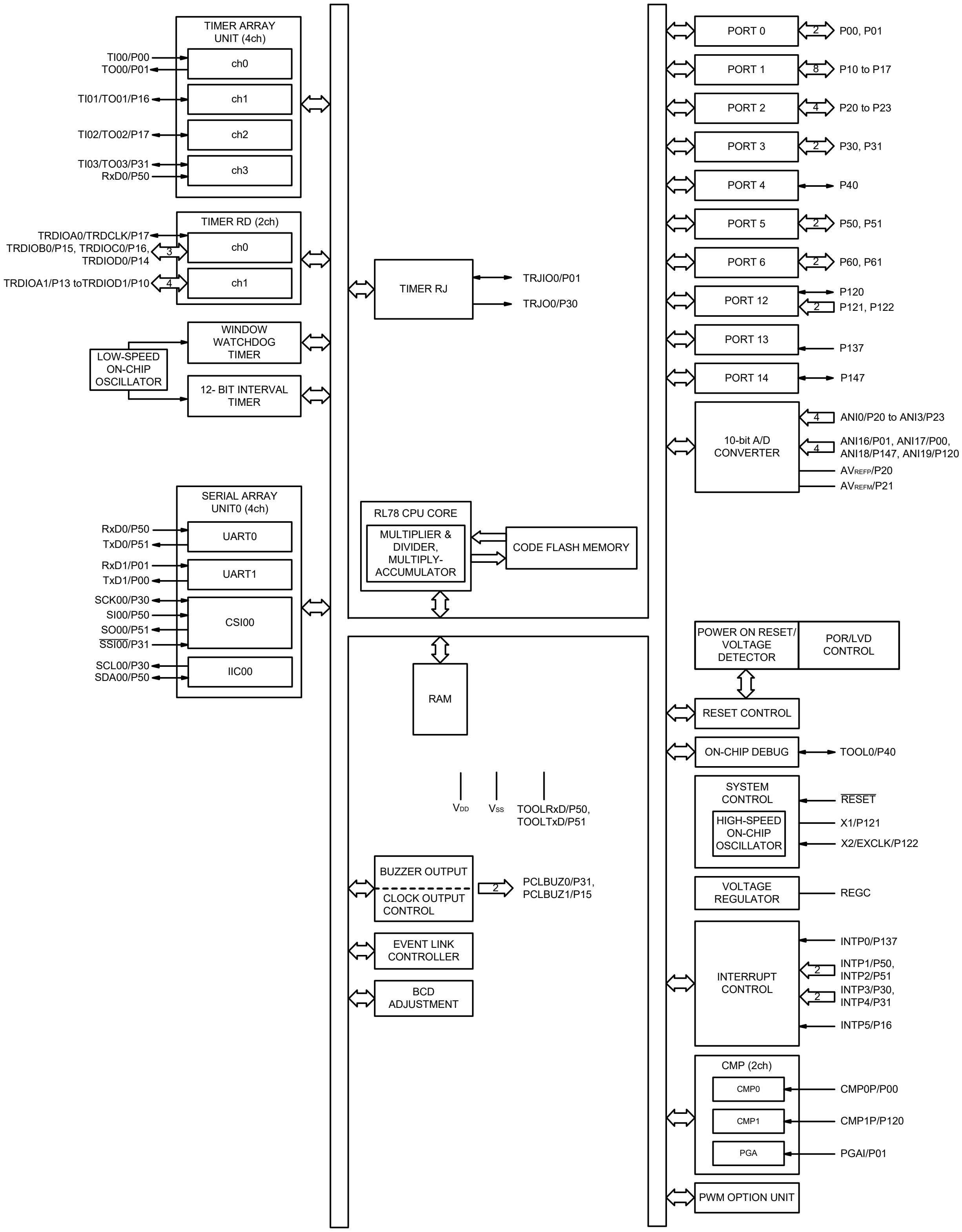
Remark 2. The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

1.4 Pin Identification

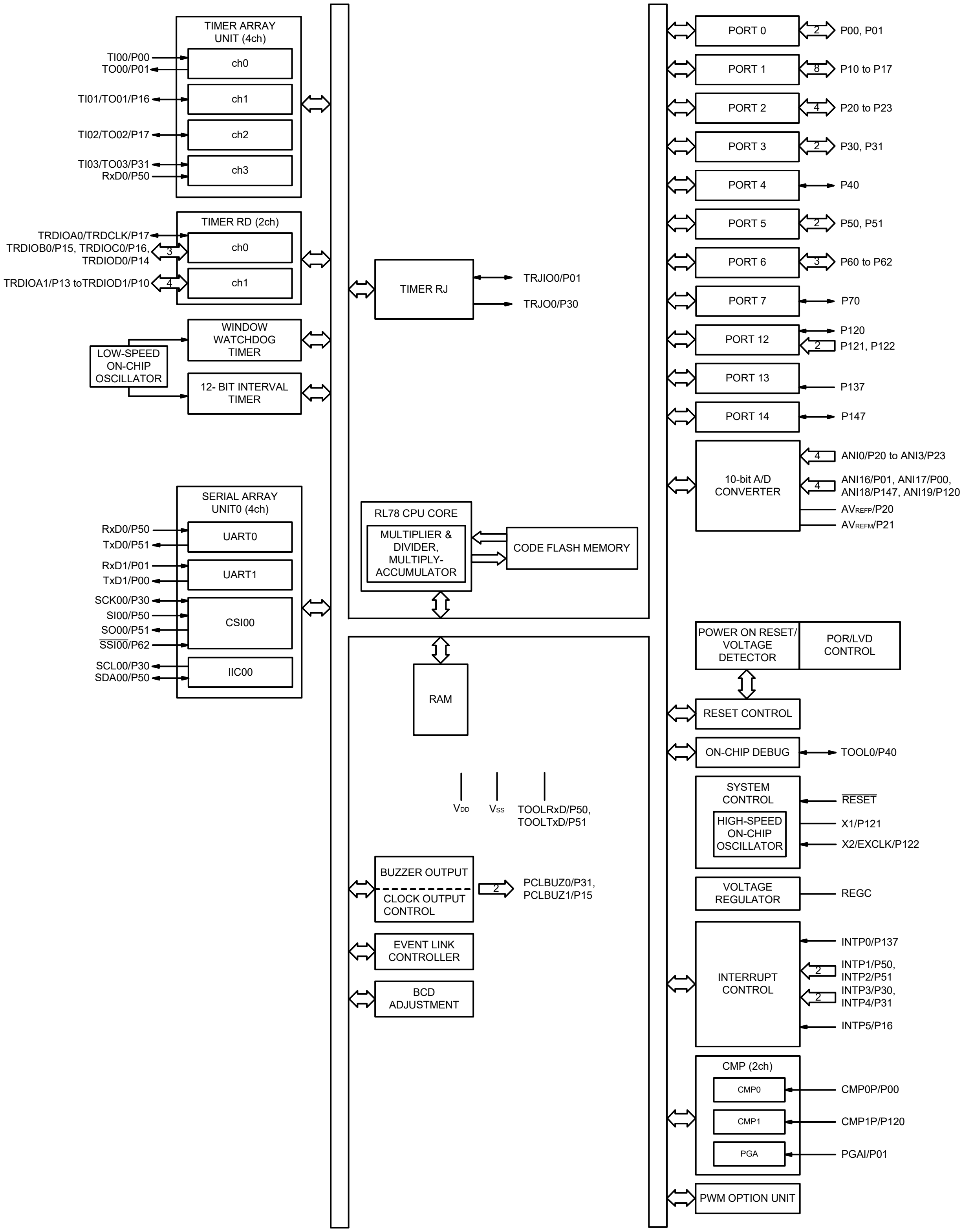
ANI0 to ANI7, ANI16 to ANI19:	Analog input
AVREFM:	A/D converter reference potential (- side) input
AVREFP:	A/D converter reference potential (+ side) input
EXCLK:	External clock input (main system clock)
INTP0 to INTP5:	External interrupt input
KR0 to KR3:	Key Return
P00, P01:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30, P31:	Port 3
P40, P41:	Port 4
P50, P51:	Port 5
P60 to P63:	Port 6
P70 to P73:	Port 7
P120 to P124:	Port 12
P137:	Port 13
P146, P147:	Port 14
PCLBUZ0, PCLBUZ1:	Programmable clock output/buzzer output
REGC:	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
RxD0, RxD1:	Receive data
SCK00:	Serial clock input/output
SCL00:	Serial clock output
SDA00:	Serial data input/output
SI00:	Serial data input
SO00:	Serial data output
$\overline{\text{SSI00}}$:	Serial interface chip select input
TI00 to TI03:	Timer input
TO00 to TO03, TRJO0:	Timer output
TOOL0:	Data input/output for tool
TOOLRxD, TOOLTxD:	Data input/output for external device
TRDCLK:	Timer external input clock
TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0,: TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1, TRJIO0	Timer input/output
TxD0, TxD1:	Transmit data
CMP0P, CMP1P:	Comparator input
PGAI:	PGA input
VDD:	Power supply
VSS:	Ground
X1, X2:	Crystal oscillator (main system clock)

1.5 Block Diagram

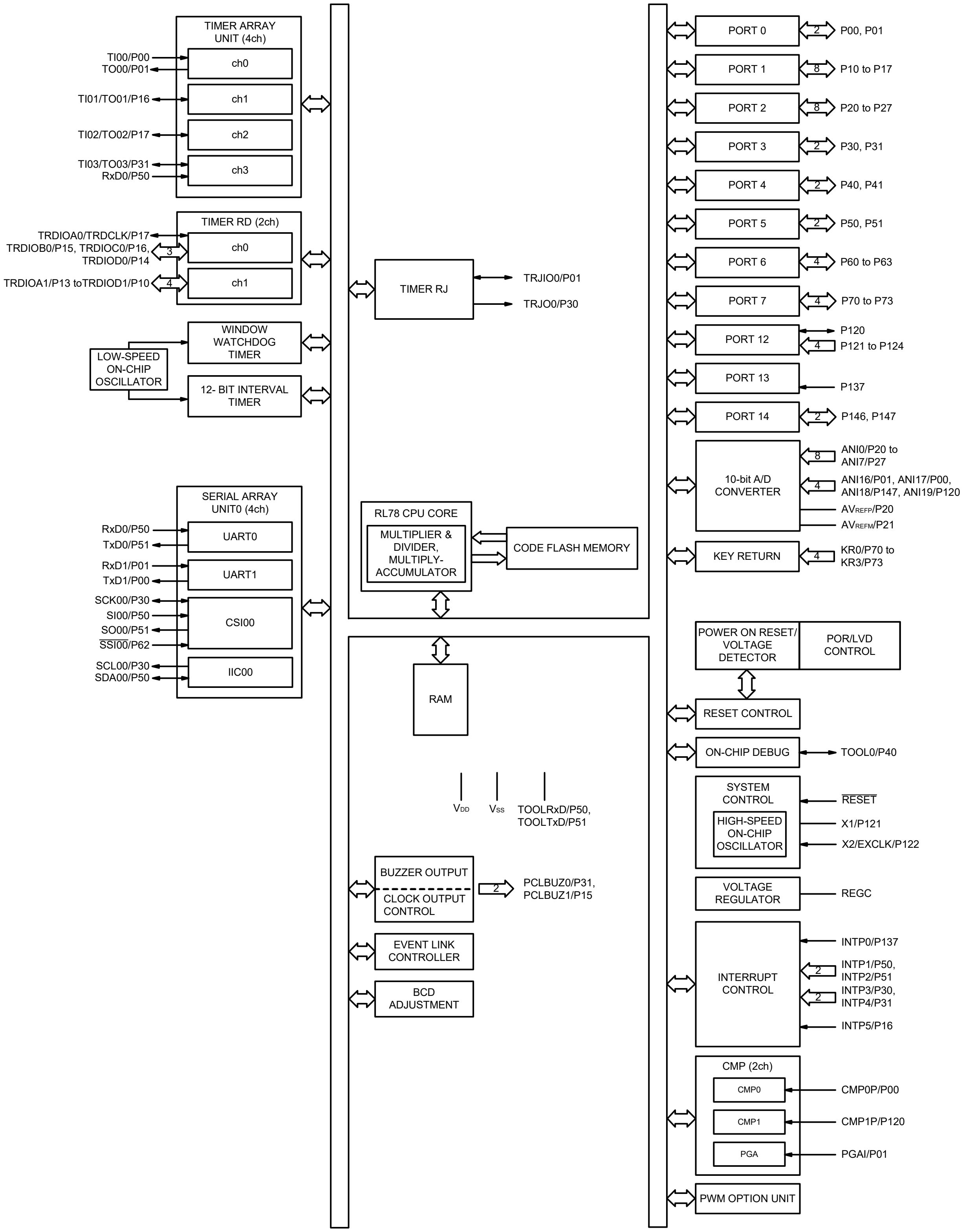
1.5.1 30-pin products



1.5.2 32-pin products



1.5.3 44-pin products



1.6 Outline of Functions

[30-pin, 32-pin, 44-pin products (code flash memory 8 KB to 16 KB)]

Caution The above outline of the functions applies when peripheral I/O redirection register 1 (PIOR1) is set to 00H.

(1/2)

Item		30-pin	32-pin	44-pin
		R5F11EA8ASP, R5F11EAAASP	R5F11EB8AFP, R5F11EBAAFP	R5F11EF8AFP, R5F11EFAAFP
Code flash memory (KB)		8 to 16		
RAM (KB)		1.5		
Address space		1 MB		
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 2.7 to 5.5 V), HS (high-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V)		
	High-speed on-chip oscillator clock (f_{IH})	LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 2.7 to 5.5 V) HS (high-speed main) mode: 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V)		
Low-speed on-chip oscillator clock		15 kHz (TYP.): V_{DD} = 2.7 to 5.5 V		
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: f_{IH} = 24 MHz operation)		
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)		
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), Division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and Accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.		
I/O port	Total	26	28	40
	CMOS I/O	23	25	35
	CMOS input	3	3	5
	CMOS output	—		
	N-ch open-drain I/O (6 V tolerance)	—		
Timer	16-bit timer	7 channels (TAU: 4 channels, Timer RJ: 1 channel, Timer RD: 2 channels)		
	Watchdog timer	1 channel		
	12-bit interval timer	1 channel		
	Timer output	Timer outputs: 14 channels PWM outputs: 9 channels		

Caution Since a library is used when rewriting the flash memory using the user program, flash ROM and RAM areas are used. Refer to the RL78 Family Flash Self-Programming Library Type01 User's Manual before using these products.

(2/2)

Item		30-pin	32-pin	44-pin
		R5F11EA8ASP, R5F11EAAASP	R5F11EB8AFP, R5F11EBAAFP	R5F11EF8AFP, R5F11EFAAFP
Clock output/buzzer output		2		
		• 2.44 kHz, 4.88 kHz, 9.77 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation)		
8/10-bit resolution A/D converter		8 channels		12 channels
Comparator		2 channels		
PGA		1 channel		
Serial interface		• Simplified SPI (CSI): 1 channel/UART0: 1 channel/simplified I^2C: 1 channel • UART1: 1 channel		
Event link controller (ELC)		Event input: 18 Event trigger output: 6		Event input: 19 Event trigger output: 6
Vectored interrupt sources	Internal	20		
	External	6		7
Key interrupt		—		4
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution Note • Internal reset by RAM parity error • Internal reset by illegal-memory access		
Power-on-reset circuit		• Power-on-reset: 1.51 ±0.03 V • Power-down-reset: 1.50 ±0.03 V		
Voltage detector		2.75 V to 4.06 V (6 stages)		
On-chip debug function		Provided		
Power supply voltage		V_{DD} = 2.7 to 5.5 V		
Operating ambient temperature		T_A = -40 to +85°C		

Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

Pin I/O buffers are powered by a single power supply in all products.

The input and output, buffer, and pull-up resistor settings for each port are also valid for the alternate function.

Table 2 - 1 Pin I/O Buffer Power Supplies

30-pin, 32-pin, 44-pin products

Power Supply	Corresponding Pins
V_{DD}	All pins

2.1.1 30-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI17/TI00/TxD1/ CMP0P/(TRJO0)
P01				ANI16/TO00/RxD1/ PGAI/(TRJIO0)
P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P15 to P17 can be set to TTL input buffer. Output of P10, P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TRDIOD1
P11				TRDIOC1
P12				TRDIOB1
P13				TRDIOA1
P14				TRDIOD0
P15				PCLBUZ1/TRDIOB0
P16				TI01/TO01/INTP5/ TRDIOC0
P17				TI02/TO02/ TRDIOA0/TRDCLK
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P30	I/O	Port 3. 2-bit I/O port. Input of P30 and P31 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP3/SCK00/ SCL00/TRJO0
P31				TI03/TO03/INTP4/ PCLBUZ0/$\overline{SSI00}$/ (TRJIO0)
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 2-bit I/O port. Input of P50 can be set to TTL input buffer. Output of P50 and P51 can be set to N-ch open drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI00/RxD0/TOOLRxD/SDA00/(TRJO0)
P51				INTP2/SO00/TxD0/TOOLTxD
P60	I/O	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units.	Input port	—
P61				—
P120	I/O	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19/CMP1P
P121	Input		Input port	X1
P122				X2/EXCLK
P137	Input	Port 13. 1-bit input port.	Input port	INTP0
P147	I/O	Port 14. 1-bit I/O port. P147 can be set to analog input. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI18

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

2.1.2 32-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI17/CMP0P/TI00/ TxD1/(TRJO0)
P01				ANI16/PGAI/TO00/ RxD1/TRJIO0
P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P15 to P17 can be set to TTL input buffer. Output of P10, P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TRDIOD1
P11				TRDIOC1
P12				TRDIOB1
P13				TRDIOA1
P14				TRDIOD0
P15				PCLBUZ1/TRDIOB0
P16				TI01/TO01/INTP5/ TRDIOC0
P17				TI02/TO02/ TRDIOA0/TRDCLK
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P30	I/O	Port 3. 2-bit I/O port. Input of P30 and P31 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP3/SCK00/ SCL00/TRJO0
P31				TI03/TO03/INTP4/ PCLBUZ0/(TRJIO0)
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 2-bit I/O port. Input of P50 can be set to TTL input buffer. Output of P50 and P51 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI00/RxD0/TOOLRxD/SDA00/(TRJO0)
P51				INTP2/SO00/TxD0/TOOLTxD
P60	I/O	Port 6. 3-bit I/O port. Input/output can be specified in 1-bit units.	Input port	—
P61				—
P62				$\overline{\text{SSI00}}$
P70	I/O	Port 7. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	—
P120	I/O	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19/CMP1P
P121	Input		Input port	X1
P122				X2/EXCLK
P137	Input	Port 13. 1-bit input port.	Input port	INTP0
P147	I/O	Port 14. 1-bit I/O port. P147 can be set to analog input. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI18

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

2.1.3 44-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI17/CMP0P/TI00/TxD1/(TRJO0)
P01				ANI16/PGAI/TO00/RxD1/TRJIO0
P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P15 to P17 can be set to TTL input buffer. Output of P10, P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TRDIOD1
P11				TRDIOC1
P12				TRDIOB1
P13				TRDIOA1
P14				TRDIOD0
P15				PCLBUZ1/TRDIOB0
P16				TI01/TO01/INTP5/TRDIOC0
P17				TI02/TO02/TRDIOA0/TRDCLK
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P24				ANI4
P25				ANI5
P26				ANI6
P27				ANI7
P30	I/O	Port 3. 2-bit I/O port. Input of P30 and P31 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP3/SCK00/SCL00/TRJO0
P31				TI03/TO03/INTP4/PCLBUZ0/(TRJIO0)
P40	I/O	Port 4. 2-bit I/O port. Input/output can be specified. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				(TRJIO0)

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 2-bit I/O port. Input of P50 can be set to TTL input buffer. Output of P50 and P51 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI00/RxD0/TOOLRxD/SDA00/(TRJO0)
P51				INTP2/SO00/TxD0/TOOLTxD
P60	I/O	Port 6. 4-bit I/O port. Input/output can be specified in 1-bit units.	Input port	—
P61				—
P62				$\overline{\text{SSI00}}$
P63				—
P70	I/O	Port 7. 4-bit I/O port. Input/output can be specified. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0
P71				KR1
P72				KR2
P73				KR3
P120	I/O	Port 12. 1-bit I/O port and 4-bit input-only port. P120 can be set to analog input. For only P120 input/output can be specified in 1-bit units. For only P120 use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19/CMP1P
P121	Input		Input port	X1
P122				X2/EXCLK
P123				—
P124				—
P137	Input	Port 13. 1-bit input port.	Input port	INTP0
P146	I/O	Port 14. 2-bit I/O port. P147 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	—
P147				ANI18

Remark The functions in parentheses shown in the above figure can be assigned by setting peripheral I/O redirection register 1 (PIOR1).

2.2 Functions Other Than Port Pins

2.2.1 With functions for each product

Function Name	44-pin	32-pin	30-pin
ANI0	√	√	√
ANI1	√	√	√
ANI2	√	√	√
ANI3	√	√	√
ANI4	√	—	—
ANI5	√	—	—
ANI6	√	—	—
ANI7	√	—	—
ANI16	√	√	√
ANI17	√	√	√
ANI18	√	√	√
ANI19	√	√	√
CMP0P	√	√	√
CMP1P	√	√	√
PGAI	√	√	√
INTP0	√	√	√
INTP1	√	√	√
INTP2	√	√	√
INTP3	√	√	√
INTP4	√	√	√
INTP5	√	√	√
KR0	√	—	—
KR1	√	—	—
KR2	√	—	—
KR3	√	—	—
PCLBUZ0	√	√	√
PCLBUZ1	√	√	√
REGC	√	√	√
$\overline{\text{RESET}}$	√	√	√
RxD0	√	√	√
RxD1	√	√	√
SCK00	√	√	√
SCL00	√	√	√
SDA00	√	√	√
SI00	√	√	√
SO00	√	√	√
$\overline{\text{SSI00}}$	√	√	√

Function Name	44-pin	32-pin	30-pin
TI00	√	√	√
TI01	√	√	√
TI02	√	√	√
TI03	√	√	√
TO00	√	√	√
TO01	√	√	√
TO02	√	√	√
TO03	√	√	√
TRJIO0	√	√	√
TRJO0	√	√	√
TRDCLK	√	√	√
TRDIOA0	√	√	√
TRDIOB0	√	√	√
TRDIOC0	√	√	√
TRDIOD0	√	√	√
TRDIOA1	√	√	√
TRDIOB1	√	√	√
TRDIOC1	√	√	√
TRDIOD1	√	√	√
TxD0	√	√	√
TxD1	√	√	√
X1	√	√	√
X2	√	√	√
EXCLK	√	√	√
V_{DD}	√	√	√
AV_{REFP}	√	√	√
AV_{REFM}	√	√	√
V_{SS}	√	√	√
TOOLRxD	√	√	√
TOOLTxD	√	√	√
TOOL0	√	√	√

2.2.2 Pins for each product (pins other than port pins)

Function Name	I/O	Function
ANI0 to ANI7, ANI16 to ANI19	Input	A/D converter analog input (see **Figure 12 - 46 Analog Input Pin Connection**)
CMP0P, CMP1P	Input	Comparator input
PGAI	Input	PGA input
INTP0 to INTP5	Input	External interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
KR0 to KR3	Input	Key interrupt input
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	—	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.
RxD0, RxD1	Input	Serial data input pins of serial interface UART0 and UART1
TxD0, TxD1	Output	Serial data output pins of serial interface UART0 and UART1
SCK00	I/O	Serial clock I/O pins of serial interface CSI00
SCL00	Output	Serial clock output pins of serial interface IIC00
SDA00	I/O	Serial data I/O pins of serial interface IIC00
SI00	Input	Serial data input pins of serial interface CSI00
$\overline{\text{SSI00}}$	Input	Chip select input pin of serial interface CSI00
SO00	Output	Serial data output pins of serial interface CSI00
TI00 to TI03	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 03
TO00 to TO03	Output	Timer output pins of 16-bit timers 00 to 03
TRJIO0	I/O	Timer RJ input/output
TRJO0	Output	Timer RJ output
TRDCLK	Input	Timer RD external clock input
TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1	I/O	Timer RD input/output
X1, X2	—	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
V_{DD}	—	Positive power supply for all pins
AV_{REFP}	Input	A/D converter reference potential (+ side) input
AV_{REFM}	Input	A/D converter reference potential (- side) input
V_{SS}	—	Ground potential for all pins
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming
TOOL0	I/O	Data I/O for flash memory programmer/debugger

Caution After reset release, the relationships between P40/TOOL0 and the operating mode are as follows.

Table 2 - 2 Relationships Between P40/TOOL0 and Operation Mode After Reset Release

P40/TOOL0	Operating mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

For details, see 25.4 Programming Method.

Remark Use bypass capacitors (about 0.1 μF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS} lines.

2.3 Connection of Unused Pins

Tables 2 - 3 and 2 - 4 show the Connection of Unused Pins.

Table 2 - 3 Connection of Unused Pins (44-pin Products) (1/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P00/ANI17/TI00/TxD1/CMP0P/(TRJO0)	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/ANI16/TO00/RxD1/PGAI/TRJIO0		
P10/TRDIOD1		
P11/TRDIOC1		
P12/TRDIOB1		
P13/TRDIOA1		
P14/TRDIOD0		
P15/PCLBUZ1/TRDIOB0		
P16/TI01/TO01/INTP5/TRDIOC0		
P17/TI02/TO02/TRDIOA0/TRDCLK		
P20/ANI0/AV_{REFP}		
P21/ANI1/AV_{REFM}		
P22/ANI2		
P23/ANI3		
P24/ANI4		
P25/ANI5		
P26/ANI6		
P27/ANI7		
P30/INTP3/SCK00/SCL00/TRJO0		
P31/TI03/TO03/INTP4/PCLBUZ0/(TRJIO0)		
P40/TOOL0		Input: Independently connect to V_{DD} via a resistor, or leave open. Output: Leave open.
P41/(TRJIO0)		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P50/INTP1/SI00/RxD0/TOOLRxD/SDA00/(TRJO0)		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P51/INTP2/SO00/TxD0/TOOLTxD		
P60		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P61		
P62/$\overline{\text{SSI00}}$		
P63		
P70/KR0		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P71/KR1		
P72/KR2		
P73/KR3		

Table 2 - 4 Connection of Unused Pins (44-pin Products) (2/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P120/ANI19	I/O	Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P121/X1	Input	Independently connect to VDD or VSS via a resistor.
P122/X2/EXCLK		
P123		
P124		
P137/INTP0		Independently connect to VDD or VSS via a resistor.
P146	I/O	Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P147/ANI18		
$\overline{\text{RESET}}$	Input	Connect to VDD directly or via a resistor.
REGC	—	Connect to VSS via a capacitor (0.47 to 1 μF: target).

CHAPTER 3 CPU ARCHITECTURE

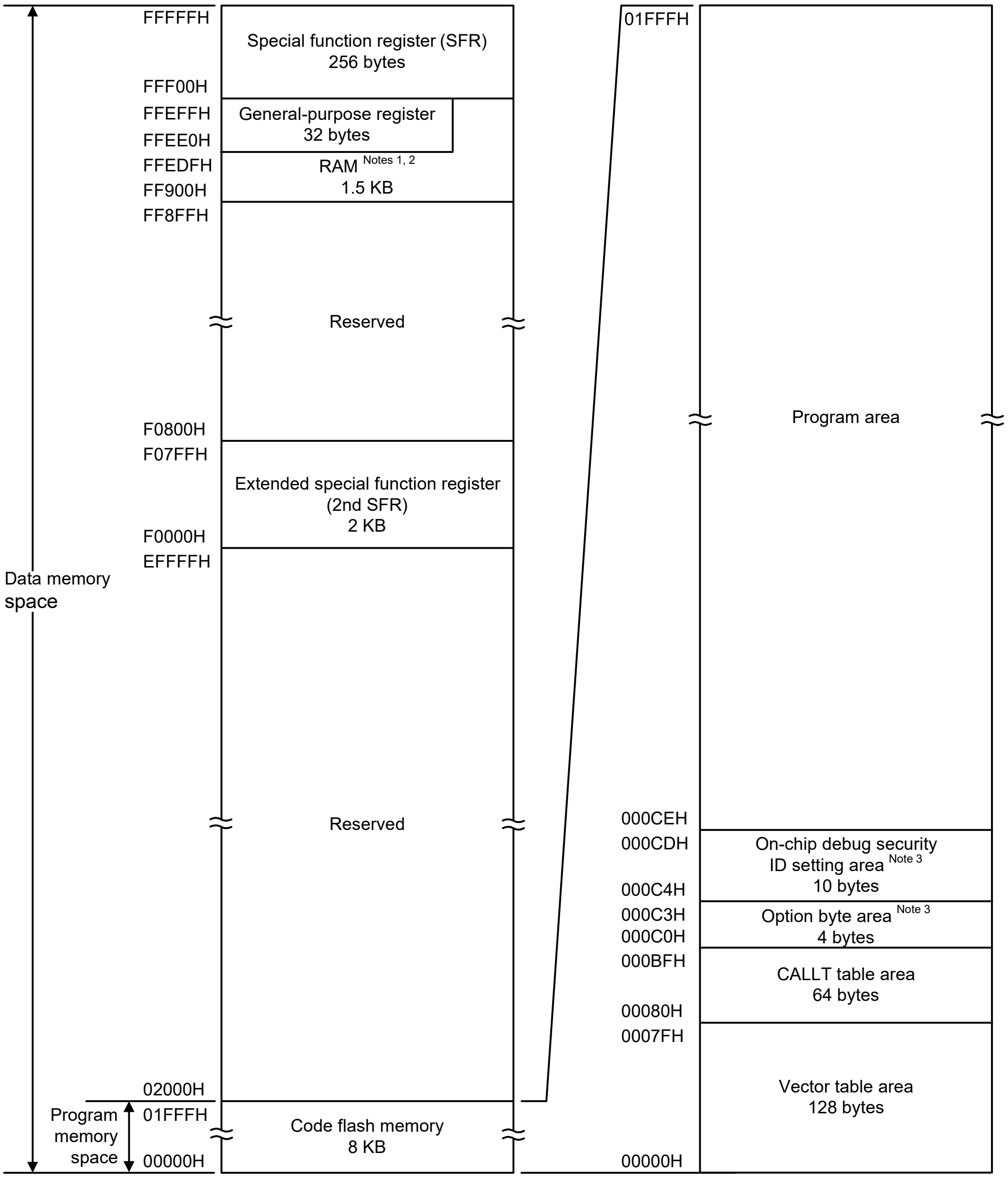
3.1 Overview

The CPU core in the RL78 microcontroller employs the Harvard architecture which has independent instruction fetch bus, address bus and data bus. In addition, through the adoption of three-stage pipeline control of fetch, decode,and memory access, the operation efficiency is remarkably improved over the conventional CPU core. The CPU core features high performance and highly functional instruction processing, and can be suited for use in various applications that require high speed and highly functional processing.

The RL78/G1G integrates the RL78-S3 core that has the following features.

- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time : One instruction per clock cycle
- General-purpose registers: Eight 8-bit registers
- Type of instruction: 81
- Data allocation: Little endian
- Multiply/divide and multiply/accumulate instructions: Supported

Figure 3 - 1 Memory Map (R5F11EA8, R5F11EB8, R5F11EF8)



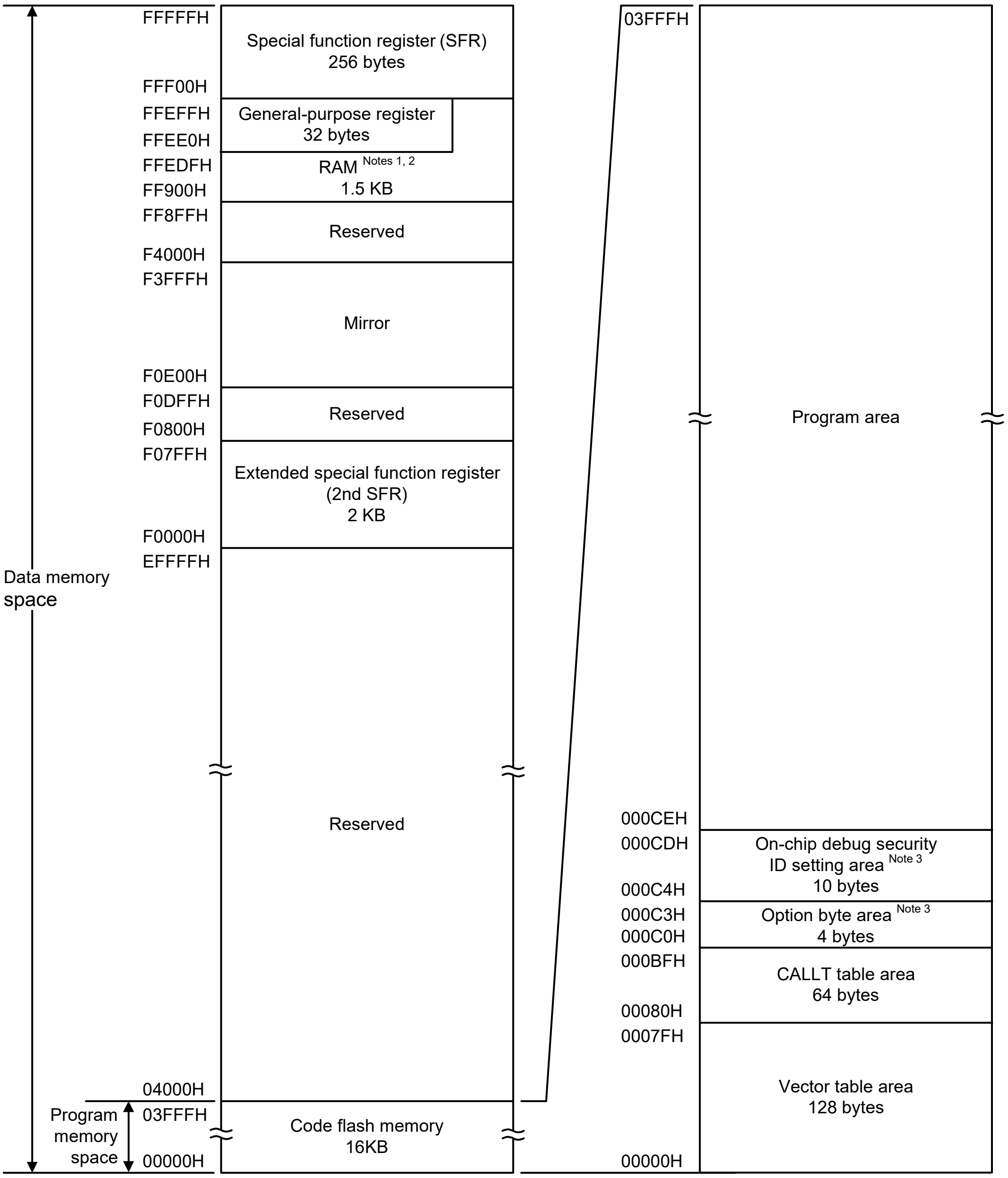
Note 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing to the area FFE20H to FFEFFH and FF900H to FFC80H when performing self-programming.

Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 22.5 RAM Parity Error Detection Function.

Figure 3 - 2 Memory Map (R5F11EAA, R5F11EBA, R5F11EFA)



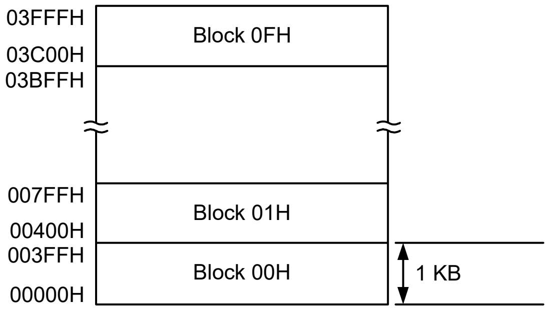
Note 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing to the area FFE20H to FFEFFH and FF900H to FFC80H when performing self-programming.

Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 22.5 RAM Parity Error Detection Function.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3 - 1 Correspondence between Address Values and Block Numbers in Flash Memory**.



(R5F11ExA (x = A, B, F))

Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3 - 1 Correspondence between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	02000H to 023FFH	08H
00400H to 007FFH	01H	02400H to 027FFH	09H
00800H to 00BFFH	02H	02800H to 02BFFH	0AH
00C00H to 00FFFH	03H	02C00H to 02FFFH	0BH
01000H to 013FFH	04H	03000H to 033FFH	0CH
01400H to 017FFH	05H	03400H to 037FFH	0DH
01800H to 01BFFH	06H	03800H to 03BFFH	0EH
01C00H to 01FFFH	07H	03C00H to 03FFFH	0FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.
The RL78/G1G products incorporate internal ROM (flash memory), as shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F11EA8, R5F11EB8, R5F11EF8	Flash memory	8192 × 8 bits (00000H to 01FFFH)
R5F11EAA, R5F11EBA, R5F11EFA		16384 × 8 bits (00000H to 03FFFH)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump address is a 64 K address of 00000H to 0FFFFH, because the vector code is assumed to be 2 bytes.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3 - 3 list the vector table. “√” indicates an interrupt source which is supported. “—” indicates an interrupt source which is not supported.

Table 3 - 3 Vector Table

Vector Table Address	Interrupt Source	44-pin	32-pin	30-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√
00004H	INTWDTI	√	√	√
00006H	INTLVI	√	√	√
00008H	INTP0	√	√	√
0000AH	INTP1	√	√	√
0000CH	INTP2	√	√	√
0000EH	INTP3	√	√	√
00010H	INTP4	√	√	√
00012H	INTP5	√	√	√
0001EH	INTST0/INTCSI00/INTIIC00	√	√	√
00020H	INTSR0	√	√	√
00022H	INTSRE0	√	√	√
	INTTM01H	√	√	√
00024H	INTST1	√	√	√
00026H	INTSR1	√	√	√
00028H	INTSRE1	√	√	√
	INTTM03H	√	√	√
0002CH	INTTM00	√	√	√
0002EH	INTTM01	√	√	√
00030H	INTTM02	√	√	√
00032H	INTTM03	√	√	√
00034H	INTAD	√	√	√
00038H	INTIT	√	√	√
0003AH	INTKR	√	—	—
00040H	INTTRJ0	√	√	√
00052H	INTCMP0	√	√	√
00054H	INTCMP1	√	√	√
00056H	INTTRD0	√	√	√
00058H	INTTRD1	√	√	√
00062H	INTFL	√	√	√
0007EH	BRK	√	√	√

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 24 OPTION BYTE**.

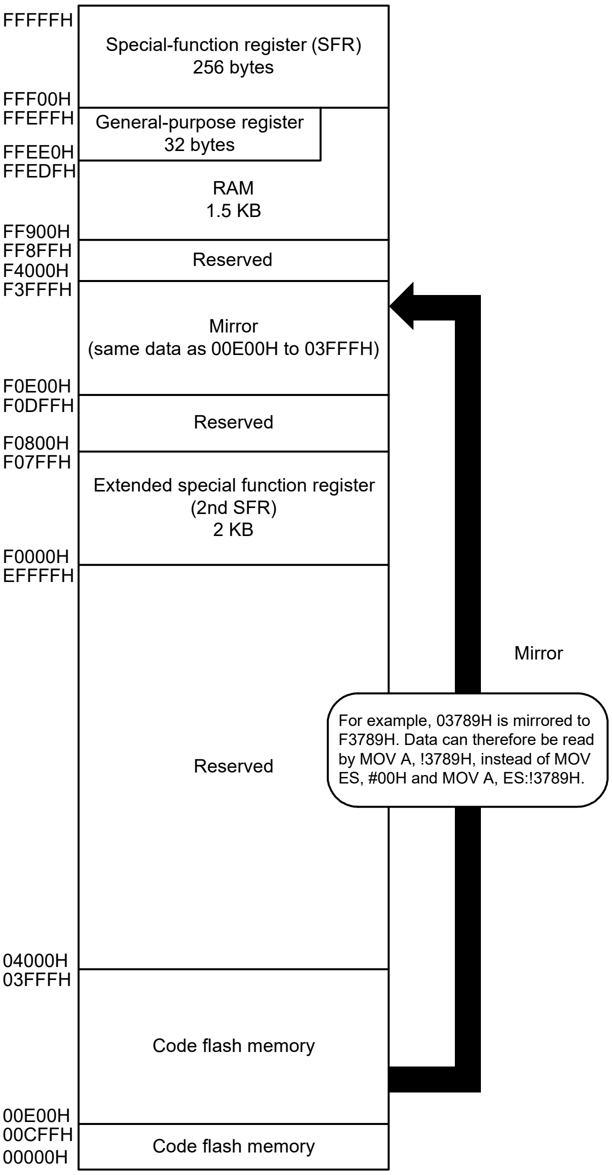
(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The R5F11EBAAFP mirror the code flash area of 00000H to 03FFFH, to F0000H to FFFFFH.
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.
See **3.1 Overview** for the mirror area of each product.
The mirror area can only be read and no instruction can be fetched from this area.
The following show examples.

Example R5F11EBAAFP (Flash memory: 16 KB, RAM: 1.5 KB)



For example, 03789H is mirrored to F3789H. Data can therefore be read by MOV A, !3789H, instead of MOV ES, #00H and MOV A, ES:!3789H.

The PMC register is described below.

- Processor mode control register (PMC)

  This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

  The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.

  Reset signal generation sets this register to 00H.

Figure 3 - 3 Format of Configuration of Processor mode control register (PMC)

Address: FFFFEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 03FFFH is mirrored to F0000H to FFFFFH
1	Setting prohibited

Caution 1. Be sure to clear bit 0 (MAA) of this register to 0 (default value).

Caution 2. After setting the PMC register, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

The RL78/G1G products incorporate the following RAMs.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R5F11EA8, R5F11EB8, R5F11EF8 R5F11EAA, R5F11EBA, R5F11EFA	1536 × 8 bits (FF900H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions). Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.

The internal RAM is used as stack memory.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Caution 2. When self-programming, do not allocate the stack used for the library and the RAM address used for the data buffer to the FFE20H to FFEFFH of the RAM area. For more details, refer to the RL78 Family Flash Self-Programming Library Type01 User’s Manual.

Caution 3. Since the FF900H to FFC80H area of the internal RAM area is used for the self-programming library, this area cannot be used.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Tables 3 - 5** to **3 - 7** in **3.2.4 Special function registers (SFRs)**).

Caution Do not access addresses to which SFRs are not assigned.

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Tables 3 - 8** to **3 - 13** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

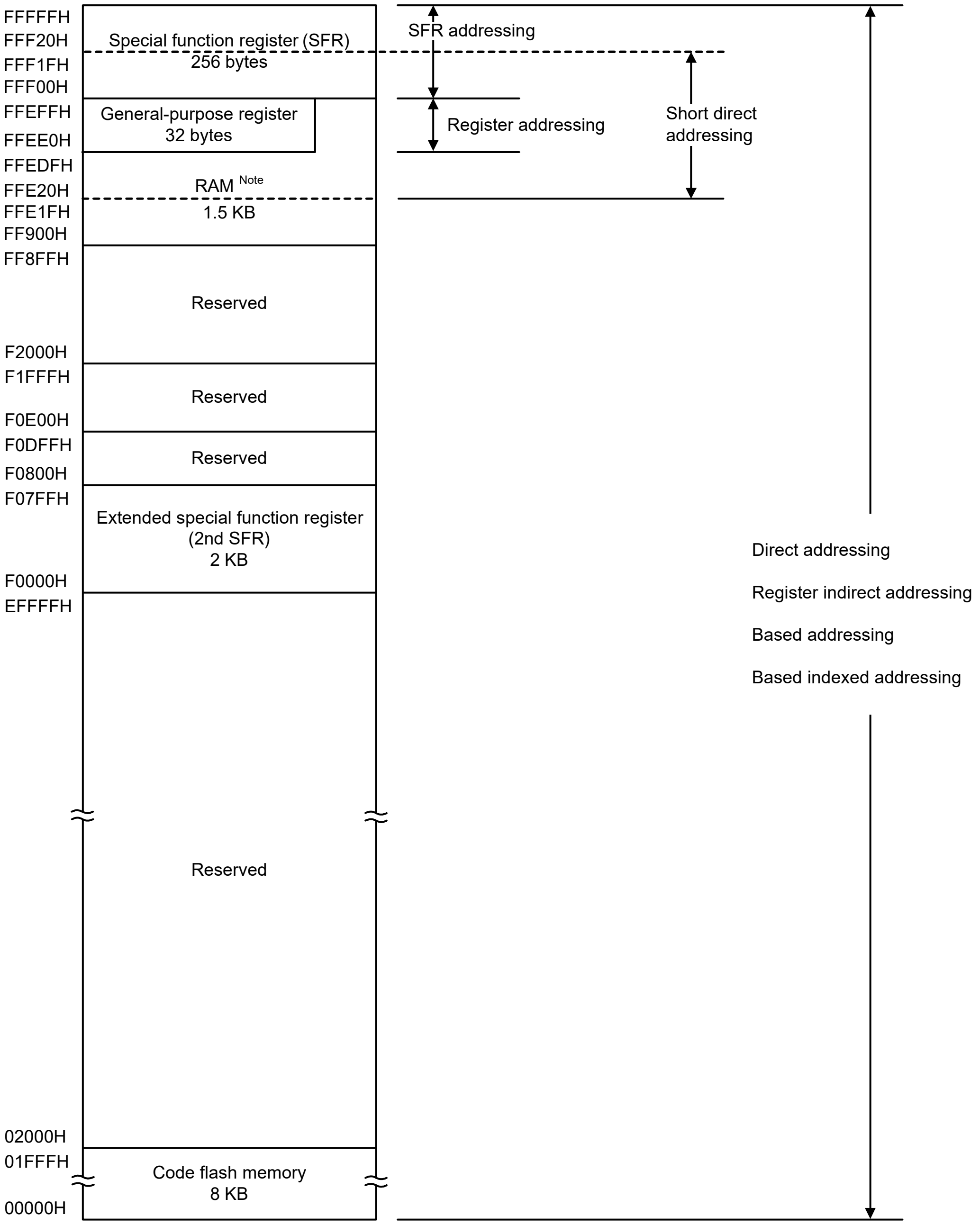
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

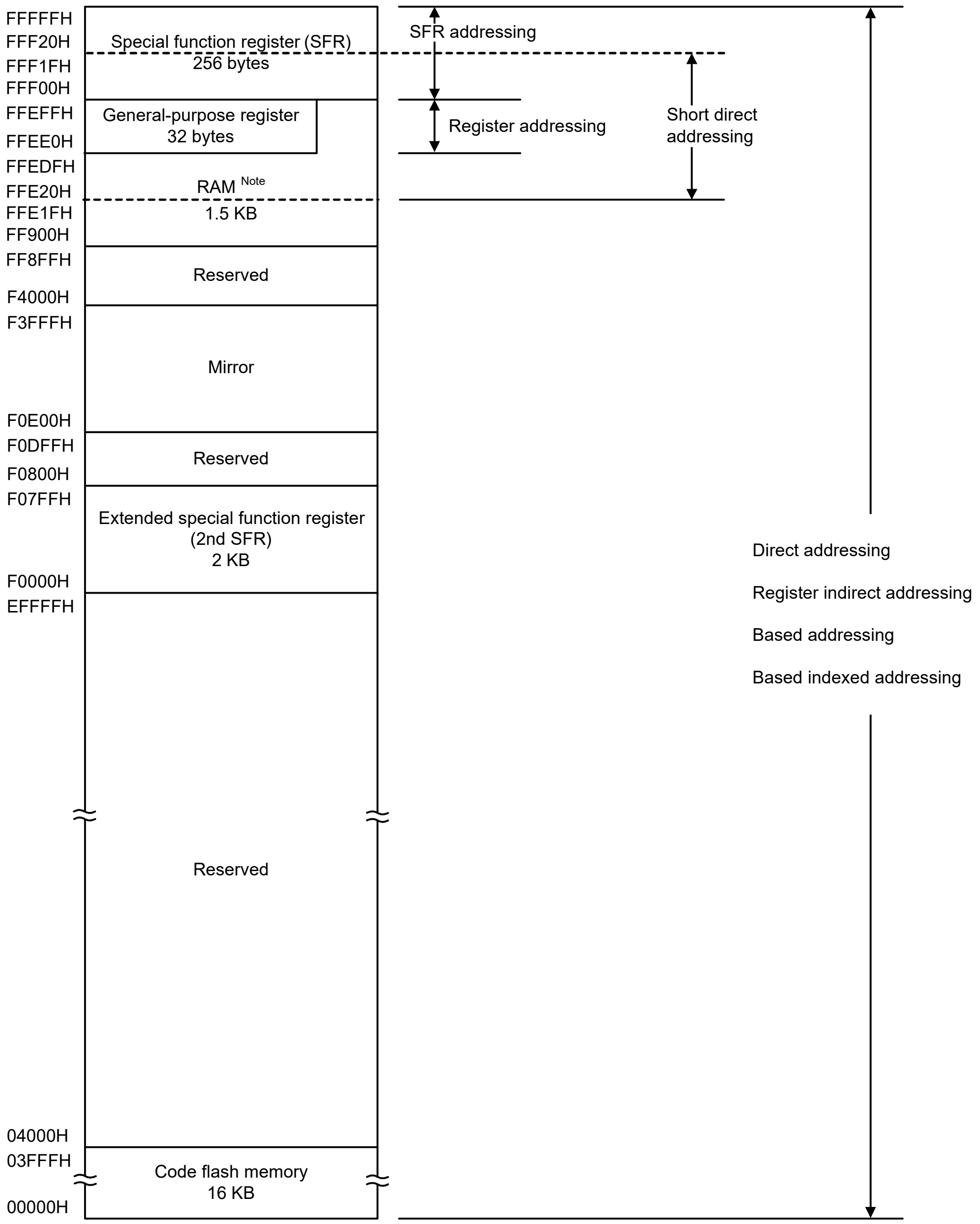
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/G1G, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figures 3 - 4 and 3 - 5 show correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 4 Correspondence between Data Memory and Addressing (R5F11EA8, R5F11EB8, R5F11EF8)



Note Use of the area FFE20H to FFEFFH and FF900H to FFC80H is prohibited when using the self-programming function, because this area is used for self-programming library.

Figure 3 - 5 Correspondence between Data Memory and Addressing (R5F11EAA, R5F11EBA, R5F11EFA)



Note Use of the area FFE20H to FFEFFH and FF900H to FFC80H is prohibited when using the self-programming function, because this area is used for self-programming library.

3.2 Processor Registers

The RL78/G1G products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

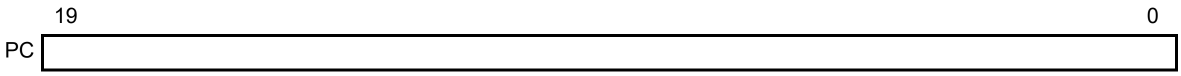
(1) Program counter (PC)

The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 00000H and 00001H to the program counter.

Figure 3 - 6 Format of Program Counter



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets the PSW register to 06H.

Figure 3 - 7 Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

(c) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flags (ISP1, ISP0)

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **16.3.3**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

Figure 3 - 8 Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

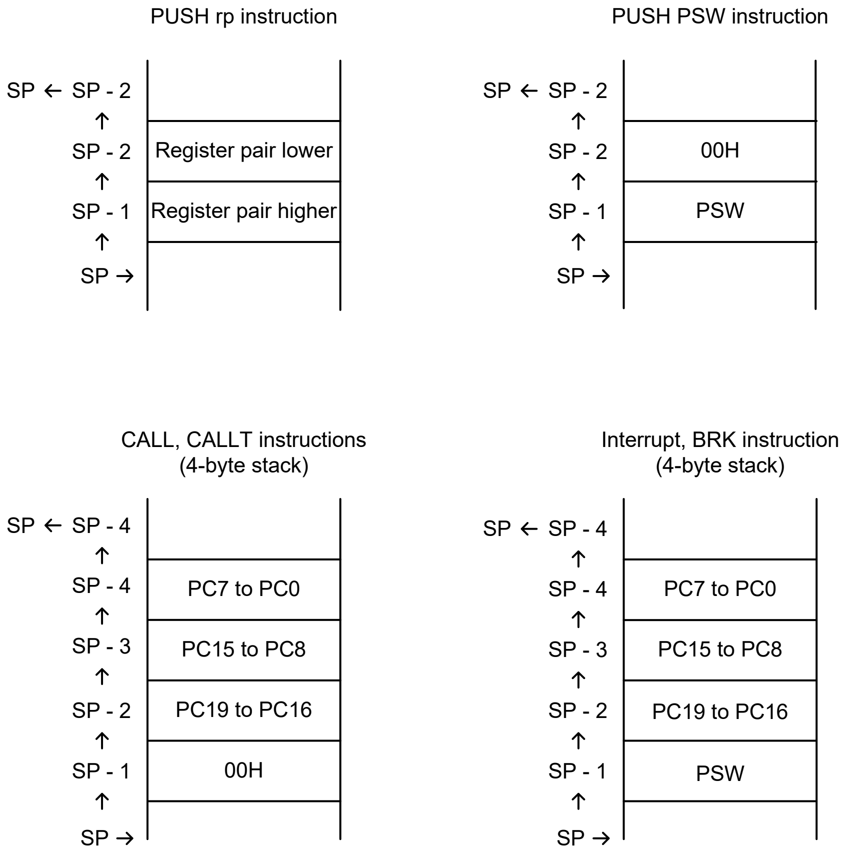
Caution 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Caution 2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.

Caution 3. When self-programming, do not allocate the stack used for the library and the RAM address used for the data buffer to the FFE20H to FFEFFH of the RAM area. For more details, refer to the RL78 Family Flash Self-Programming Library Type01 User's Manual.

Caution 4. Since the FF900H to FFC80H area of the internal RAM area is used for the self-programming library, this area cannot be used.

Figure 3 - 9 Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

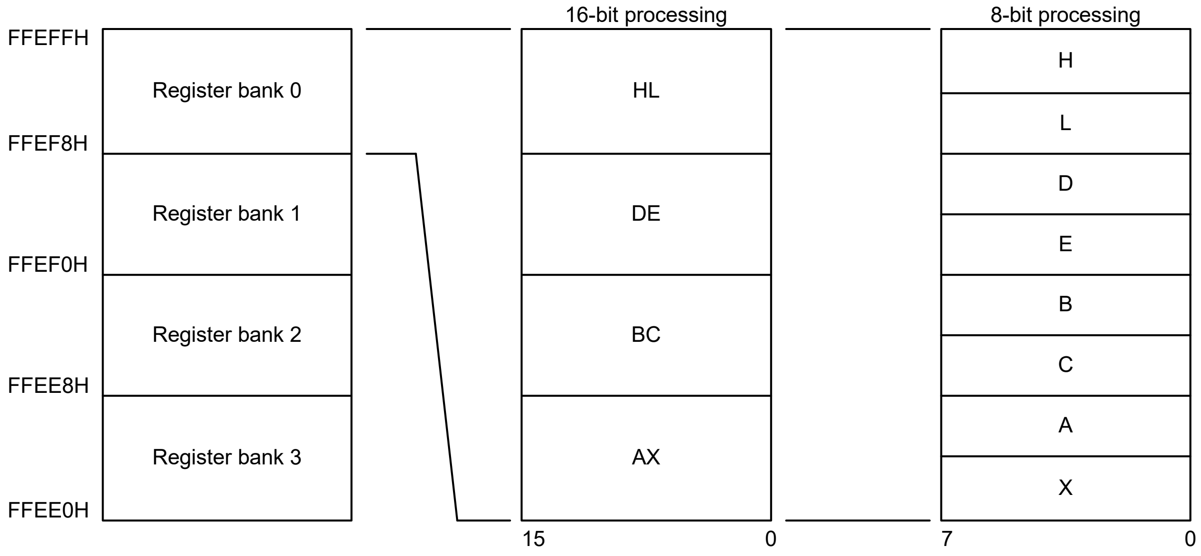
Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Caution 2. When self-programming, do not allocate the stack used for the library and the RAM address used for the data buffer to the FFE20H to FFEFFH of the RAM area. For more details, refer to the RL78 Family Flash Self-Programming Library Type01 User's Manual.

Caution 3. Since the FF900H to FFC80H area of the internal RAM area is used for the self-programming library, this area cannot be used.

Figure 3 - 10 Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register direct addressing), respectively.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

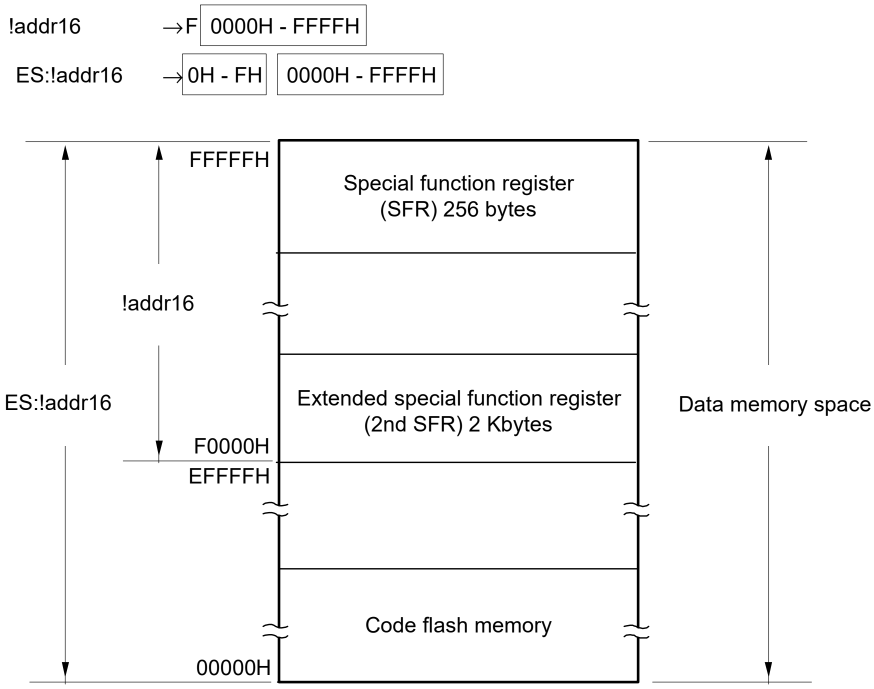
Figure 3 - 11 Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3 - 12 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>.<Bit number> or <Address>.<Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (sfrp).
  When specifying an address, describe an even address.

Tables 3 - 5 to 3 - 7 give lists of the SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

Caution **Do not access addresses to which SFRs are not assigned.**

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3 - 5 Special Function Register (SFR) List (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	—	00H
FFF01H	Port register 1	P1		R/W	√	√	—	00H
FFF02H	Port register 2	P2		R/W	√	√	—	00H
FFF03H	Port register 3	P3		R/W	√	√	—	00H
FFF04H	Port register 4	P4		R/W	√	√	—	00H
FFF05H	Port register 5	P5		R/W	√	√	—	00H
FFF06H	Port register 6	P6		R/W	√	√	—	00H
FFF07H	Port register 7	P7		R/W	√	√	—	00H
FFF0CH	Port register 12	P12		R/W	√	√	—	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	—	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	—	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	√	√	00H
FFF1BH		TDR01H			—	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	—	—	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	√	—	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	—	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	—	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	—	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	—	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	—	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	—	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	—	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	—	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF37H	Key return mode register	KRM		R/W	√	√	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	—	00H

Table 3 - 6 Special Function Register (SFR) List (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF58H	Timer RD general register C0	TRDGRC0		R/W	—	—	√	FFFFH Note
FFF59H								
FFF5AH	Timer RD general register D0	TRDGRD0		R/W	—	—	√	FFFFH Note
FFF5BH								
FFF5CH	Timer RD general register C1	TRDGRC1		R/W	—	—	√	FFFFH Note
FFF5DH								
FFF5EH	Timer RD general register D1	TRDGRD1		R/W	—	—	√	FFFFH Note
FFF5FH								
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√		00H
FFF90H	12-bit interval timer control register	ITMC		R/W	—	—	√	0FFFH
FFF91H								
FFFA0H	Clock operation mode control register	CMC		R/W	—	√	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	√	—	07H
FFFA4H	System clock control register	CKC		R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	—	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	—	00H

Note The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 3 - 7 Special Function Register (SFR) List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00H Note 2
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	—	00H/01H/81H Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	—	√	—	9AH/1AH Note 4
FFFACH	CRC input register	CRCIN		R/W	—	√	—	00H
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H	IF2	R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H	MK2	R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H	PR02	R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H	PR12	R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiply and accumulation register (L)	MACRL		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	—	—	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	—	00H

Note 1. The reset value of the RESF register varies depending on the reset source.

Note 2. The reset value of the LVIM register varies depending on the reset source.

Note 3. The reset value of the LVIS register varies depending on the reset source and the setting of the option byte.

Note 4. The reset value of the WDTE register is determined by the setting of the option byte.

Remark For extended SFRs (2^{nd} SFRs), see **Tables 3 - 8** to **3 - 13 Extended Special Function Register (2nd SFR) List**.

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (!addr16.bit)
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>.<Bit number> or <Address>.<Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Tables 3 - 8 to 3 - 13 give lists of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding extended SFR can be read or written.
  R/W:Read/write enable
  R:Read only
  W:Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3 - 8 Extended Special Function Register (2nd SFR) List (1/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	—	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	—	√	—	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	—	√	—	00H
F0013H	A/D test register	ADTES	R/W	—	√	—	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	—	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	—	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	—	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	—	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	—	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	—	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	—	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	—	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	—	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	—	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	—	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	—	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	—	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	—	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	—	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	—	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	—	FFH
F006EH	Port mode control register 14	PMC14	R/W	√	√	—	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	—	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	—	00H
F0073H	Input switch control register	ISC	R/W	√	√	—	00H
F0074H	Timer input select register 0	TIS0	R/W	—	√	—	00H
F0076H	A/D port configuration register	ADPC	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F0079H	Peripheral I/O redirection register 1	PIOR1	R/W	—	√	—	00H
F007AH	Peripheral enable register 1	PER1	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	—	√	—	Note
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	√	—	Undefined

Note The value after a reset is adjusted at the time of shipment.

Table 3 - 9 Extended Special Function Register (2nd SFR) List (2/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	—	00H
F00F3H	Operation speed mode control register	OSMC		R/W	—	√	—	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	√	—	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	—	√	√	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	√	√	0000H
F0103H		—			—	—		
F0104H	Serial status register 02	SSR02L	SSR02	R	—	√	√	0000H
F0105H		—			—	—		
F0106H	Serial status register 03	SSR03L	SSR03	R	—	√	√	0000H
F0107H		—			—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	√	√	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	√	√	0000H
F010BH		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								

Note This value varies depending on the products.

Table 3 - 10 Extended Special Function Register (2nd SFR) List (3/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	√	√	0000H
F0139H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	—	—	√	FFFFH
F0187H								
F0190H	Timer mode register 00	TMR00		R/W	—	—	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	—	—	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	—	—	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	—	—	√	0000H
F0197H								
F01A0H	Timer status register 00	TSR00L	TSR00	R	—	√	√	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	√	√	0000H
F01A3H		—			—	—		

Table 3 - 11 Extended Special Function Register (2nd SFR) List (4/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	√	√	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	√	√	0000H
F01A7H		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		—			—	—		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	√	√	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—	—		
F0240H	Timer RJ control register 0	TRJCR0		R/W	—	√	—	00H
F0241H	Timer RJ I/O control register 0	TRJIOC0		R/W	√	√	—	00H
F0242H	Timer RJ mode register 0	TRJMR0		R/W	√	√	—	00H
F0243H	Timer RJ event pin select register 0	TRJISR0		R/W	√	√	—	00H
F0260H	Timer RD ELC register	TRDELC		R/W	√	√	—	00H Note
F0263H	Timer RD start register	TRDSTR		R/W	—	√	—	0CH Note
F0264H	Timer RD mode register	TRDMR		R/W	√	√	—	00H Note
F0265H	Timer RD PWM function select register	TRDPMR		R/W	√	√	—	00H Note
F0266H	Timer RD function control register	TRDFCR		R/W	√	√	—	80H Note
F0267H	Timer RD output master enable register 1	TRDOER1		R/W	√	√	—	FFH Note
F0268H	Timer RD output master enable register 2	TRDOER2		R/W	√	√	—	00H Note
F0269H	Timer RD output control register	TRDOCR		R/W	√	√	—	00H Note
F026AH	Timer RD digital filter function select register 0	TRDDF0		R/W	√	√	—	00H Note
F026BH	Timer RD digital filter function select register 1	TRDDF1		R/W	√	√	—	00H Note

Note The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 3 - 12 Extended Special Function Register (2nd SFR) List (5/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F026CH	6-phase PWM option mode register	OPMR	R/W	—	√	—	00H
F026DH	6-phase PWM option status register	OPSR	R	—	√	—	00H
F026EH	6-phase PWM option Hi-Z start trigger register	OPHS	W	—	√	—	00H
F026FH	6-phase PWM option Hi-Z stop trigger register	OPHT	W	—	√	—	00H
F0270H	Timer RD control register 0	TRDCR0	R/W	√	√	—	00H Note
F0271H	Timer RD I/O control register A0	TRDIORA0	R/W	√	√	—	00H Note
F0272H	Timer RD I/O control register C0	TRDIORC0	R/W	√	√	—	88H Note
F0273H	Timer RD status register 0	TRDSR0	R/W	√	√	—	00H Note
F0274H	Timer RD interrupt enable register 0	TRDIER0	R/W	√	√	—	00H Note
F0275H	Timer RD PWM function output level control register 0	TRDPOCR0	R/W	√	√	—	00H Note
F0276H	Timer RD counter 0	TRD0	R/W	—	—	√	0000H Note
F0277H							
F0278H	Timer RD general register A0	TRDGRA0	R/W	—	—	√	FFFFH Note
F0279H							
F027AH	Timer RD general register B0	TRDGRB0	R/W	—	—	√	FFFFH Note
F027BH							
F0280H	Timer RD control register 1	TRDCR1	R/W	√	√	—	00H Note
F0281H	Timer RD I/O control register A1	TRDIORA1	R/W	√	√	—	00H Note
F0282H	Timer RD I/O control register C1	TRDIORC1	R/W	√	√	—	88H Note
F0283H	Timer RD status register 1	TRDSR1	R/W	√	√	—	00H Note
F0284H	Timer RD interrupt enable register 1	TRDIER1	R/W	√	√	—	00H Note
F0285H	Timer RD PWM function output level control register 1	TRDPOCR1	R/W	√	√	—	00H Note
F0286H	Timer RD counter 1	TRD1	R/W	—	—	√	0000H Note
F0287H							
F0288H	Timer RD general register A1	TRDGRA1	R/W	—	—	√	FFFFH Note
F0289H							
F028AH	Timer RD general register B1	TRDGRB1	R/W	—	—	√	FFFFH Note
F028BH							
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	—	00H

Note The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 3 - 13 Extended Special Function Register (2nd SFR) List (6/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	—	—	√	0000H
F02FAH	CRC data register	CRCD	R/W	—	—	√	0000H
F0300H	Event output destination select register 00	ELSELR00	R/W	—	√	—	00H
F0301H	Event output destination select register 01	ELSELR01	R/W	—	√	—	00H
F0302H	Event output destination select register 02	ELSELR02	R/W	—	√	—	00H
F0303H	Event output destination select register 03	ELSELR03	R/W	—	√	—	00H
F0304H	Event output destination select register 04	ELSELR04	R/W	—	√	—	00H
F0305H	Event output destination select register 05	ELSELR05	R/W	—	√	—	00H
F0306H	Event output destination select register 06	ELSELR06	R/W	—	√	—	00H
F0307H	Event output destination select register 07	ELSELR07	R/W	—	√	—	00H
F0308H	Event output destination select register 08	ELSELR08	R/W	—	√	—	00H
F0309H	Event output destination select register 09	ELSELR09	R/W	—	√	—	00H
F030AH	Event output destination select register 10	ELSELR10	R/W	—	√	—	00H
F030BH	Event output destination select register 11	ELSELR11	R/W	—	√	—	00H
F030CH	Event output destination select register 12	ELSELR12	R/W	—	√	—	00H
F030DH	Event output destination select register 13	ELSELR13	R/W	—	√	—	00H
F0310H	Event output destination select register 16	ELSELR16	R/W	—	√	—	00H
F0311H	Event output destination select register 17	ELSELR17	R/W	—	√	—	00H
F0312H	Event output destination select register 18	ELSELR18	R/W	—	√	—	00H
F0340H	Comparator mode setting register	COMPMDR	R/W	√	√	—	00H
F0341H	Comparator filter control register	COMPFIR	R/W	—	√	—	00H
F0342H	Comparator output control register	COMPOCR	R/W	√	√	—	00H
F0343H	Comparator internal reference voltage control register	CVRCTL	R/W	√	√	—	00H
F0344H	Comparator internal reference voltage select register 0	C0RVM	R/W	—	√	—	00H
F0345H	Comparator internal reference voltage select register 1	C1RVM	R/W	—	√	—	00H
F0346H	PGA control register	PGACTL	R/W	√	√	—	00H
F0500H	Timer RJ counter register 0	TRJ0	R/W	—	—	√	FFFFH
F0501H							

Remark For SFRs in the SFR area, see **Tables 3 - 5** to **3 - 7 Special Function Register (SFR) List**.

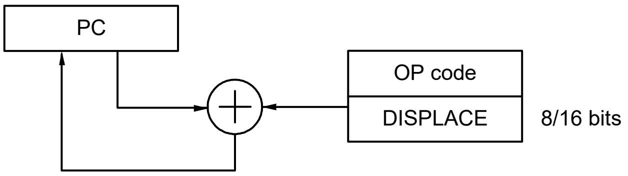
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: -128 to +127 or -32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3 - 13 Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3 - 14 Example of CALL !!addr20/BR !!addr20

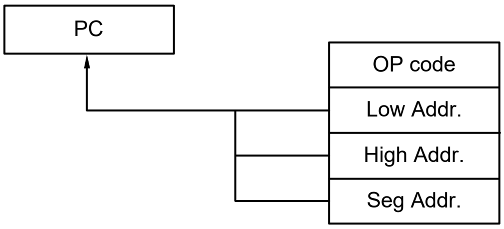
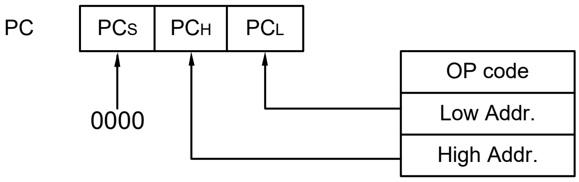


Figure 3 - 15 Example of CALL !addr16/BR !addr16



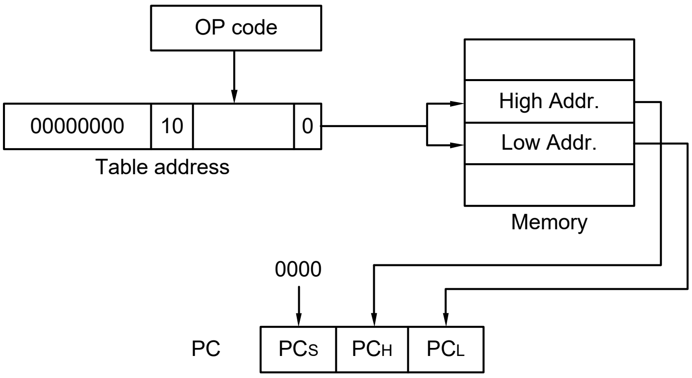
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3 - 16 Outline of Table Indirect Addressing

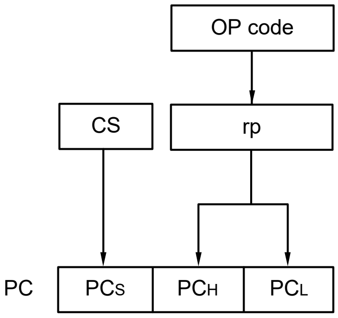


3.3.4 Register direct addressing

[Function]

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3 - 17 Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

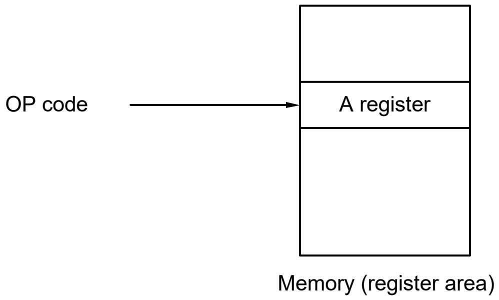
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

Implied addressing can be applied only to MULU X.

Figure 3 - 18 Outline of Implied Addressing



3.4.2 Register addressing

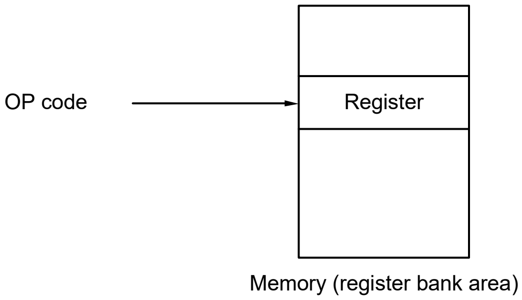
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3 - 19 Outline of Register Addressing



3.4.3 Direct addressing

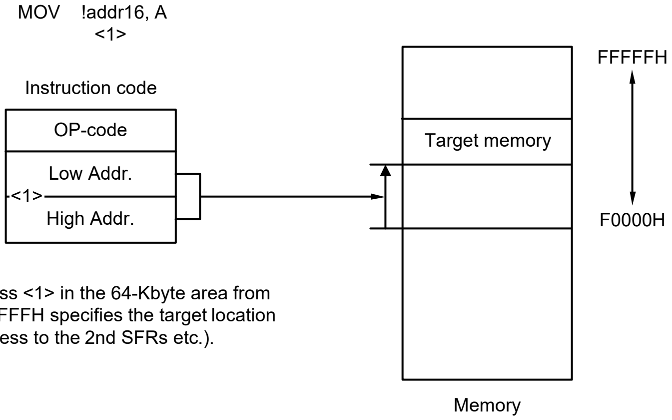
[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

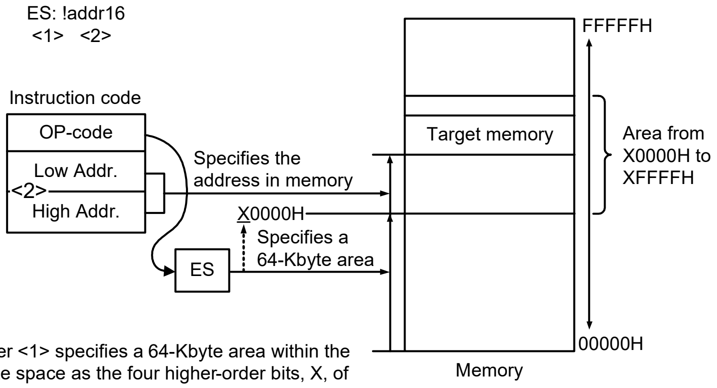
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 20 Example of !addr16



- A 16-bit address <1> in the 64-Kbyte area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

Figure 3 - 21 Example of ES:!addr16



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

3.4.4 Short direct addressing

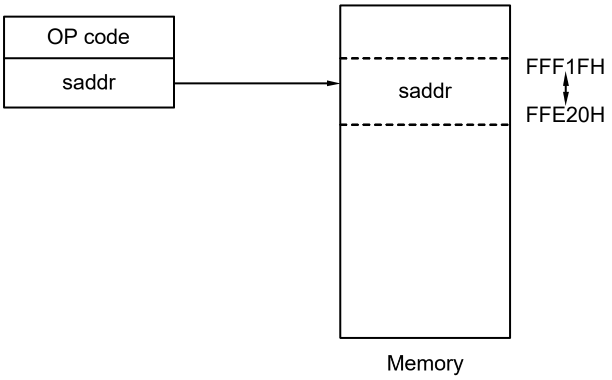
[Function]

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

[Operand format]

Identifier	Description
SADDR	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (only the space from FFE20H to FFF1FH is specifiable)
SADDRP	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3 - 22 Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

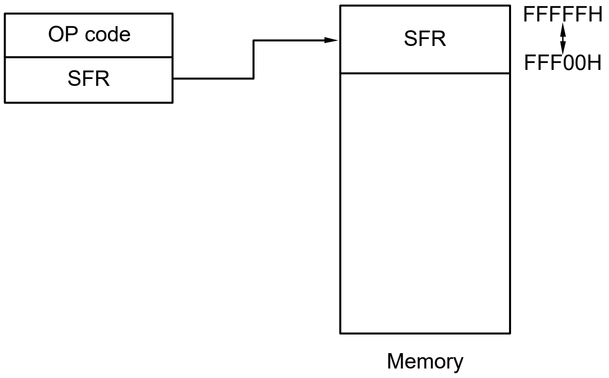
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address only)

Figure 3 - 23 Outline of SFR Addressing



3.4.6 Register indirect addressing

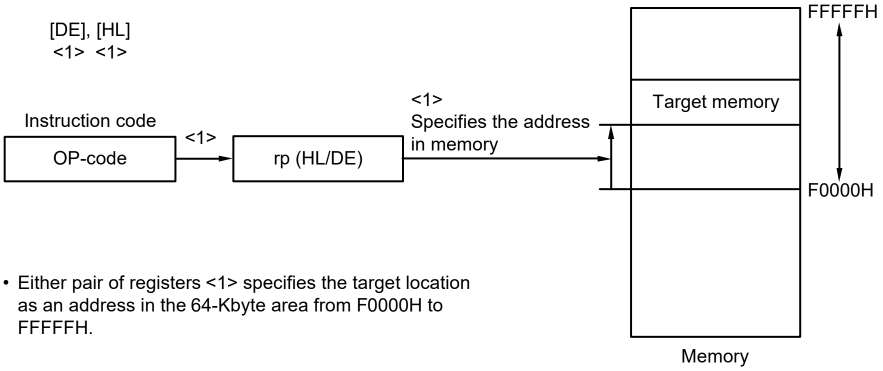
[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

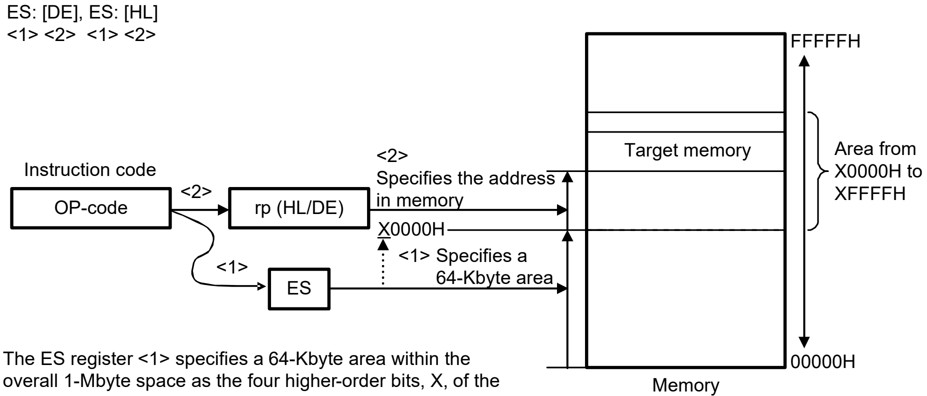
Identifier	Description
—	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 24 Example of [DE], [HL]



- Either pair of registers <1> specifies the target location as an address in the 64-Kbyte area from F0000H to FFFFFH.

Figure 3 - 25 Example of ES:[DE], ES:[HL]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

3.4.7 Based addressing

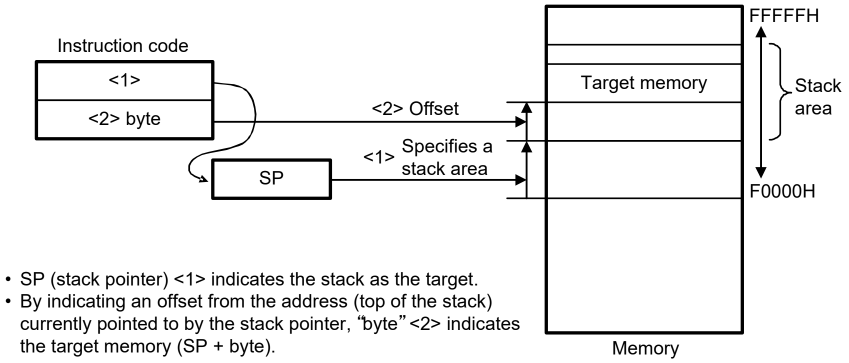
[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

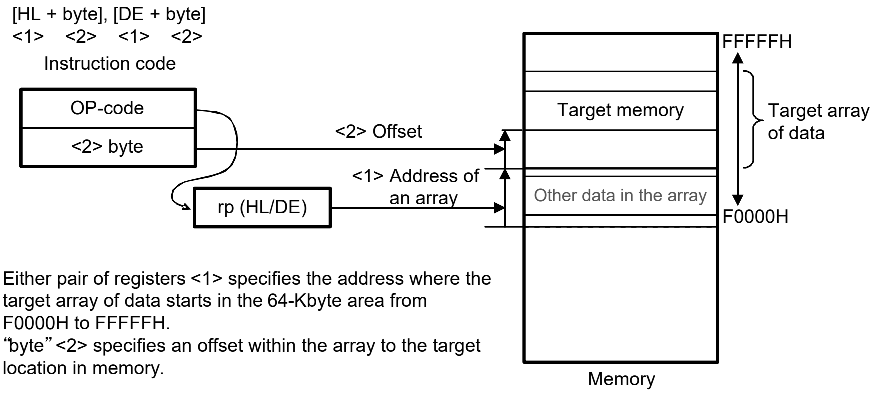
Identifier	Description
—	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
—	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
—	word[BC] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register)
—	ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register)
—	ES:word[BC] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 26 Example of [SP + byte]



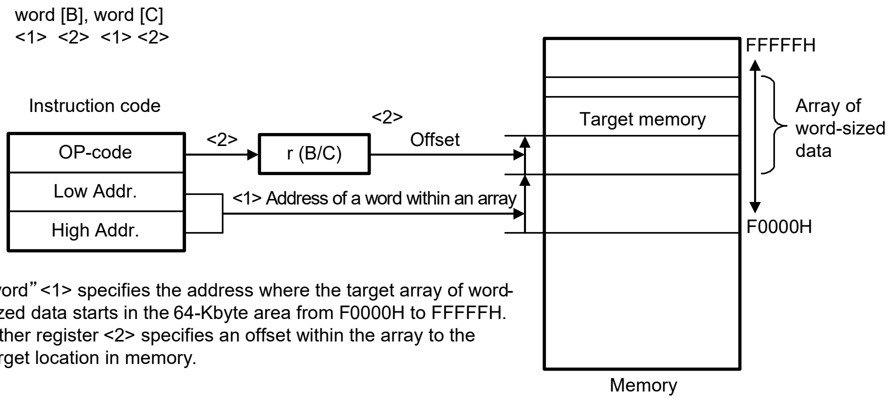
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, "byte" <2> indicates the target memory (SP + byte).

Figure 3 - 27 Example of [HL + byte], [DE + byte]



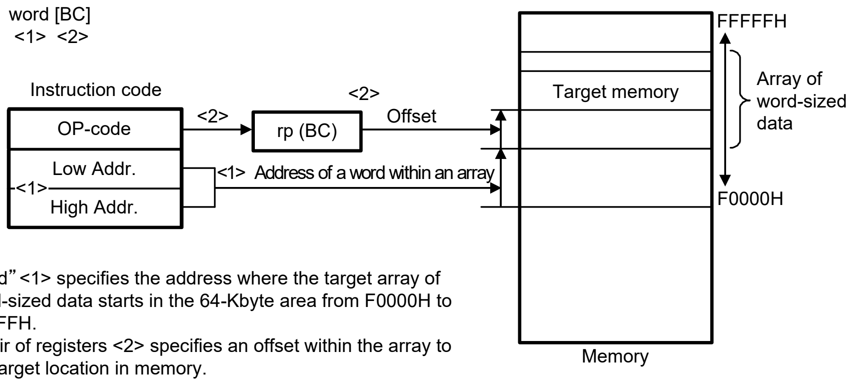
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- “byte” <2> specifies an offset within the array to the target location in memory.

Figure 3 - 28 Example of word [B], word [C]



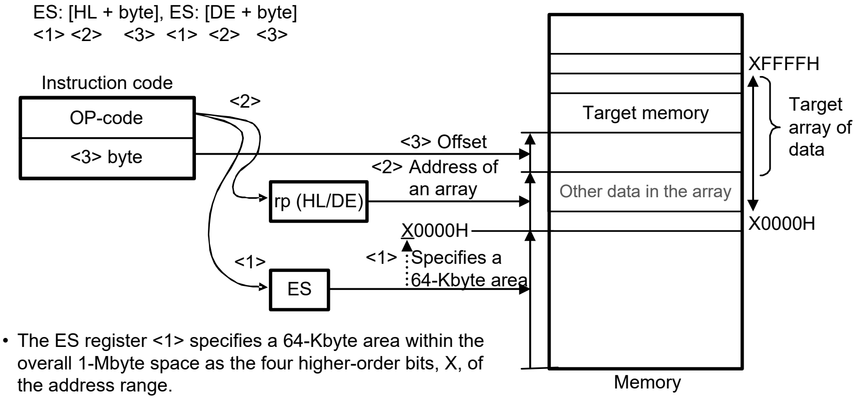
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 29 Example of word [BC]



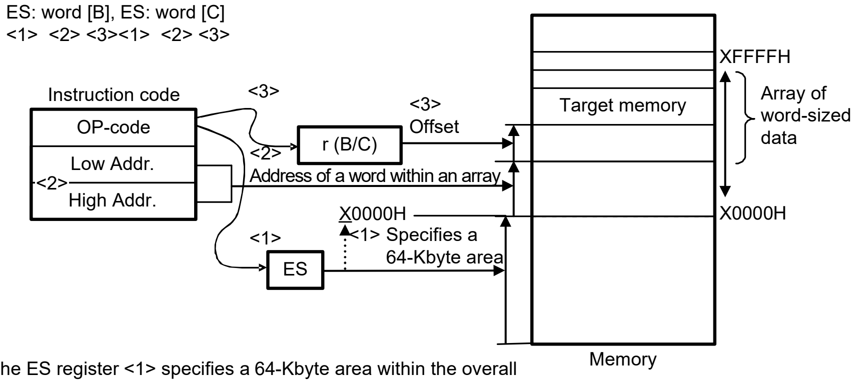
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3 - 30 Example of ES:[HL + byte], ES:[DE + byte]



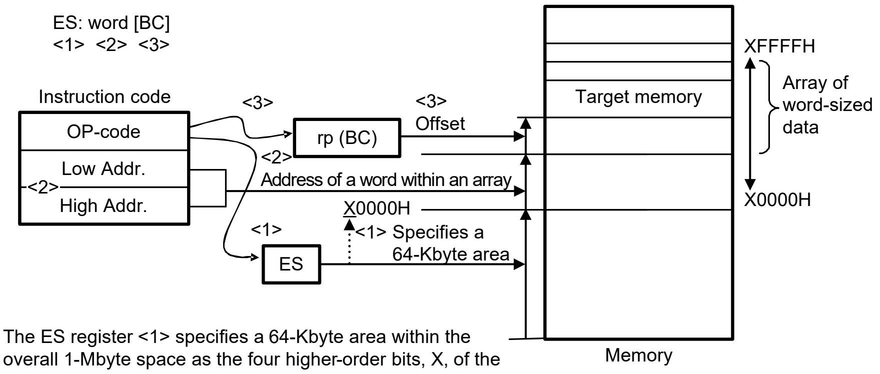
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies an offset within the array to the target location in memory.

Figure 3 - 31 Example of ES:word[B], ES:word[C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3 - 32 Example of ES:word[BC]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

3.4.8 Based indexed addressing

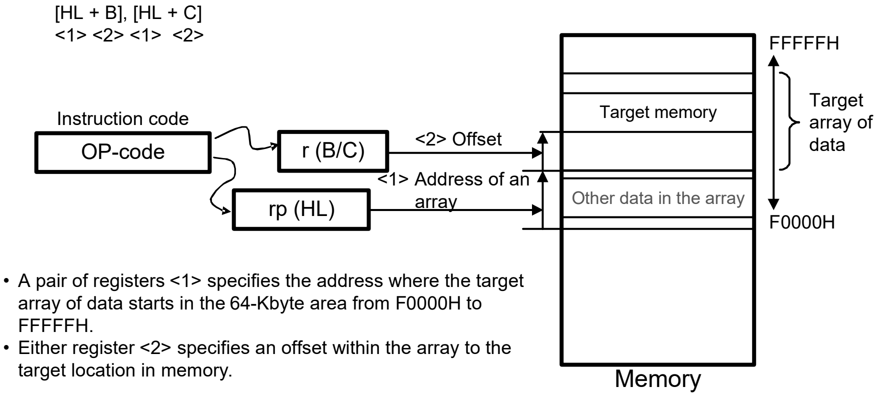
[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

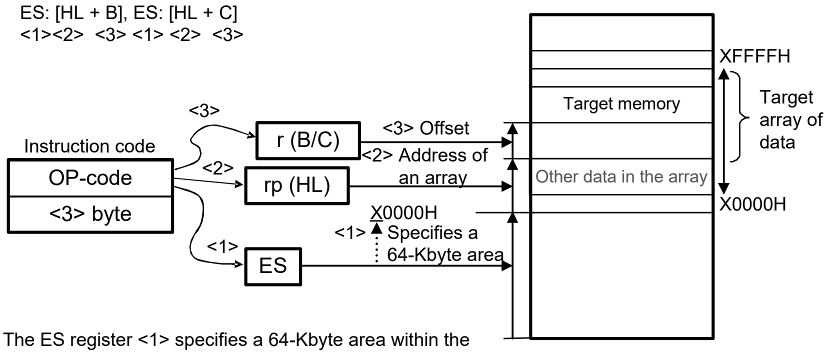
Identifier	Description
—	[HL + B], [HL + C] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + B], ES:[HL + C] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 33 Example of [HL + B], [HL + C]



- A pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 34 Example of ES:[HL + B], ES:[HL + C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) values. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

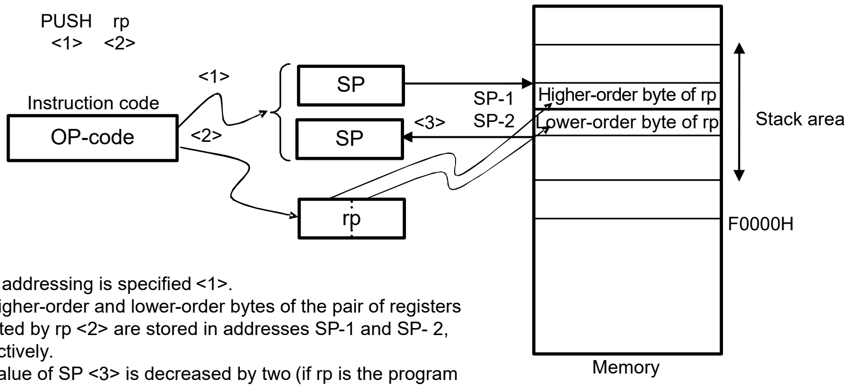
Only the internal RAM area can be set as the stack area.

[Operand format]

Identifier	Description
—	PUSH AX/BC/DE/HL POP AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

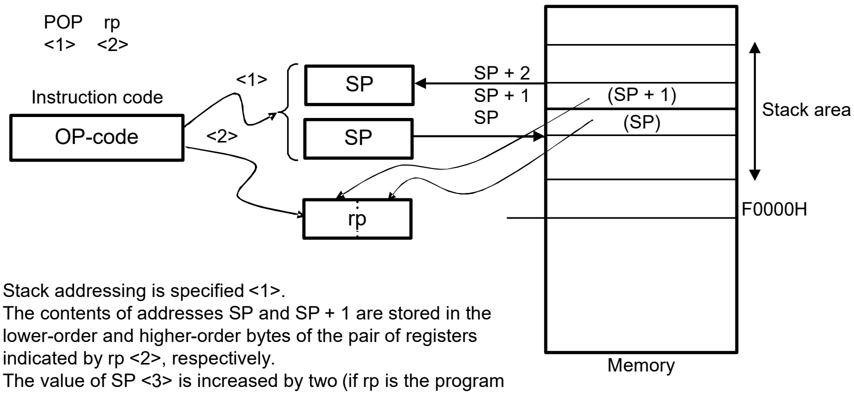
The data to be saved/restored by each stack operation is shown in Figures 3 - 35 to 3 - 40.

Figure 3 - 35 Example of PUSH rp



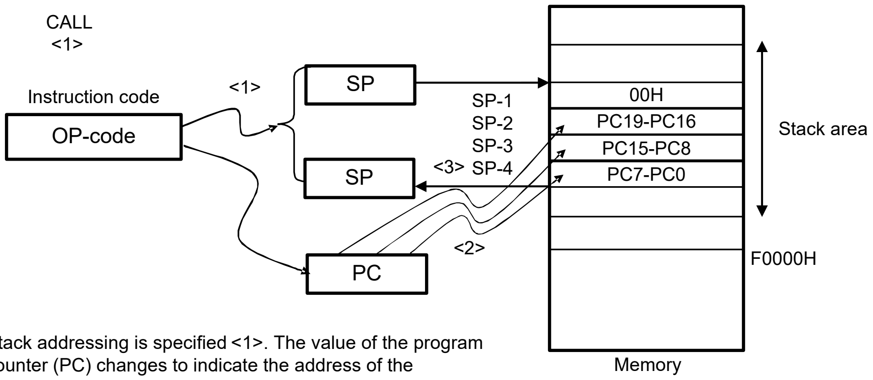
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP-1 and SP- 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP-1 and 0 is stored in SP- 2).

Figure 3 - 36 Example of POP



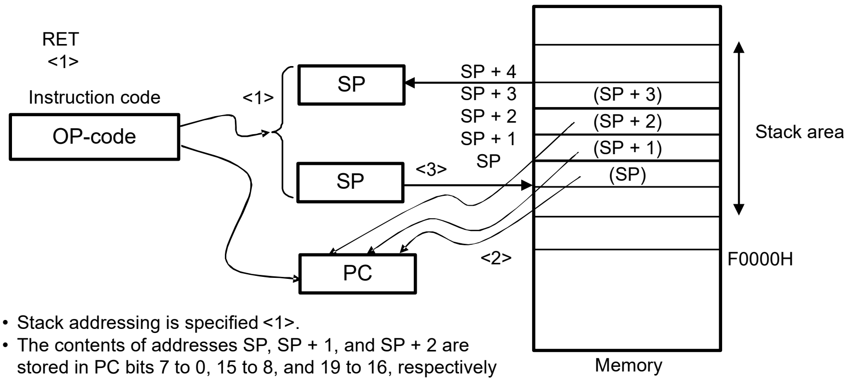
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3 - 37 Example of CALL, CALLT



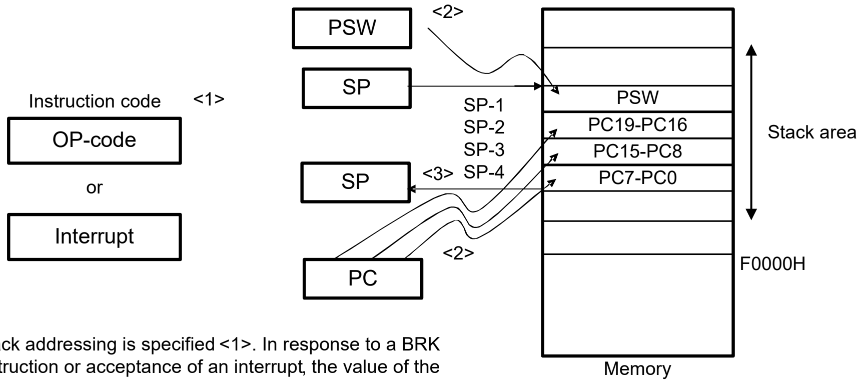
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3 - 38 Example of RET



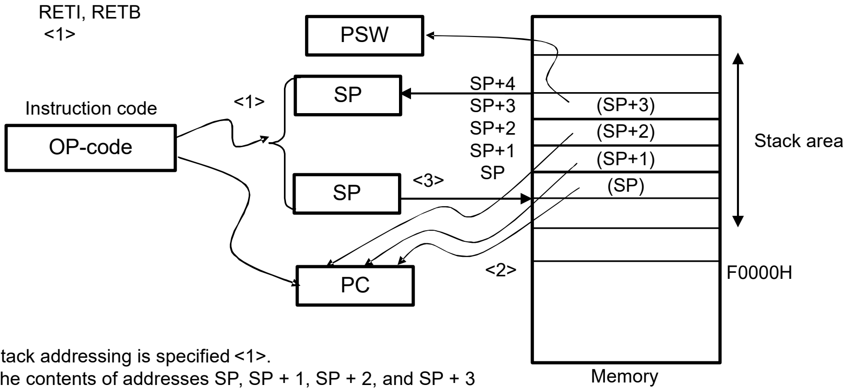
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 39 Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3 - 40 Example of RETI, RETB



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/G1G microcontrollers are provided with digital I/O ports, which enable variety of control operations.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4 - 1 Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0-PM7, PM12, PM14) Port registers (P0-P7, P12-P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU12, PU14) Port input mode registers (PIM0, PIM1, PIM3, PIM5) Port output mode registers (POM0, POM1, POM3, POM5) Port mode control registers (PMC0, PMC12, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR1)
Port	• 30-pin products Total: 26 (CMOS I/O: 23, CMOS input: 3) • 32-pin products Total: 28 (CMOS I/O: 25, CMOS input: 3) • 44-pin products Total: 40 (CMOS I/O: 35, CMOS input: 5)
Pull-up resistor	• 30-pin products Total: 19 • 32-pin products Total: 21 • 44-pin products Total: 27

Caution **Most of the following descriptions in this chapter use the 44-pin products with the 00H setting in peripheral I/O redirection register 1 (PIOR1) as an example.**

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P01 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P00 pin can be specified as N-ch open-drain output (V$_{DD}$ tolerance) in 1-bit units using port output mode register 0 (POM0).

Input to the P00 and P01 pins can be specified as analog input or digital input in 1-bit units, using port mode control register 0 (PMC0).

This port can also be used for timer I/O, A/D converter analog input, serial interface data I/O, clock I/O, comparator input, and PGA input.

When reset signal is generated, the following configuration will be set.

- Analog input

Table 4 - 2 Settings of Registers When Using Port 0

Pin Name		PM0x	PIM0x	POM0x	PMC0x	Alternate Function Setting $^{\text{Note 4}}$	Remark
Name	I/O						
P00	Input	1	—	×	0	×	
	Output	0		0	0	TxD1 output = 1 $^{\text{Note 1}}$ (TRJO0 = 0)	CMOS output
		0		1	0		N-ch O.D. output
P01	Input	1	0	—	0	×	CMOS input
		1	1		0	×	TTL input
	Output	0	×		0	TO00 output = 0 $^{\text{Note 2}}$ TRJIO0 output = 1 $^{\text{Note 3}}$	

Note 1. To use a pin multiplexed with the serial array unit function as a general-purpose port, set the SOmn bit in serial output register m (SOm), the SOEmn bit in serial output enable register m (SOEm), and the SEmn bit in serial channel enable status register m (SEm) for the corresponding unit channel to the default value. (mn = 02)

Note 2. To use a pin multiplexed with the timer output function of the timer array unit as a general-purpose port, set the TOmn bit in timer output register m (TOm) and the TOEmn bit in timer output enable register m (TOEm) for the corresponding unit channel to the default status. (m = 0, n = 0)

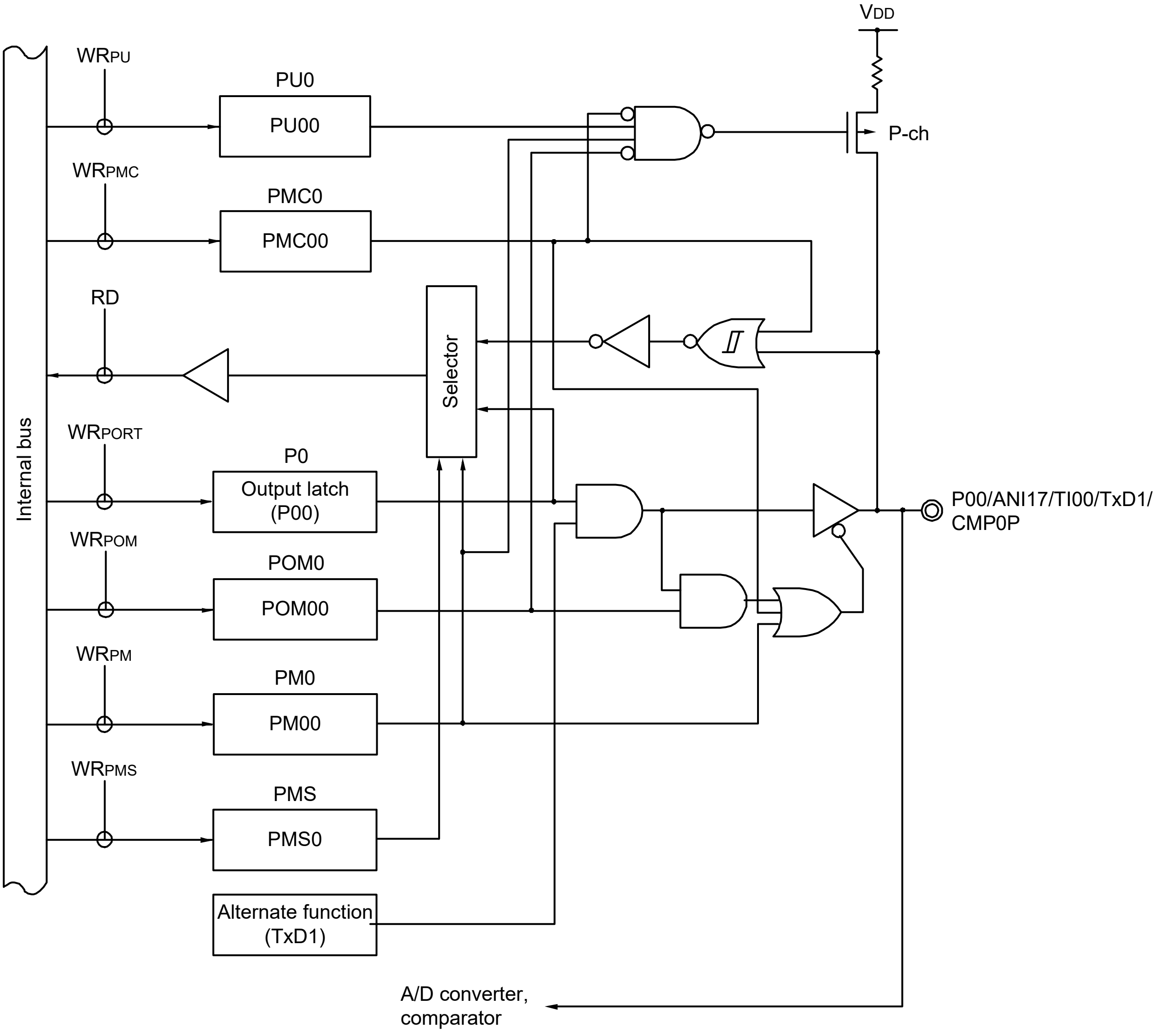
Note 3. To use a pin multiplexed with the timer I/O function of timer RJ as a general-purpose port, set bits TMOD2 to TMOD0 in timer RJ mode register 0 (TRJMR0) to the default value or a value other than 001B.

Note 4. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

Remark ×: don't care
PM0x: Port mode register 0
PIM0x: Port input mode register 0
POM0x: Port output mode register 0
PMC0x: Port mode control register 0

For example, Figures 4 - 1 and 4 - 2 show block diagrams of port 0 for 44-pin products when PIOR1 = 00H.

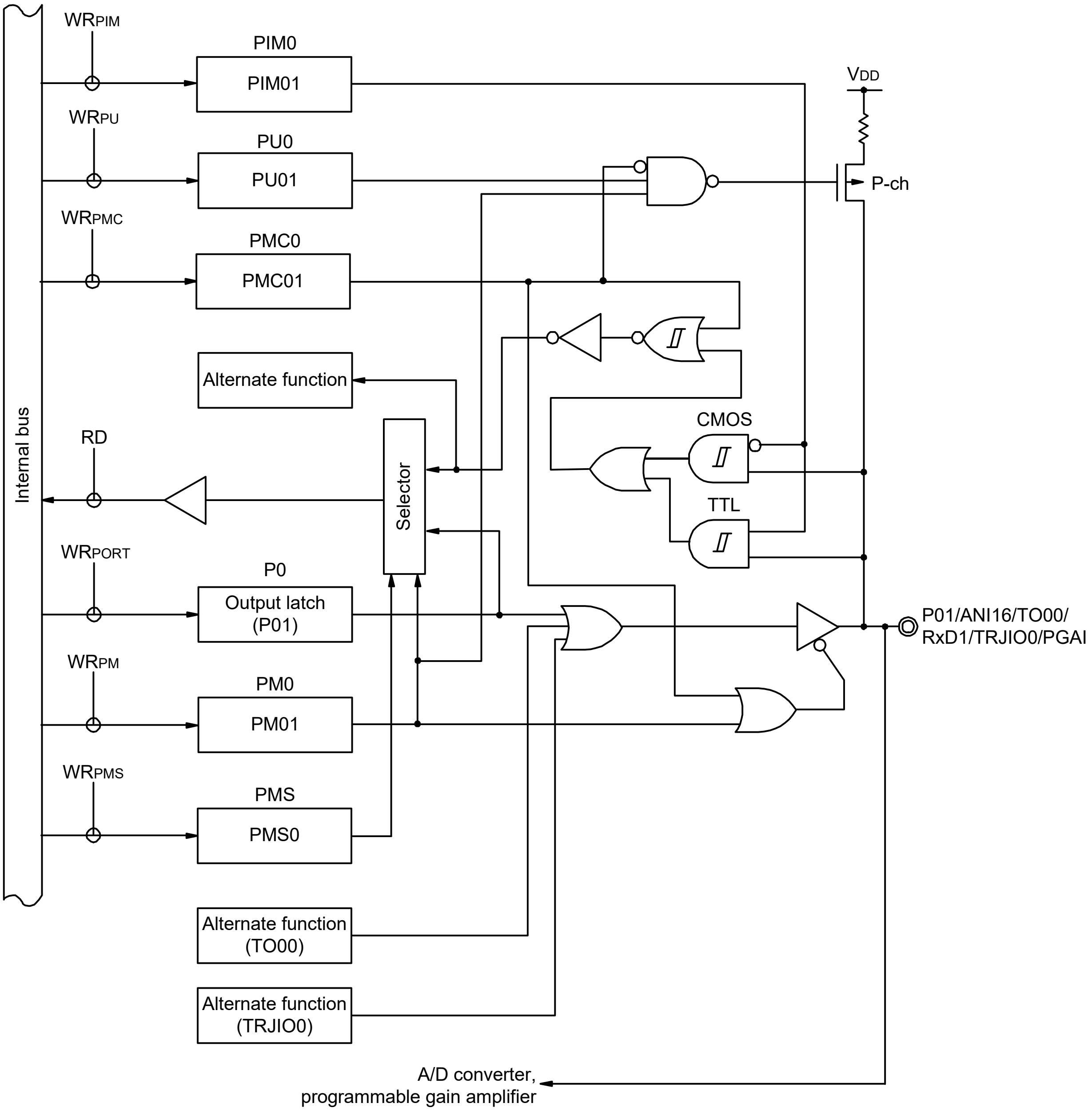
Figure 4 - 1 Block Diagram of P00



Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

P0:	Port register 0
PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
POM0:	Port output mode register 0
PMC0:	Port mode control register 0
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 2 Block Diagram of P01



Caution	**Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.** **It is recommended to input a low level to prevent increase of the electric power.**

P0:	Port register 0
PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
PIM0:	Port input mode register 0
PMC0:	Port mode control register 0
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10 and P15 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P10, P15, and P17 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

This port can also be used for clock I/O, timer I/O, external interrupt request input, clock/buzzer output.

Reset signal generation sets port 1 to input mode.

Table 4 - 3 Settings of Registers When Using Port 1 (1/2)

Pin Name		PM1x	PIM1x	POM1x	Alternate Function Setting	Remark
Name	I/O					
P10	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	TRDIOD1 output = 0 Note 1	CMOS output
		0	×	1		N-ch O.D. output
P11	Input	1	—	—	×	
	Output	0			TRDIOC1 output = 0 Note 1	CMOS output
P12	Input	1	—	—	×	
	Output	0			TRDIOB1 output = 0 Note 1	
P13	Input	1	—	—	×	
	Output	0			TRDIOA1 output = 0 Note 1	CMOS output
P14	Input	1	—	—	×	CMOS input
	Output	0			TRDIOD0 output = 0 Note 1	CMOS output
P15	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	PCLBUZ1 output = 0 Note 2 TRDIOB0 output = 0 Note 1	CMOS output
		0	×	1		N-ch O.D. output
P16	Input	1	0	—	×	CMOS input
		1	1		×	TTL input
	Output	0	×		TO01 output = 0 Note 3 TRDIOC0 output = 0 Note 1	

Table 4 - 4 Settings of Registers When Using Port 1 (2/2)

Pin Name		PM1x	PIM1x	POM1x	Alternate Function Setting	Remark
Name	I/O					
P17	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	TO02 output = 0 Note 3 TRDIOA0 output = 0 Note 1	CMOS output
		0	×	1		N-ch O.D. output

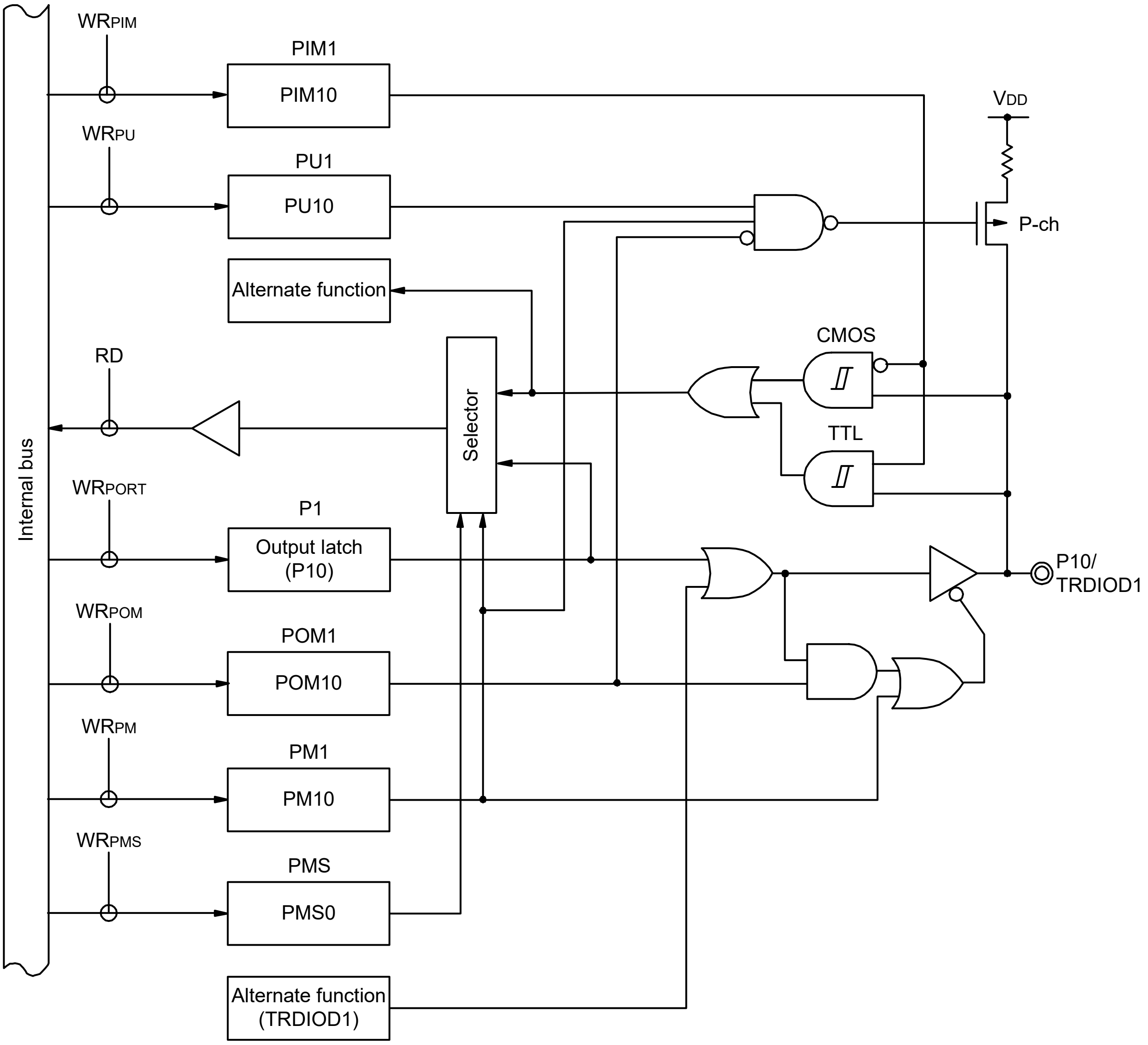
Note 1. To use a pin multiplexed with the timer RD function as a general-purpose port, set the output control bit in timer RD output master enable register 1 (TRDOER1) for the corresponding TRDIOij pin to the default value. (i = A, B, C, D; j = 0, 1)

Note 2. To use a pin multiplexed with the clock/buzzer output function as a general-purpose port, set the PCLOEi bit in clock output select register i (CKSi) to the default status. (i = 1)

Note 3. To use a pin multiplexed with the timer output function of the timer array unit as a general-purpose port, set the TOmn bit in timer output register m (TOm) and the TOEmn bit in timer output enable register m (TOEm) for the corresponding unit channel to the default status. (m = 0, n = 1, 2)

Remark
×: don't care
PM1x: Port mode register 1
PIM1x: Port input mode register 1
POM1x: Port output mode register 1

Figure 4 - 3 Block Diagram of P10

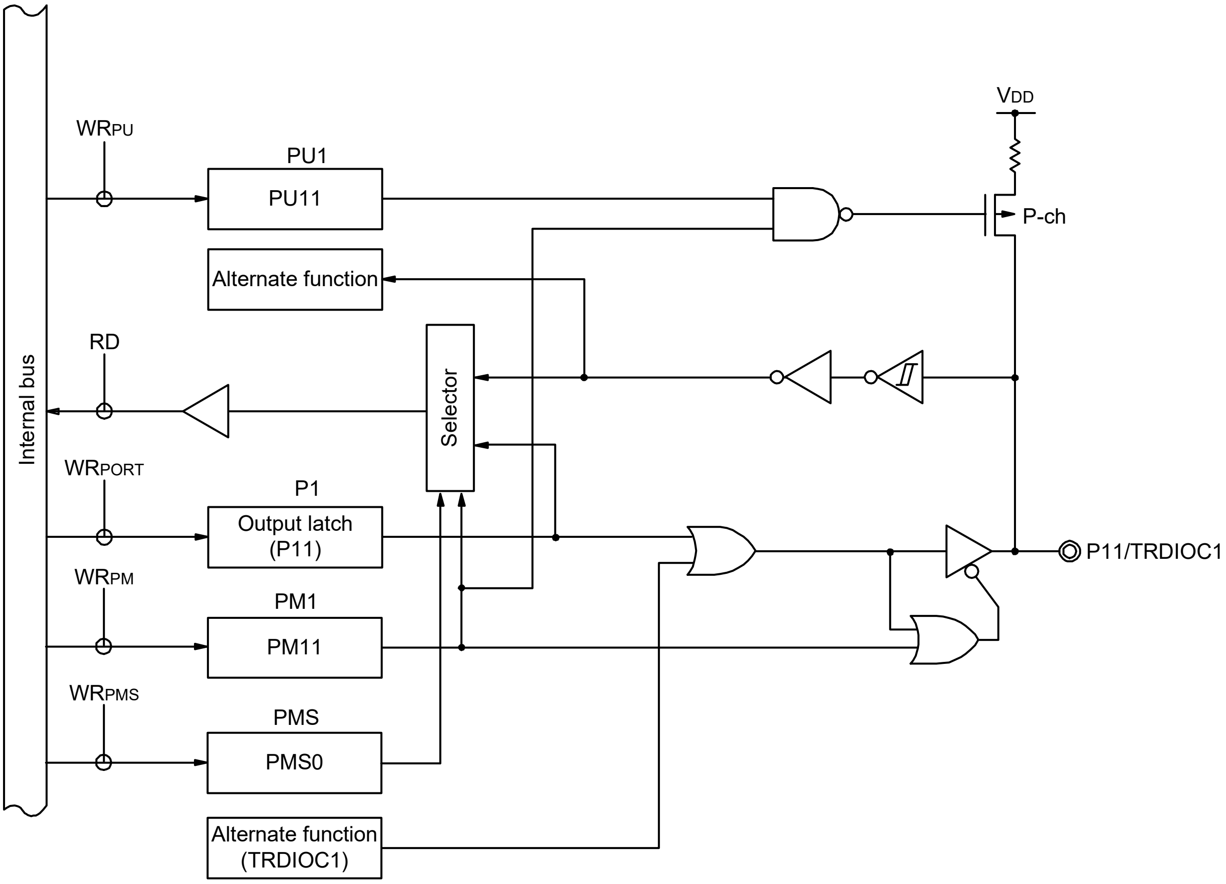


Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.
It is recommended to input a low level to prevent increase of the electric power.

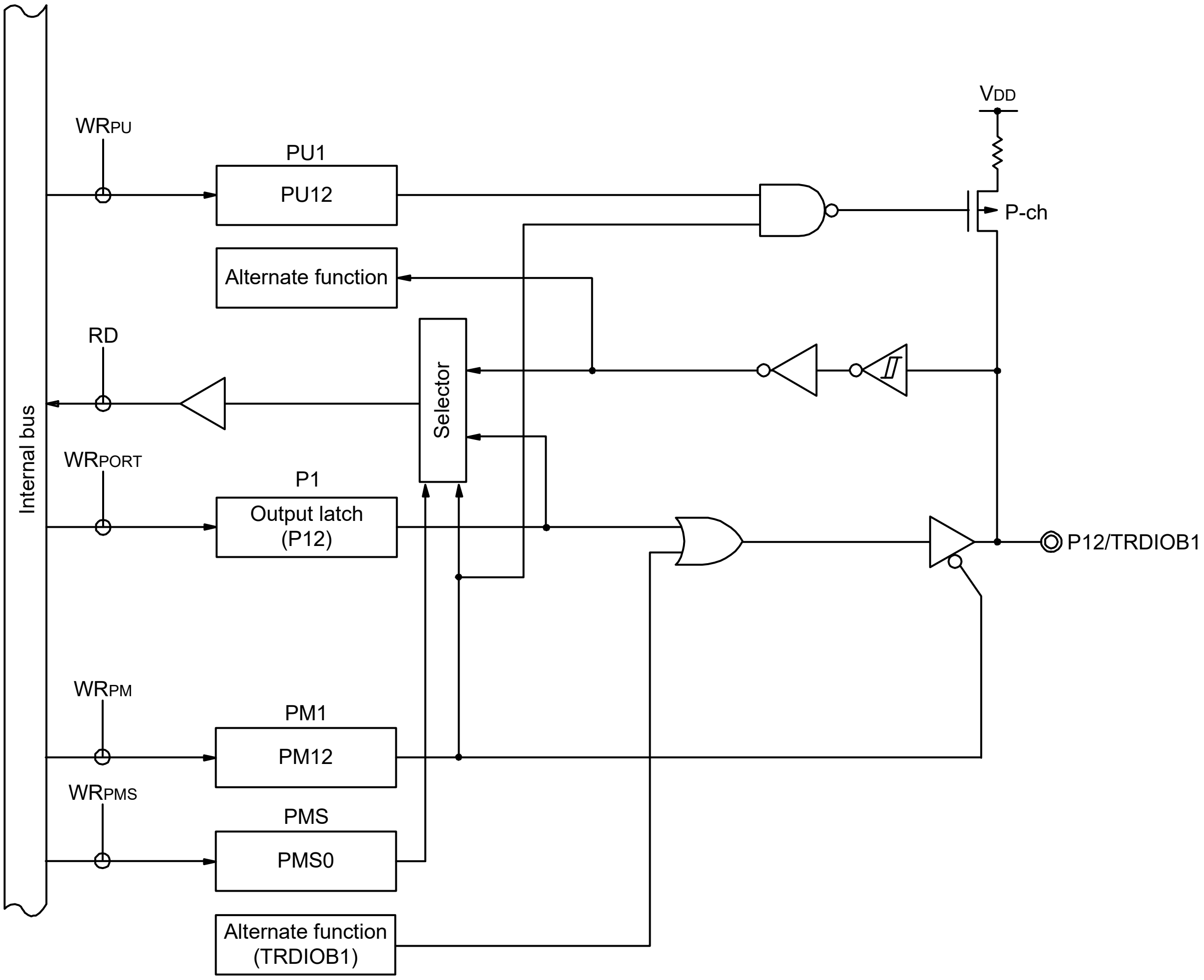
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PIM1:	Port input mode register 1
POM1:	Port output mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 4 Block Diagram of P11



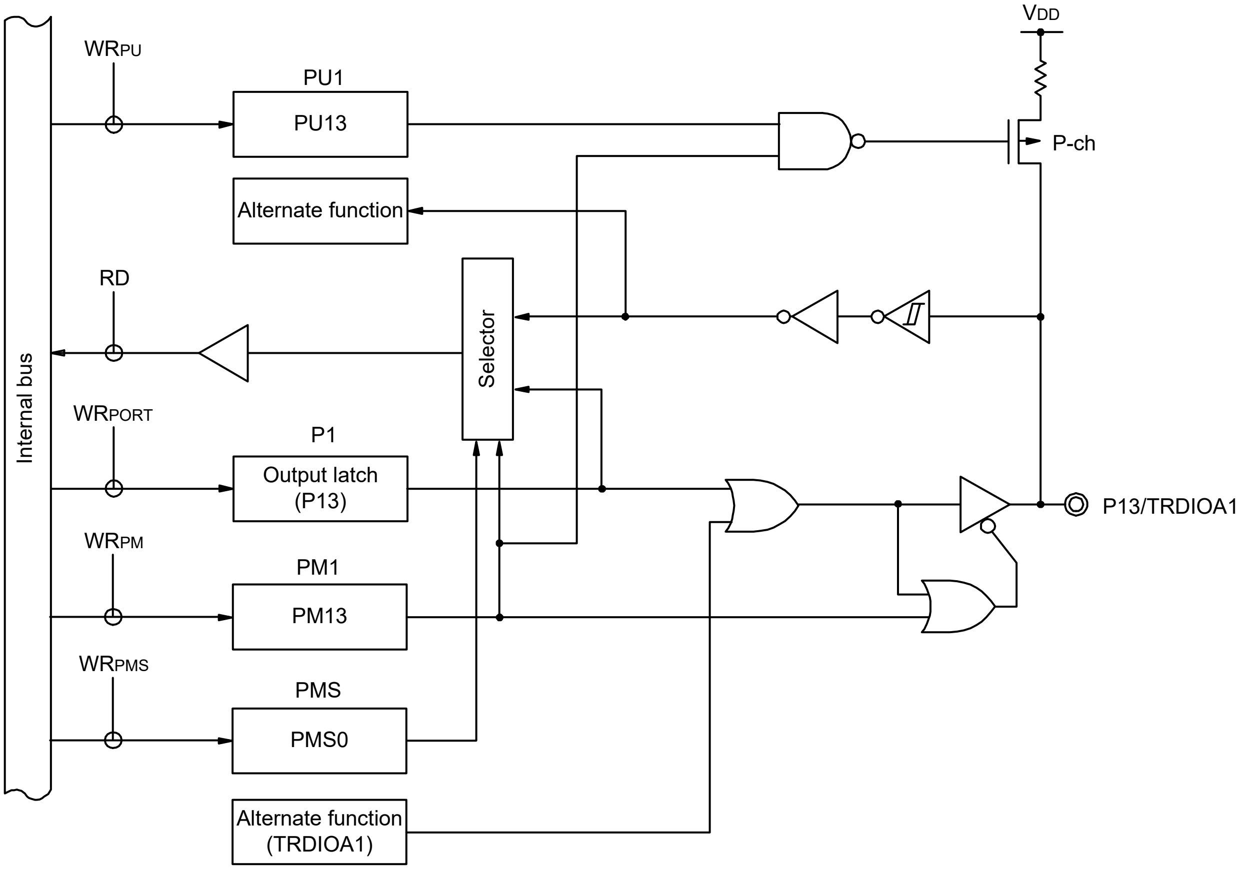
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 5 Block Diagram of P12



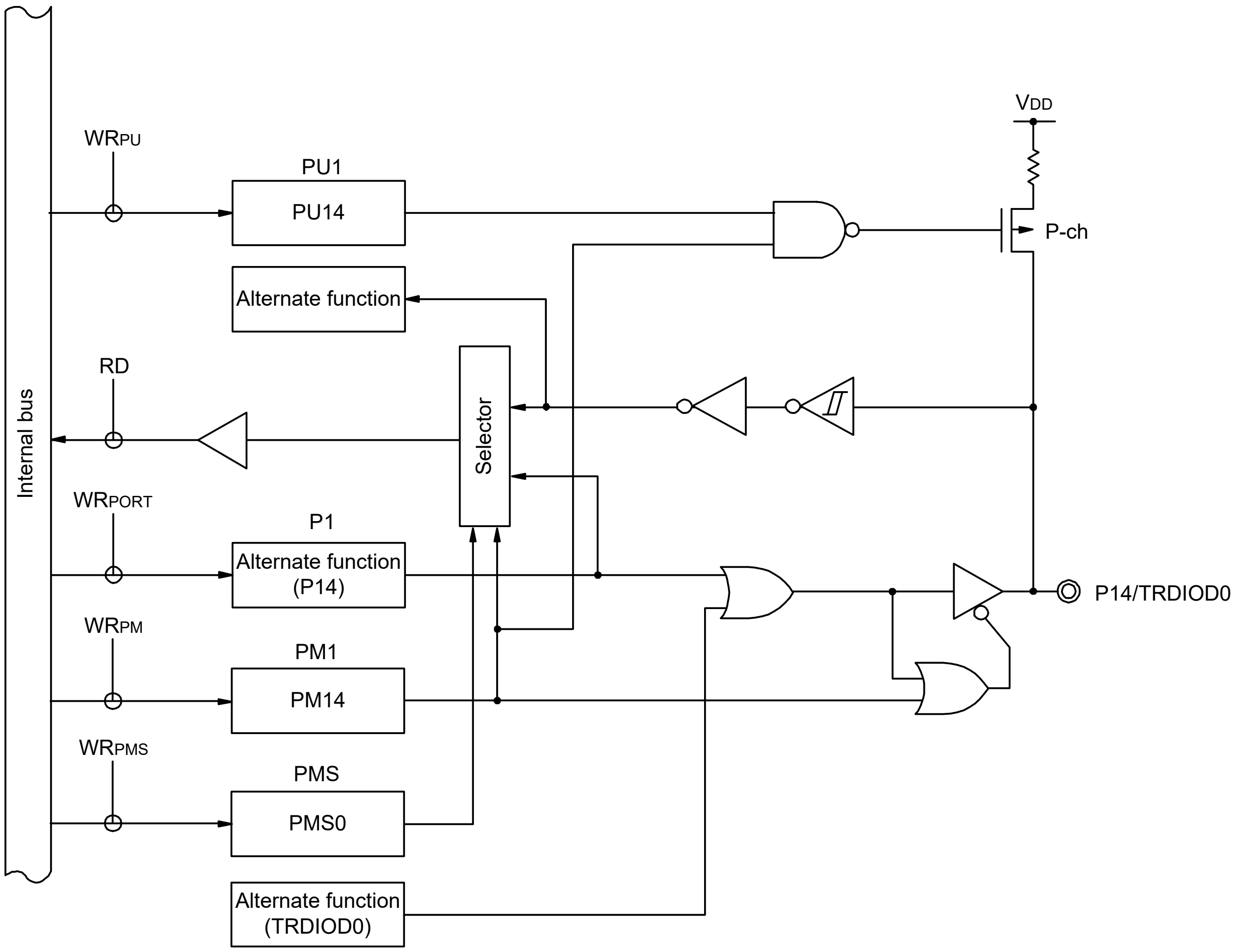
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 6 Block Diagram of P13



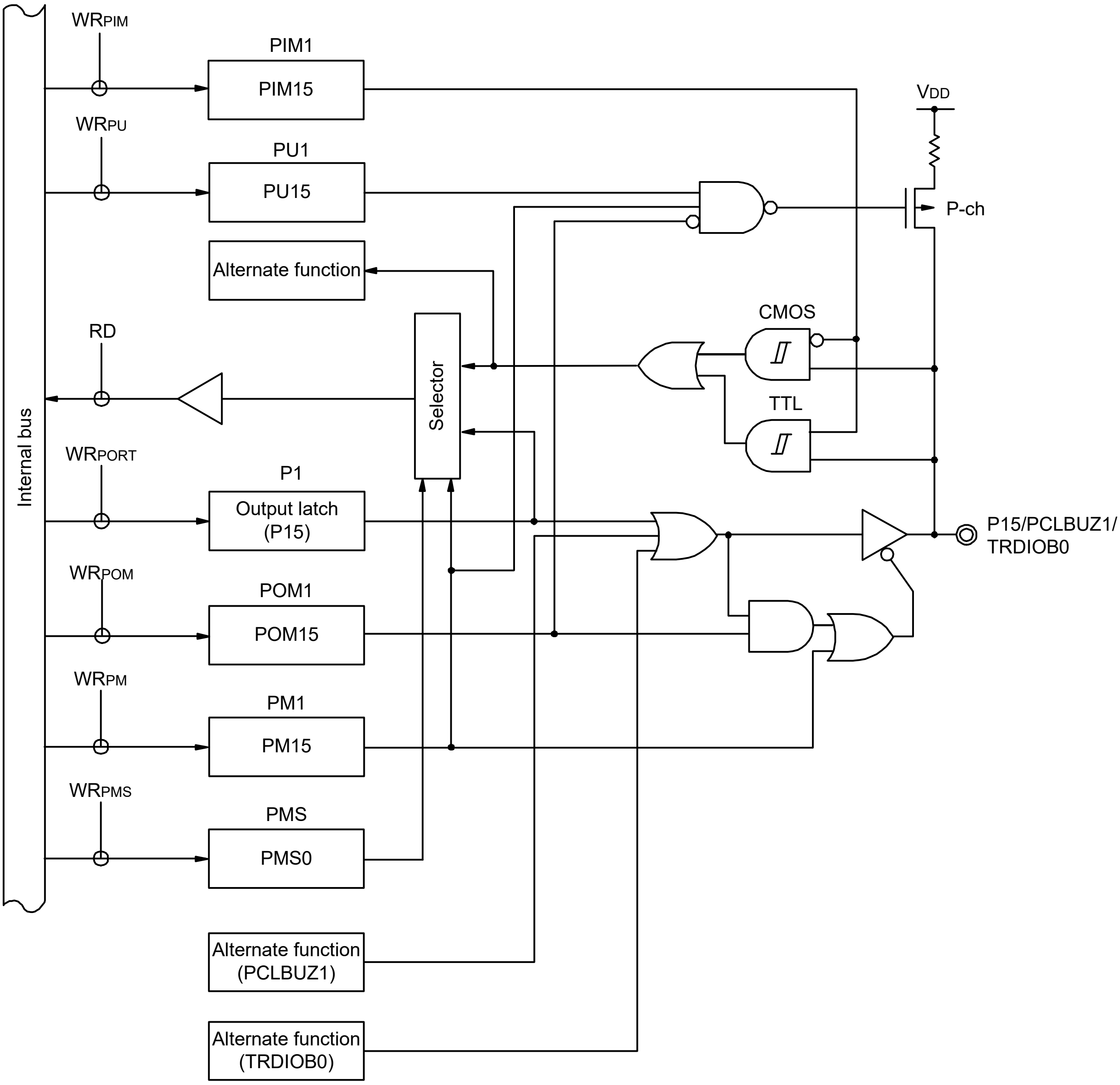
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 7 Block Diagram of P14



P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PIM1:	Port input mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 8 Block Diagram of P15



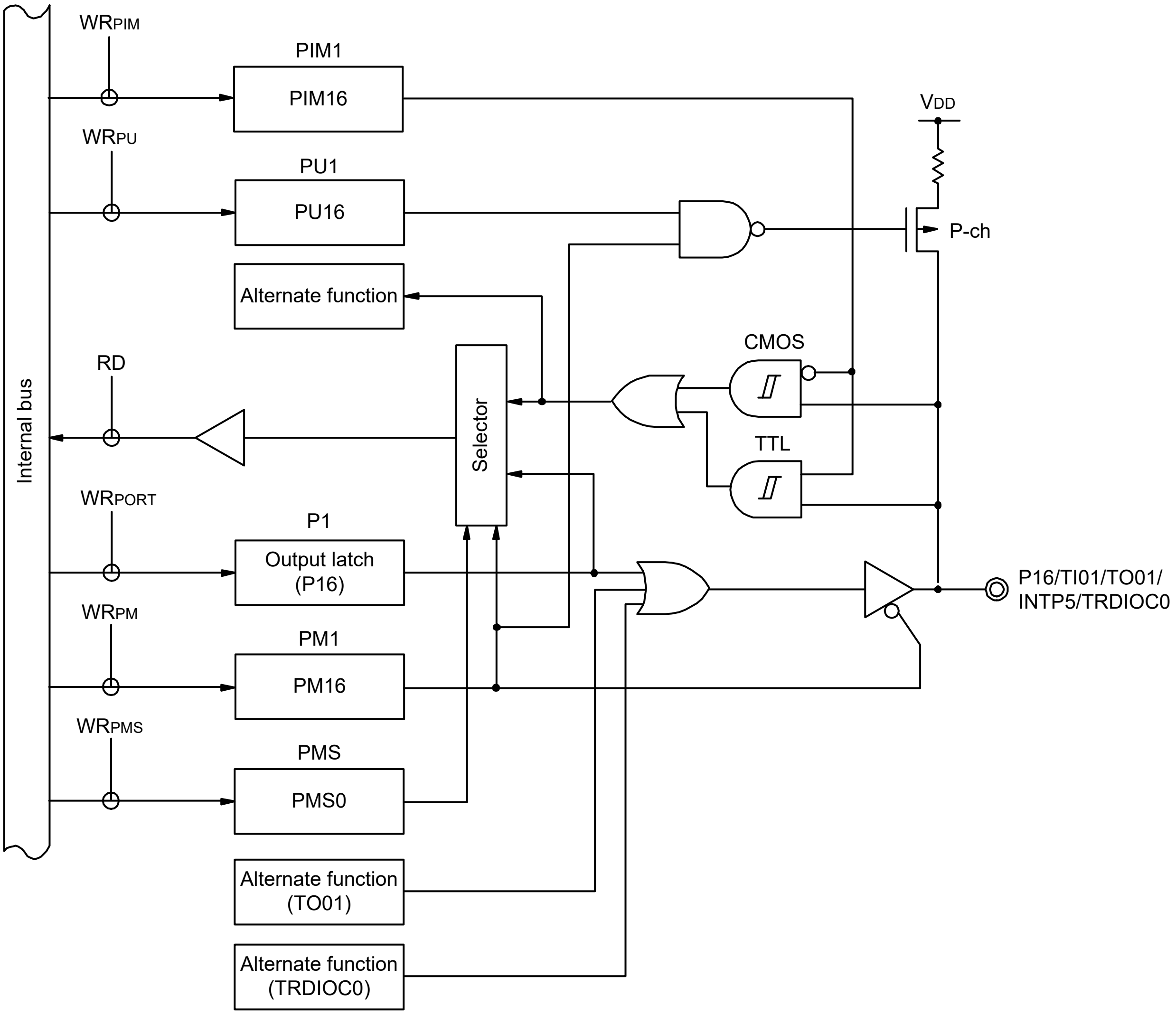
Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.

It is recommended to input a low level to prevent increase of the electric power.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PIM1:	Port input mode register 1
POM1:	Port output mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

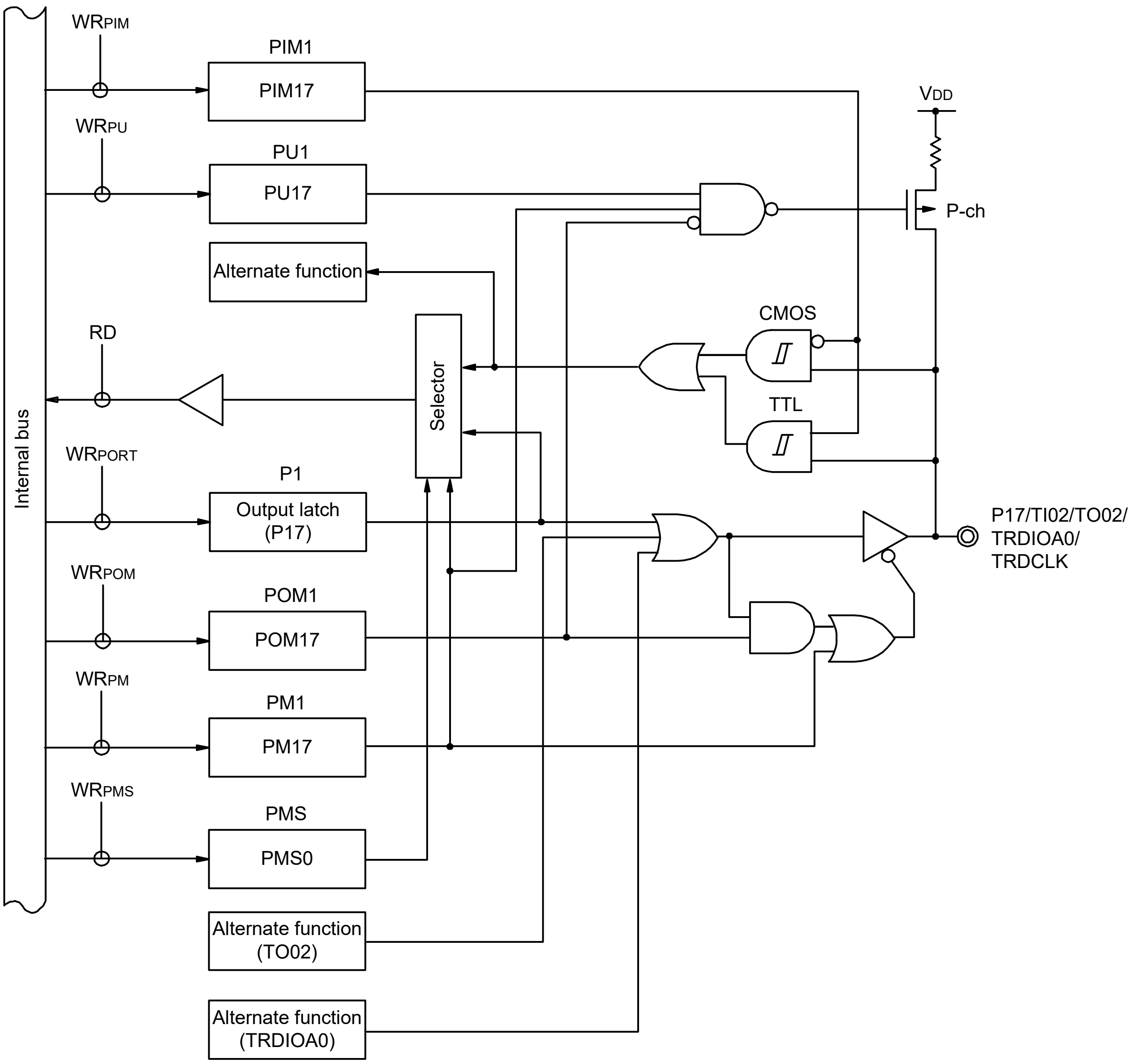
Figure 4 - 9 Block Diagram of P16



Caution **Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.**
It is recommended to input a low level to prevent increase of the electric power.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PIM1:	Port input mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 10 Block Diagram of P17



Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.

It is recommended to input a low level to prevent increase of the electric power.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
PIM1:	Port input mode register 1
POM1:	Port output mode register 1
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

This port can also be used for A/D converter analog input, and A/D converter and comparator (+side and - side) reference voltage input.

To use P20/ANI0/AVREFP, P21/ANI1/AVREFM, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the upper bit.

To use P20/ANI0/AVREFP, P21/ANI1/AVREFM, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM2 register.

To use P20/ANI0/AVREFP, P21/ANI1/AVREFM, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 as analog I/O pins, set them in the analog I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4 - 5 Settings of Registers When Using Port 2

Pin Name		PM2x	ADPC	Alternate Function Setting	Remark
Name	I/O				
P2n	Input	1	01 to n + 1H	—	To use P2n as a port, use these pins from a higher bit.
	Output	0	01 to n + 1H		

Remark 1. PM2x: Port mode register 2
ADPC: A/D port configuration register

Remark 2. n = 0 to 7

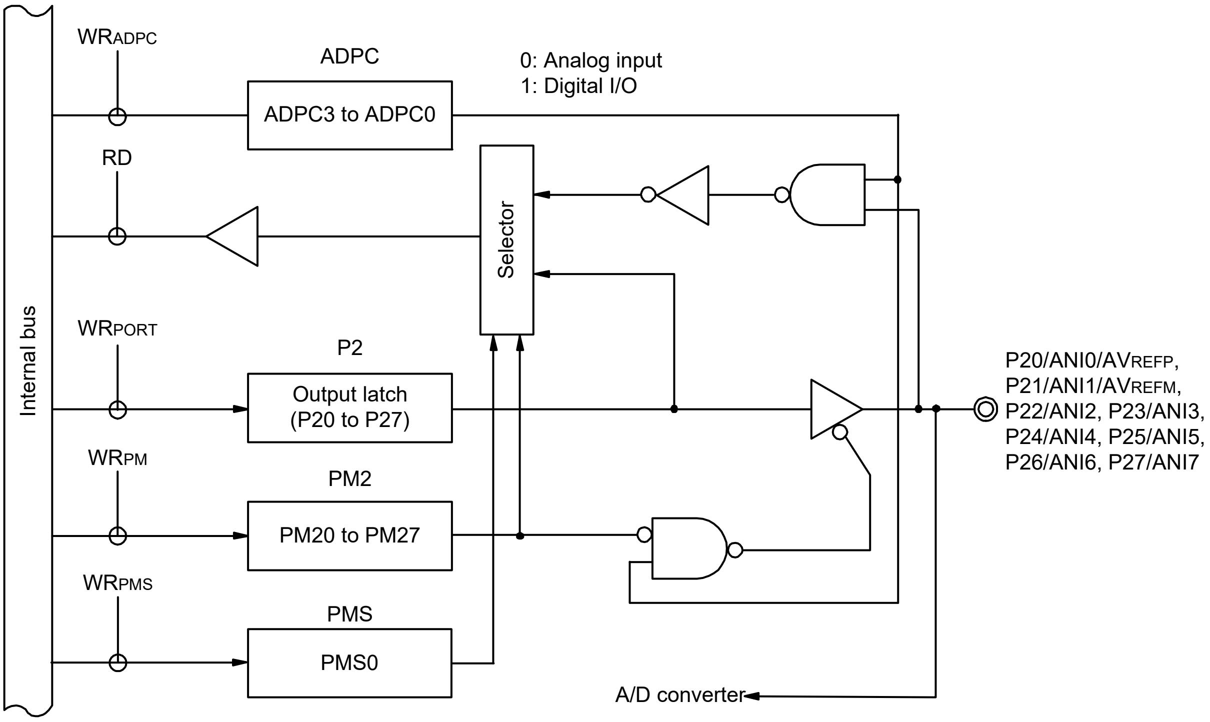
Table 4 - 6 Setting Functions of P20/ANI0 to P27/ANI7 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P27/ANI7 are set in the analog input mode when the reset signal is generated.

For example, Figure 4 - 11 shows a block diagram of port 2 for 44-pin products.

Figure 4 - 11 Block Diagram of P20, P21, P22, P23, P24, P25, P26, P27



P2:	Port register 2
PM2:	Port mode register 2
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 and P31 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

Input to the P30 and P31 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 3 (PIM3).

Output from the P30 pin can be specified as N-ch open-drain output (VDD tolerance) in 1-bit units using port output mode register 3 (POM3).

This port can also be used for external interrupt request input, clock/buzzer output, serial interface clock I/O, and timer I/O.

Reset signal generation sets port 3 to input mode.

Figures 4 - 12 and 4 - 13 show block diagrams of port 3.

Table 4 - 7 Settings of Registers When Using Port 3

Pin Name		PM3x	PIM3x	POM3x	Alternate Function Setting Note 6	Remark
Name	I/O					
P30	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	SCK00/SCL00 output = 0 Note 1 TRJO0 output = 0 Note 2	CMOS output
		0	×	1		N-ch O.D. output
P31	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	—	—	TO03 output = 0 Note 3 PCLBUZ0 output = 0 Note 4 (TRJIO0 output = 0 Note 5)	

Note 1. To use a pin multiplexed with the serial array unit function as a general-purpose port, set the CKOmn bit in serial output register m (SOm), the SOEmn bit in serial output enable register m (SOEm), and the SEmn bit in serial channel enable status register m (SEm) for the corresponding unit channel to the default value. (mn = 00)

Note 2. To use a pin multiplexed with the output function of timer RJ as a general-purpose port, set bit 2 (TOENA) in timer RJ I/O control register 0 (TRJIOC0) to the default value.

Note 3. To use a pin multiplexed with the timer output function of the timer array unit as a general-purpose port, set the TOmn bit in timer output register m (TOm) and the TOEmn bit in timer output enable register m (TOEm) for the corresponding unit channel to the default status. (m = 0, n = 3)

Note 4. To use a pin multiplexed with the clock/buzzer output function as a general-purpose port, set the PCLOEi bit in clock output select register i (CKSi) to the default value. (i = 0)

Note 5. To use a pin multiplexed with the timer I/O function of timer RJ as a general-purpose port when bits PIOR11 and PIOR10 in peripheral I/O redirection register 1 (PIOR1) are 01B, set bits TMOD2 to TMOD0 in timer RJ mode register 0 (TRJMR0) to the default value or a value other than 001B.

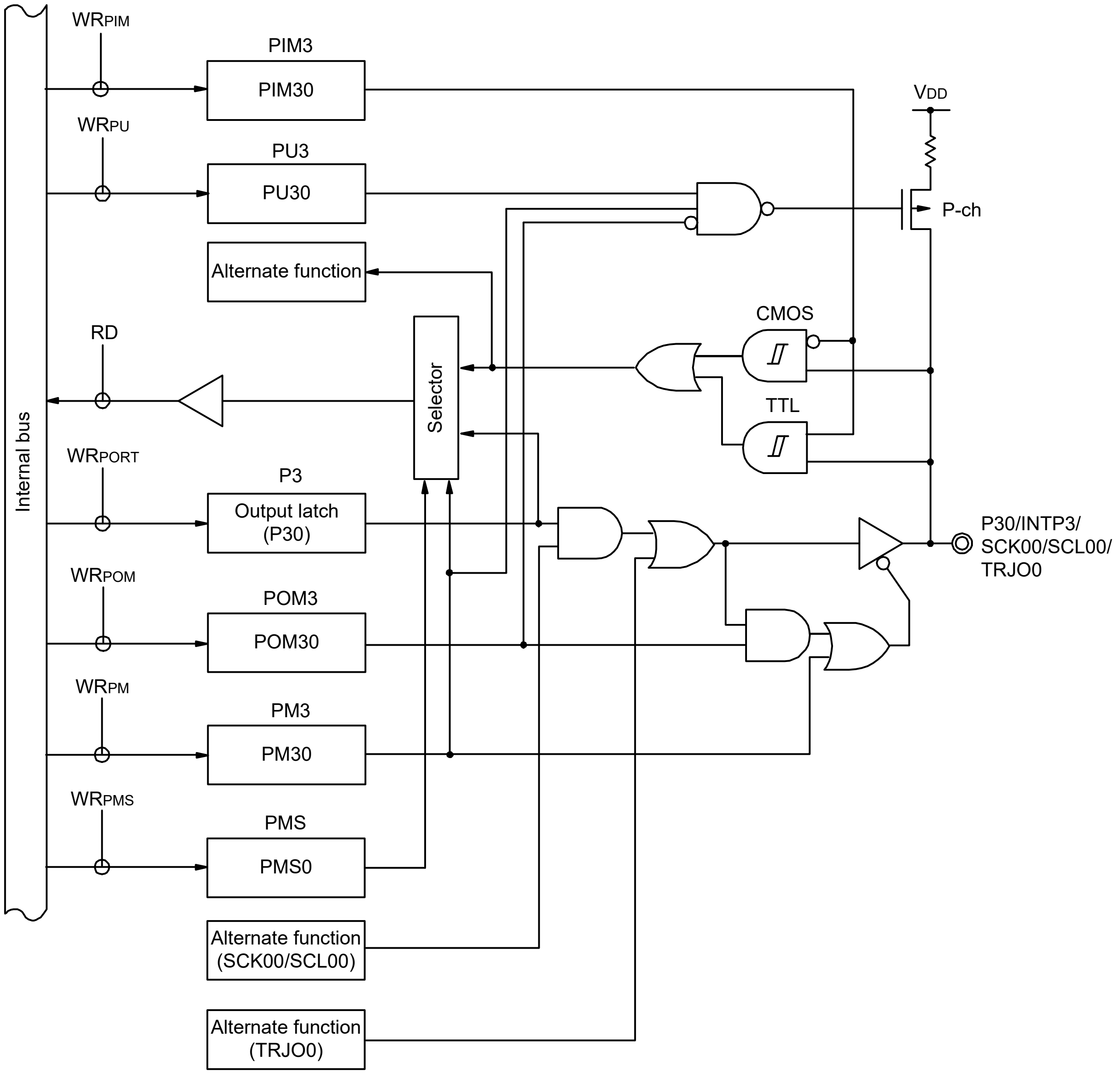
Note 6. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

Remark

×:	don't care
PM3x:	Port mode register 3
PIM3x:	Port input mode register 3
POM3x:	Port output mode register 3

For example, Figures 4 - 12 and 4 - 13 show block diagrams of port 3 for 44-pin products when PIOR1 = 00H.

Figure 4 - 12 Block Diagram of P30

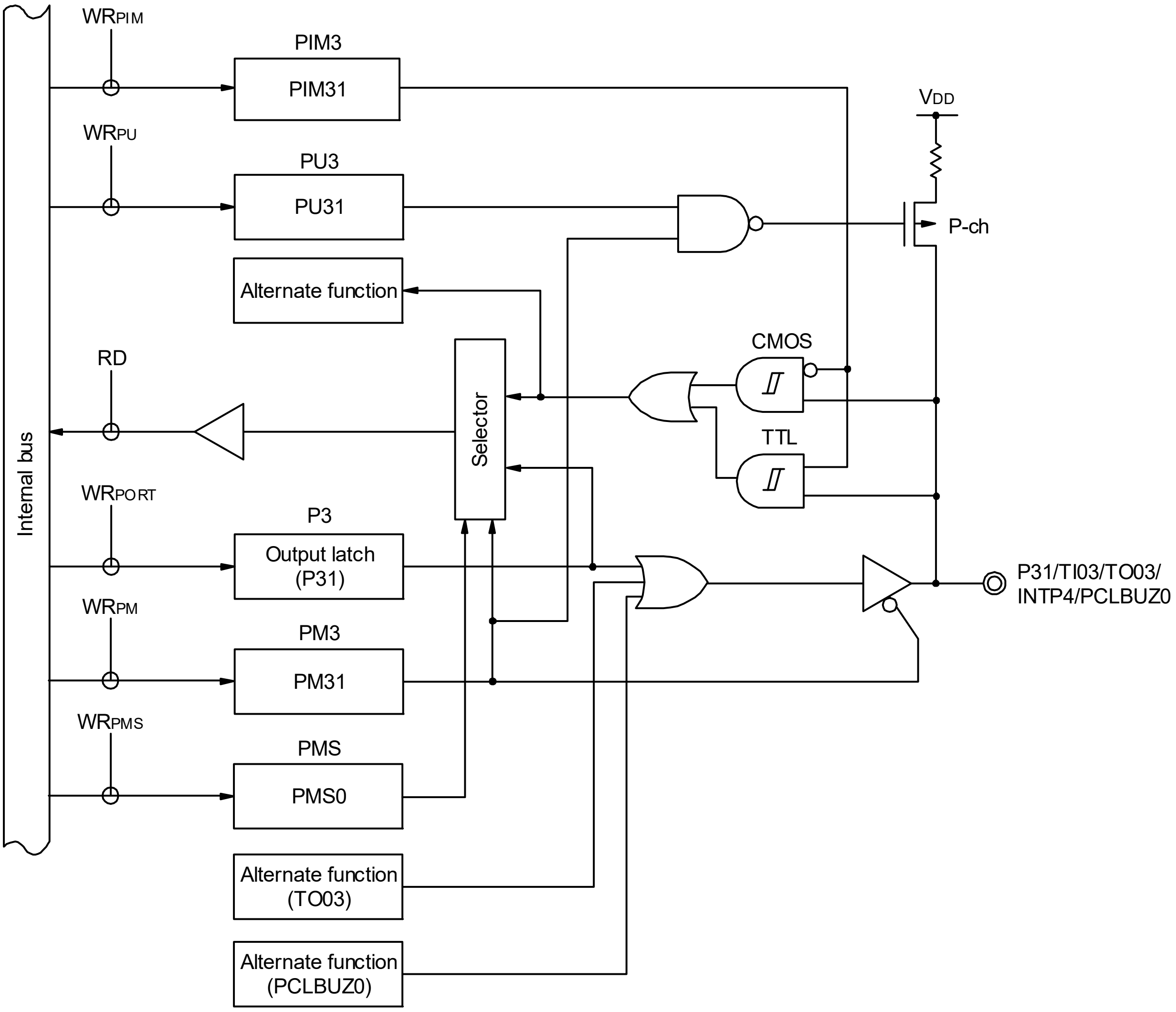


Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.
It is recommended to input a low level to prevent increase of the electric power.

P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
PIM3:	Port input mode register 3
POM3:	Port output mode register 3
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 13 Block Diagram of P31



Caution **Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.**

It is recommended to input a low level to prevent increase of the electric power.

P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
PIM3:	Port input mode register 3
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.5 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 and P41 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

This port can also be used for data I/O for a flash memory programmer/debugger.

Reset signal generation sets port 4 to input mode.

Table 4 - 8 Settings of Registers When Using Port 4

Pin Name		PM4x	PIM4x	POM4x	Alternate Function Setting Note	Remark
Name	I/O					
P40	Input	1	—	—	×	
	Output	0			×	
P41	Input	1	—	—	×	
	Output	0			(TRJIO0 output = 0)	

Note Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

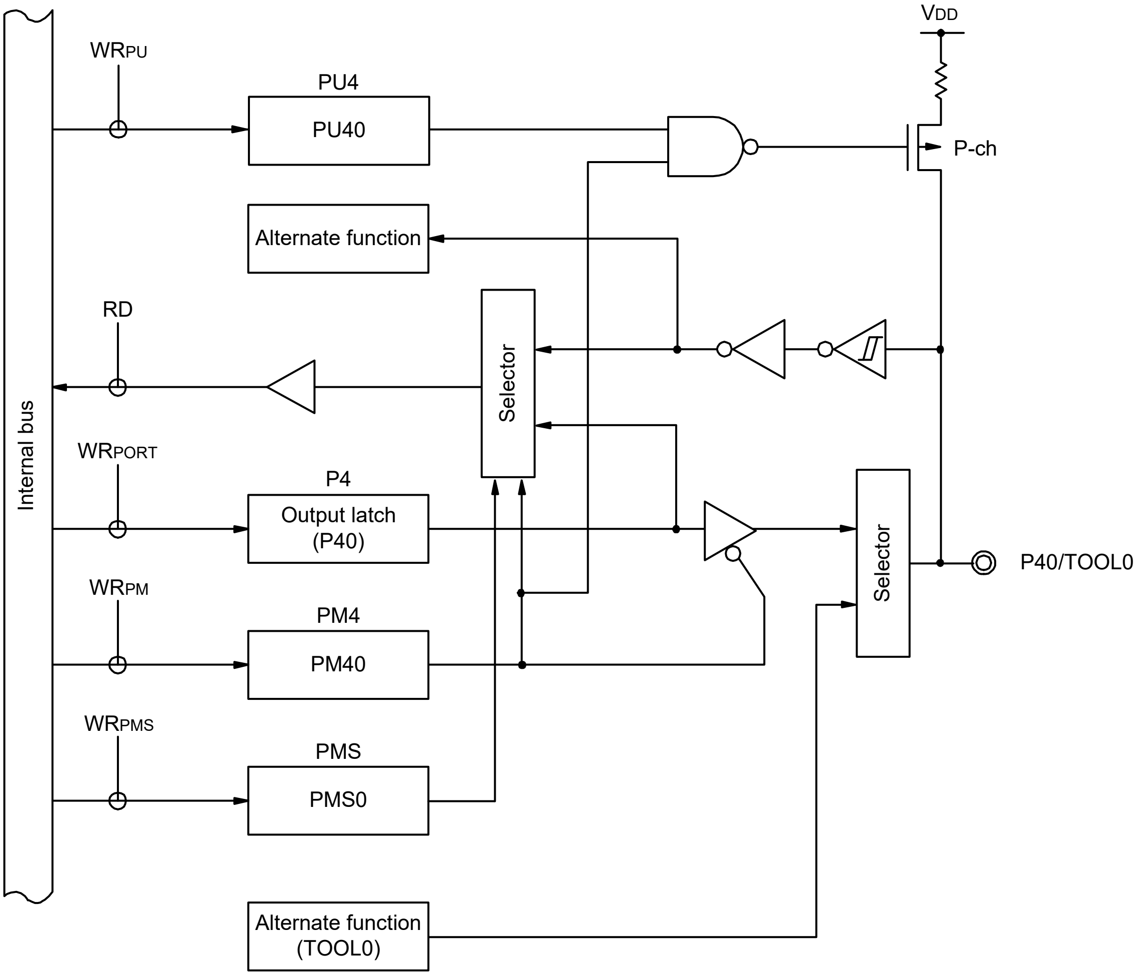
Caution **When a tool is connected, the P40 pin cannot be used as a port pin.**

Remark

×:	don't care
PM4x:	Port mode register 4
PIM4x:	Port input mode register 4
POM4x:	Port output mode register 4

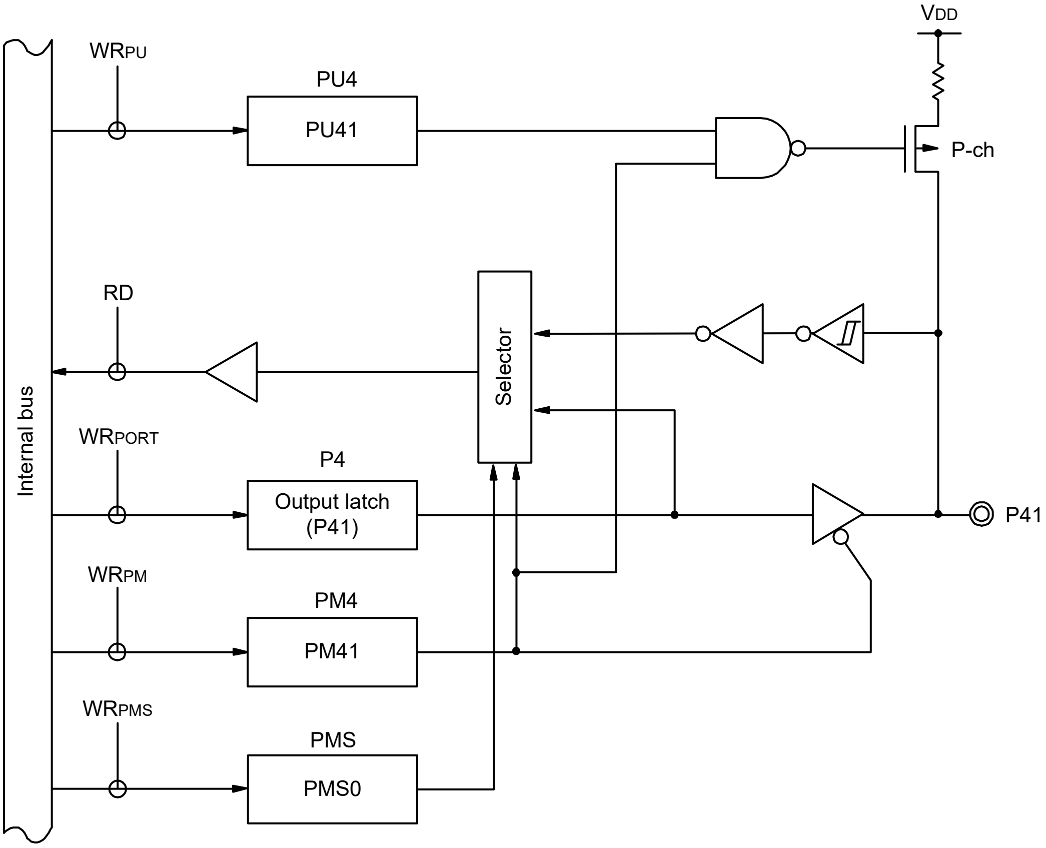
For example, Figures 4 - 14 and 4 - 15 show block diagrams of port 4 for 44-pin products.

Figure 4 - 14 Block Diagram of P40



P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 15 Block Diagram of P41



P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 and P51 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Input to the P50 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).

Output from the P50 and P51 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).

This port can also be used for external interrupt request input, serial interface data I/O, and programming UART transmission/reception.

Reset signal generation sets port 5 to input mode.

Table 4 - 9 Settings of Registers When Using Port 5

Pin Name		PM5x	PIM5x	POM5x	Alternate Function Setting Note 3	Remark
Name	I/O					
P50	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	SDA00 output = 1 Note 1 (TRJO0 output = 0)	CMOS output
		0	×	1		N-ch O.D. output
P51	Input	1	—	×	×	
	Output	0		0	SO00/TxD0 output = 1 Note 2	CMOS output
		0		1		N-ch O.D. output

Note 1. To use a pin multiplexed with the serial array unit function as a general-purpose port, set the CKOmn bit in serial output register m (SOm), the SOEmn bit in serial output enable register m (SOEm), and the SEmn bit in serial channel enable status register m (SEm) for the corresponding unit channel to the default value. (mn = 00)

Note 2. To use a pin multiplexed with the serial array unit function as a general-purpose port, set the SOmn bit in serial output register m (SOm), the SOEmn bit in serial output enable register m (SOEm), and the SEmn bit in serial channel enable status register m (SEm) for the corresponding unit channel to the default value. (mn = 00)

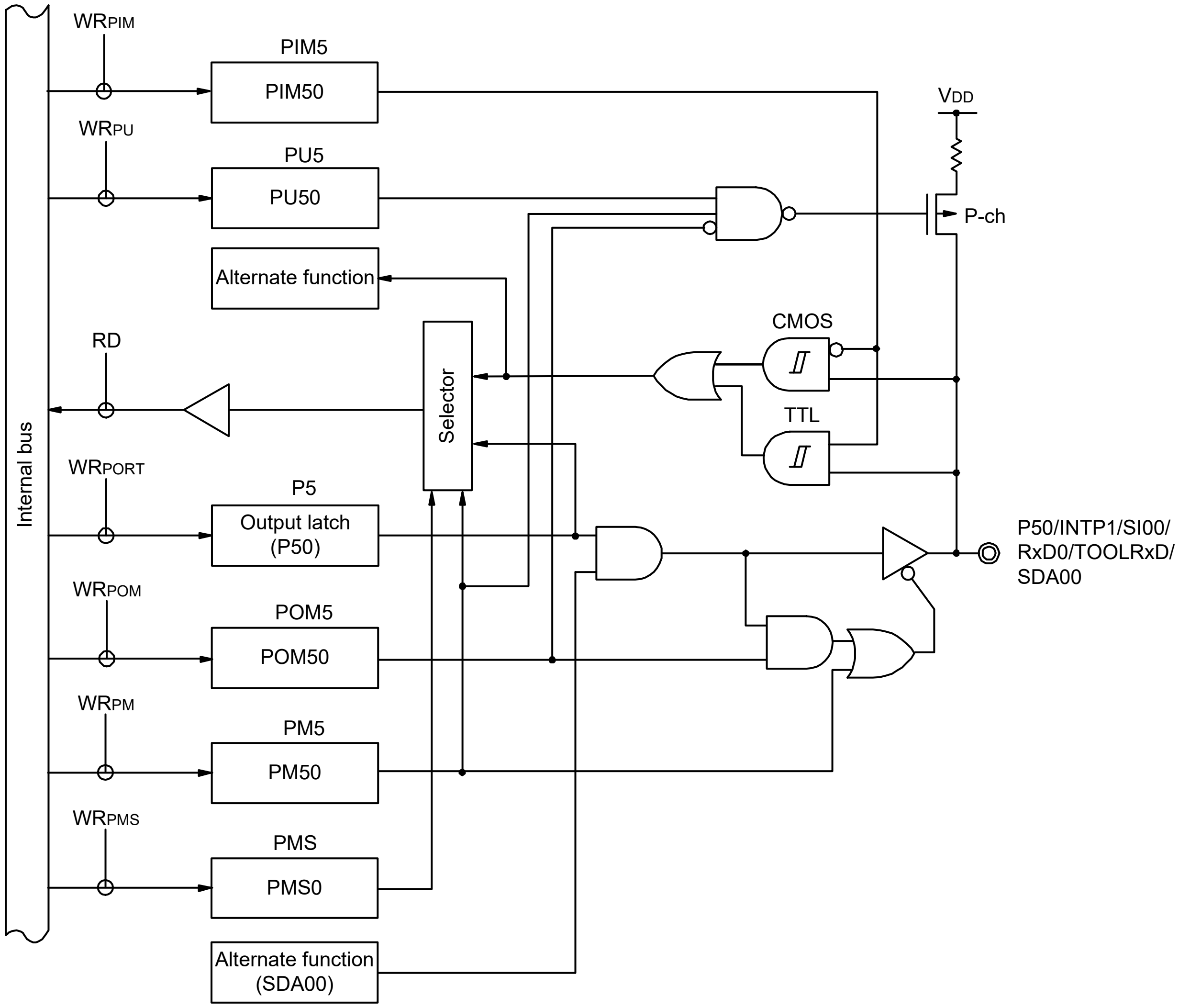
Note 3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

Remark

×:	don't care
PM5x:	Port mode register 5
PIM5x:	Port input mode register 5
POM5x:	Port output mode register 5

For example, Figures 4 - 16 and 4 - 17 show block diagrams of port 5 for 44-pin products when PIOR1 = 00H.

Figure 4 - 16 Block Diagram of P50

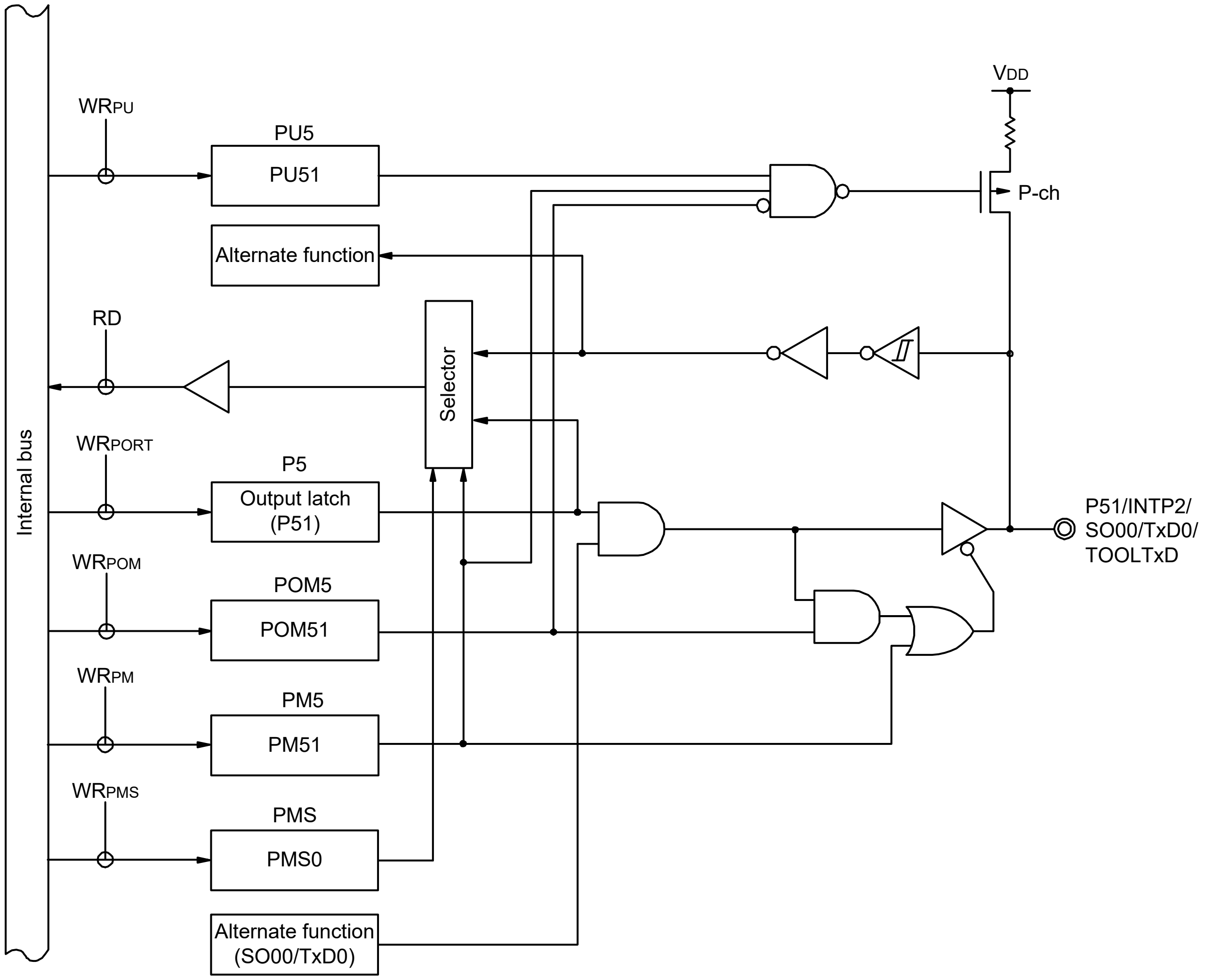


Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, electric power may increase in the case of high level input.
It is recommended to input a low level to prevent increase of the electric power.

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIM5: Port input mode register 5
POM5: Port output mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4 - 17 Block Diagram of P51



Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
POM5: Port output mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

4.2.7 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 6 (PU6).

This port can also be used for chip select input.

Reset signal generation sets port 6 to input mode.

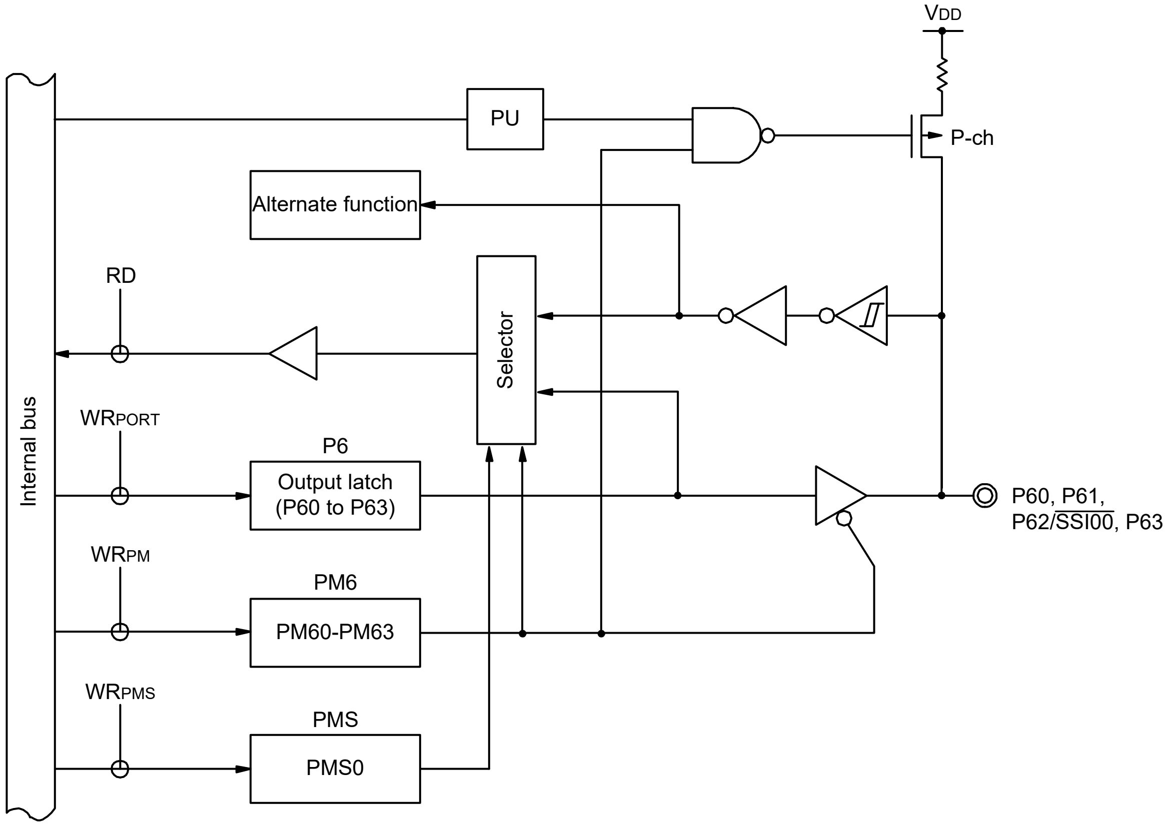
Table 4 - 10 Settings of Registers When Using Port 6

Pin Name		PM6x	Alternate Function Setting	Remark
Name	I/O			
P60	Input	1	×	
	Output	0	×	
P61	Input	1	×	
	Output	0	×	
P62	Input	1	×	
	Output	0	×	
P63	Input	1	×	
	Output	0	×	

Remark ×: don't care
PM6x: Port mode register 6

For example, Figure 4 - 18 show block diagrams of port 6 for 44-pin products.

Figure 4 - 18 Block Diagram of P60 to P63



P6:	Port register 6
PM6:	Port mode register 6
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.8 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for key interrupt input.

Reset signal generation sets port 7 to input mode.

Table 4 - 11 Settings of Registers When Using Port 7

Pin Name		PM7x	Alternate Function Setting	Remark
Name	I/O			
P70	Input	1	×	
	Output	0	×	
P71	Input	1	×	
	Output	0	×	
P72	Input	1	×	
	Output	0	×	
P73	Input	1	×	
	Output	0	×	

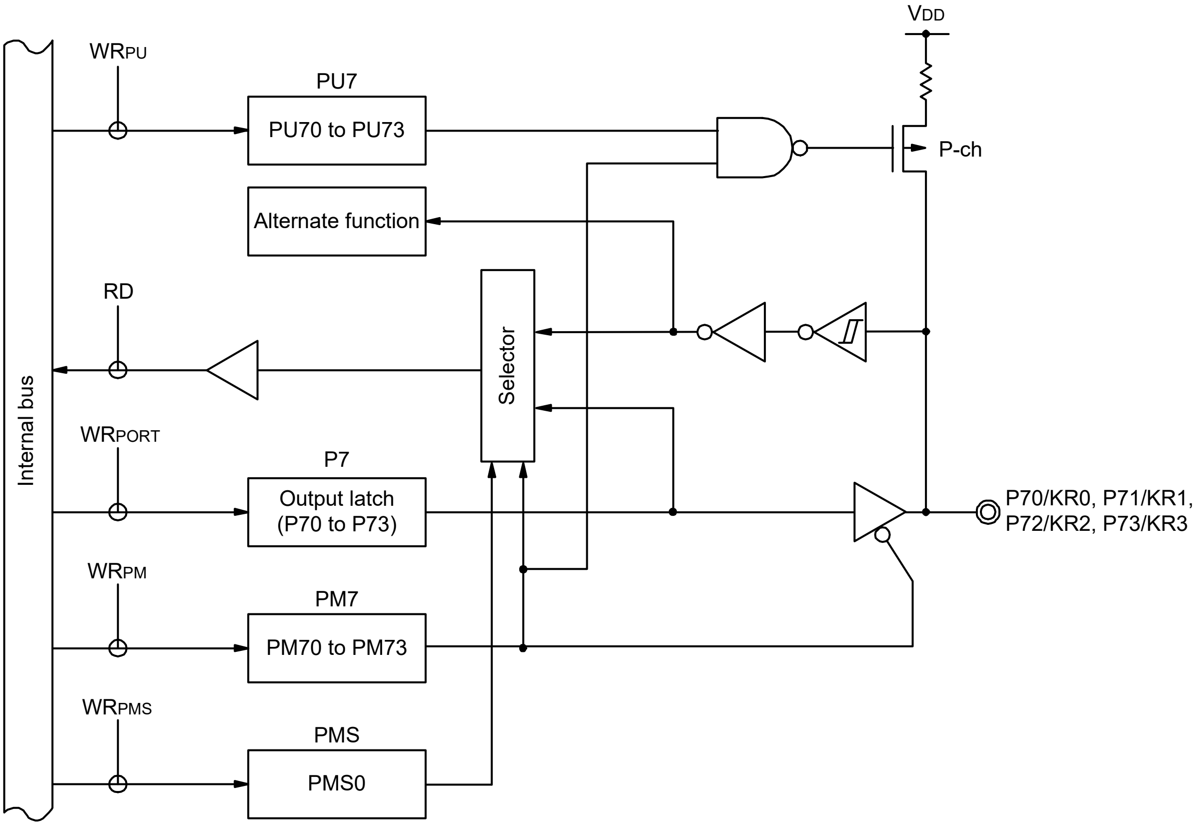
Remark ×: don't care

PM7x: Port mode register 7

POM7x: Port output mode register 7

For example, Figure 4 - 19 show block diagrams of port 7 for 44-pin products.

Figure 4 - 19 Block Diagram of P70 to P73



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.2.9 Port 12

P120 is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).
P121 to P124 are 4-bit input ports.
Input to the P120 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 12 (PMC12).
This port can also be used for A/D converter analog input, comparator analog input, connecting resonator for main system clock, and external clock input for main system clock.
Reset signal generation sets P120 to analog input, and sets P121 to P124 to input mode.

Table 4 - 12 Settings of Registers When Using Port 12

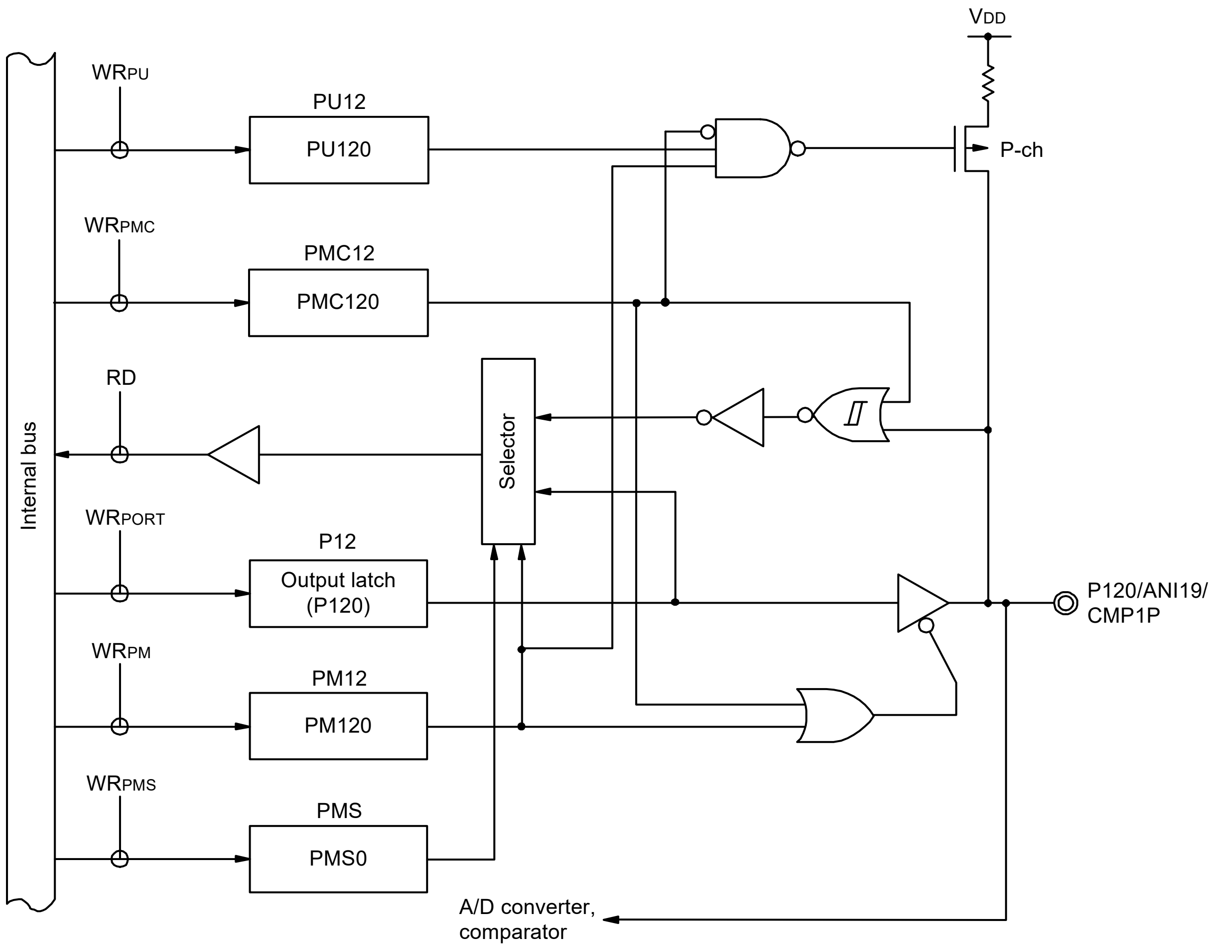
Pin Name		PM12	PMC12x	Alternate Function Setting	Remark
Name	I/O				
P120	Input	1	0	×	
	Output	0	0	×	
P121	Input	—	—	OSCSEL bit of CMC register = 0 or EXCLK bit = 1	
P122	Input	—	—	OSCSEL bit of CMC register = 0	
P123	Input	—	—	×	
P124	Input	—	—	×	

Caution **The function setting on P121 and P122 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.**

Remark ×: don't care
PM12x: Port mode register 12
PMC12x: Port mode control register 12

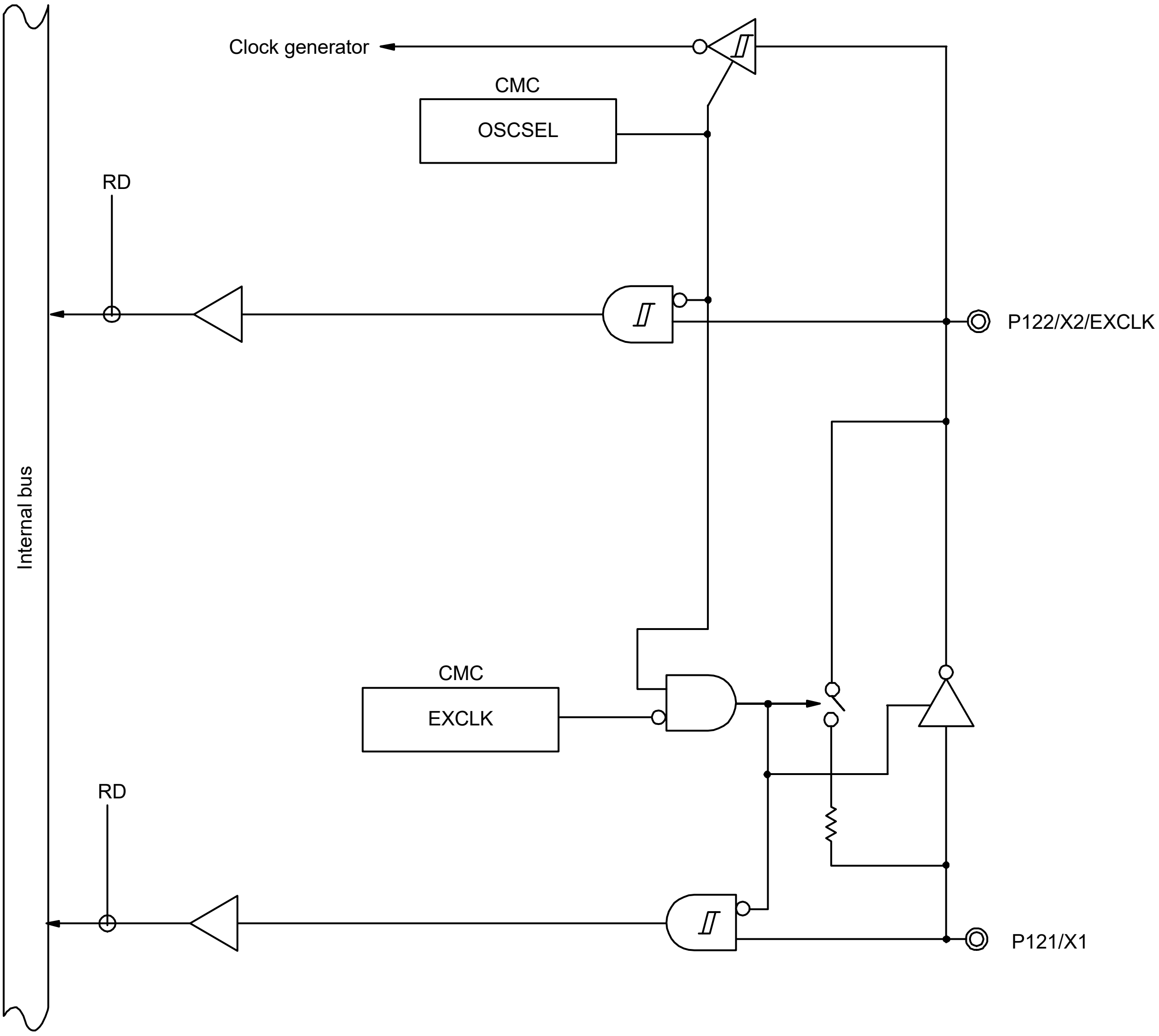
For example, Figures 4 - 20 and 4 - 21 show block diagrams of port 12 for 44-pin products.

Figure 4 - 20 Block Diagram of P120



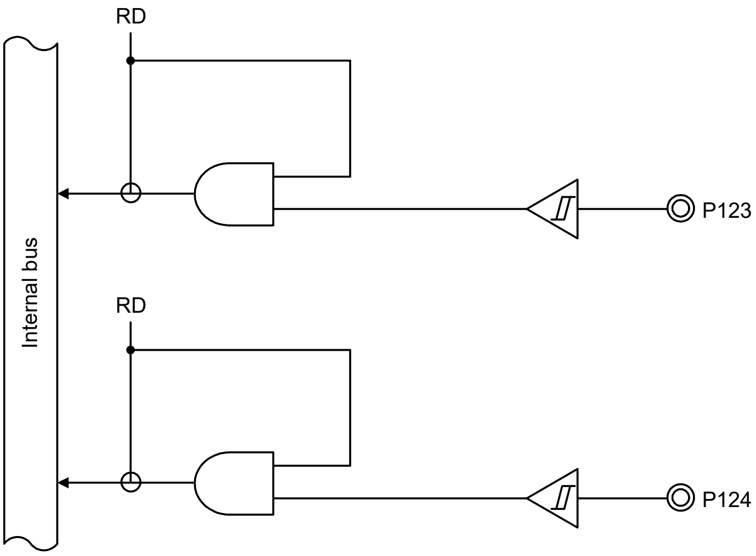
P12:	Port register 12
PU12:	Pull-up resistor option register 12
PM12:	Port mode register 12
PMC12:	Port mode control register 12
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 21 Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4 - 22 Block Diagram of P123 and P124



RD: Read signal

4.2.10 Port 13

P137 is a 1-bit input-only port.

P137 is fixed an input ports.

This port can also be used for external interrupt request input.

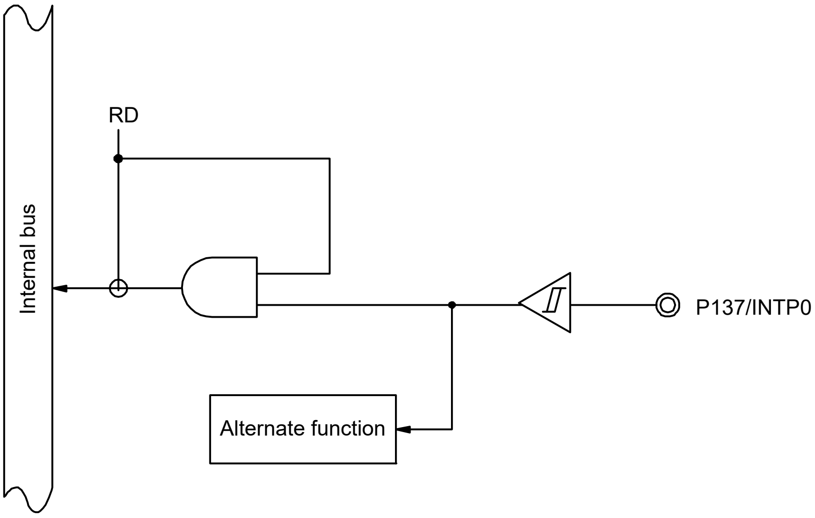
Table 4 - 13 Settings of Registers When Using Port 13

Pin Name		Alternate Function Setting	Remark
Name	I/O		
P137	Input	×	

Remark ×:don't care

For example, Figure 4 - 23 show block diagrams of port 13 for 44-pin products.

Figure 4 - 23 Block Diagram of P137



RD: Read signal

4.2.11 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P146 and P147 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

Input to the P147 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 14 (PMC14).

The P147 pin can also be used for A/D converter analog input.

Reset signal generation sets P146 to input mode, and sets P147 to analog input mode.

Table 4 - 14 Settings of Registers When Using Port 14

Pin Name		PM14x	PMC14x	Alternate Function Setting	Remark
Name	I/O				
P146	Input	1	—	×	
	Output	0	—	×	
P147	Input	1	0	×	
	Output	0	0	×	

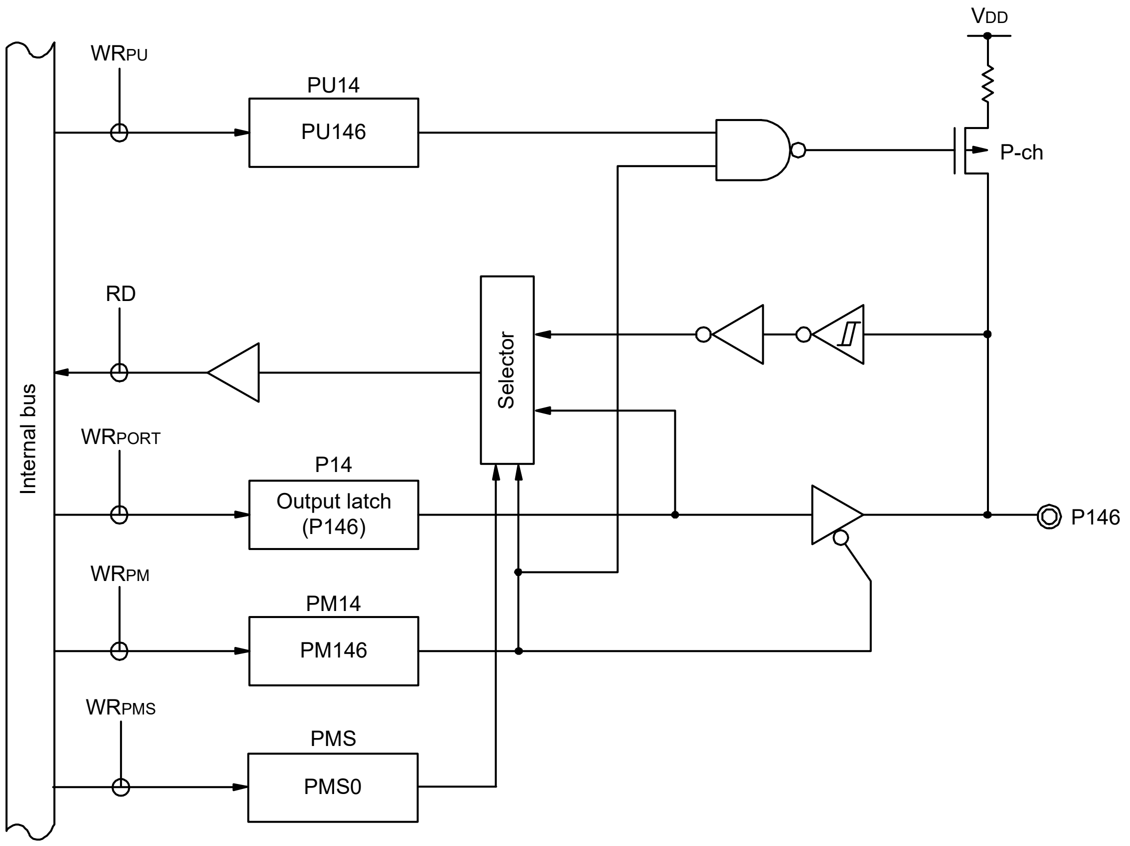
Remark ×: don't care

PM14x: Port mode register 14

PMC14x: Port mode control register 14

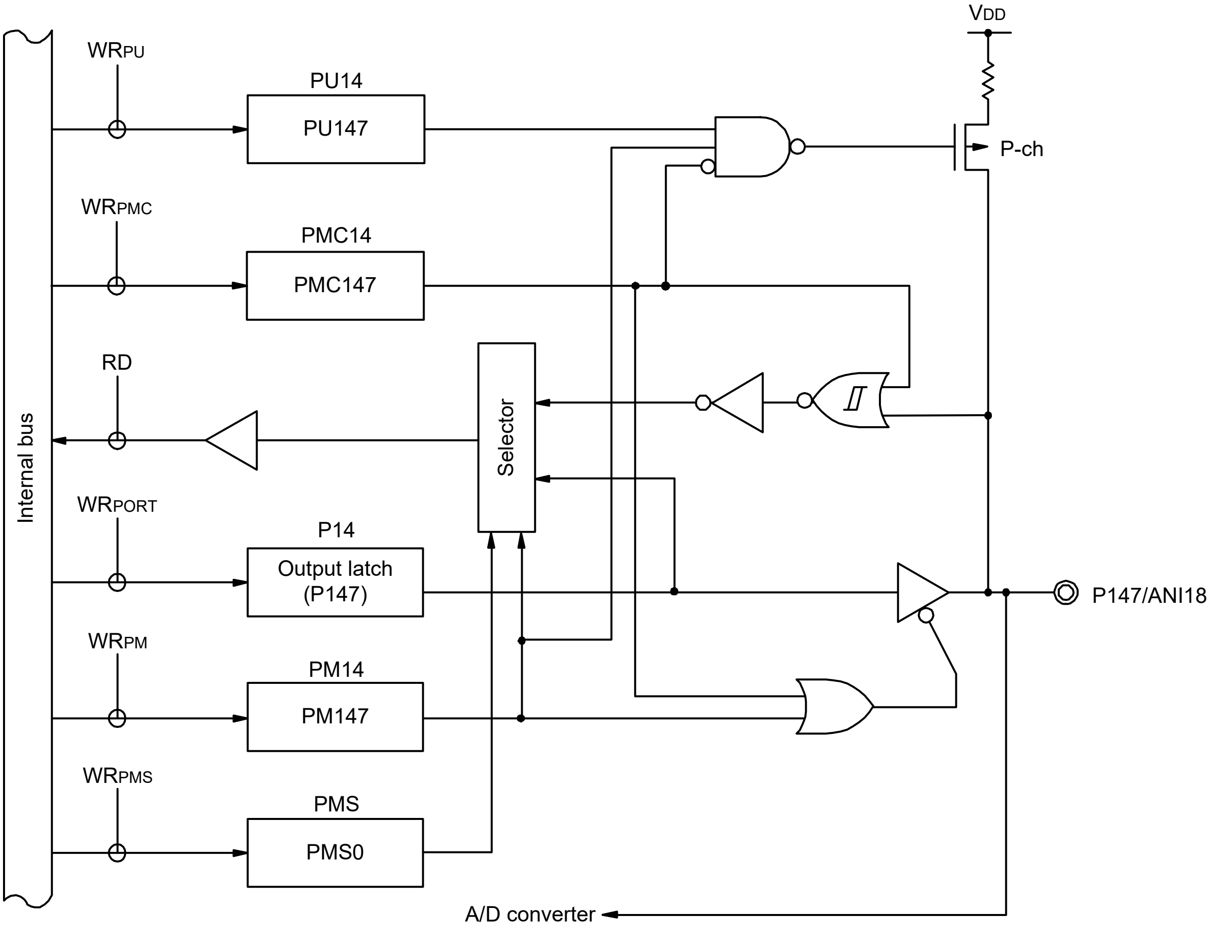
For example, Figures 4 - 24 and 4 - 25 show block diagrams of port 14 for 44-pin products.

Figure 4 - 24 Block Diagram of P146



P14:	Port register 14
PU14:	Pull-up resistor option register 14
PM14:	Port mode register 14
PMC14:	Port mode control register 14
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

Figure 4 - 25 Block Diagram of P147



P14:	Port register 14
PU14:	Pull-up resistor option register 14
PM14:	Port mode register 14
PMC14:	Port mode control register 14
PMS:	Port mode select register
RD:	Read signal
WRxx:	Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)
- A/D port configuration register (ADPC)
- Peripheral I/O redirection register 1 (PIOR1)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Tables 4 - 15 to 4 - 17. Be sure to set bits that are not mounted to their initial values.

Table 4 - 15 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and The Bits Mounted on Each Product (30-pin Products, 32-pin Products, and 44-pin Products) (1/3)

Port		Bit name						44-pin	32-pin	30-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register			
Port 0	0	PM00	P00	PU00	—	POM00	PMC00	√	√	√
	1	PM01	P01	PU01	PIM01	—	PMC01	√	√	√
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	—	√	√	√
	1	PM11	P11	PU11	—	—	—	√	√	√
	2	PM12	P12	PU12	—	—	—	√	√	√
	3	PM13	P13	PU13	—	—	—	√	√	√
	4	PM14	P14	PU14	—	—	—	√	√	√
	5	PM15	P15	PU15	PIM15	POM15	—	√	√	√
	6	PM16	P16	PU16	PIM16	—	—	√	√	√
	7	PM17	P17	PU17	PIM17	POM17	—	√	√	√
Port 2	0	PM20	P20	—	—	—	—	√	√	√
	1	PM21	P21	—	—	—	—	√	√	√
	2	PM22	P22	—	—	—	—	√	√	√
	3	PM23	P23	—	—	—	—	√	√	√
	4	PM24	P24	—	—	—	—	√	—	—
	5	PM25	P25	—	—	—	—	√	—	—
	6	PM26	P26	—	—	—	—	√	—	—
	7	PM27	P27	—	—	—	—	√	—	—

Table 4 - 16 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and The Bits Mounted on Each Product (30-pin Products, 32-pin Products, and 44-pin Products) (2/3)

Port		Bit name						44-pin	32-pin	30-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register			
Port 3	0	PM30	P30	PU30	PIM30	POM30	—	√	√	√
	1	PM31	P31	PU31	PIM31	—	—	√	√	√
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	√	√	√
	1	PM41	P41	PU41	—	—	—	√	—	—
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 5	0	PM50	P50	PU50	PIM50	POM50	—	√	√	√
	1	PM51	P51	PU51	—	POM51	—	√	√	√
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	PU60	—	—	—	√	√	√
	1	PM61	P61	PU61	—	—	—	√	√	√
	2	PM62	P62	PU62	—	—	—	√	√	—
	3	PM63	P63	PU63	—	—	—	√	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	√	√	—
	1	PM71	P71	PU71	—	—	—	√	—	—
	2	PM72	P72	PU72	—	—	—	√	—	—
	3	PM73	P73	PU73	—	—	—	√	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—

Table 4 - 17 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and The Bits Mounted on Each Product (30-pin Products, 32-pin Products, and 44-pin Products) (3/3)

Port		Bit name						44-pin	32-pin	30-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register			
Port 12	0	PM120	P120	PU120	—	—	PMC120	√	√	√
	1	—	P121	—	—	—	—	√	√	√
	2	—	P122	—	—	—	—	√	√	√
	3	—	P123	—	—	—	—	√	—	—
	4	—	P124	—	—	—	—	√	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—
Port 13	0	—	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	√	√	√
Port 14	0	—	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	√	—	—
	7	PM147	P147	PU147	—	—	PMC147	√	√	√

The format of each register is described below. The description here uses the 44-pin products as an example.

For the registers mounted on others than 44-pin products, refer to **Tables 4 - 15** to **4 - 17**.

4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Setting for Used Port and Alternate Functions**.

Figure 4 - 26 Format of Port mode register (44-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	1	1	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	1	1	1	1	1	1	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 2 to 7 of the PM0 register, bits 2 to 7 of the PM3 register, bits 2 to 7 of the PM4 register, bits 2 to 7 of the PM5 register, bits 4 to 7 of the PM6 register, bits 4 to 7 of the PM7 register, bits 1 to 7 of the PM12 register, and bits 0 to 5 of the PM14 register to "1".

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read Note.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Note If P00, P01, P20 to P27, P120, and P147 are set up as analog inputs of the A/D converter, or when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4 - 27 Format of Port register (44-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	0	0	0	0	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W Note
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R/W Note
P14	P147	P146	0	0	0	0	0	0	FFF0EH	00H (output latch)	R/W

Pmn	m = 0 to 7, 12 to 14; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124, and P137 are read-only.

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H (Only PU4 is set to 01H).

Figure 4 - 28 Format of Pull-up resistor option register (44-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU6	0	0	0	0	PU63	PU62	PU61	PU60	F0036H	00H	R/W
PU7	0	0	0	0	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	PU146	0	0	0	0	0	0	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

4.3.4 Port input mode registers (PIMxx)

These registers set the input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4 - 29 Format of Port input mode register (44-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	0	0	0	0	PIM10	F0041H	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	PIM30	F0043H	00H	R/W
PIM5	0	0	0	0	0	0	0	PIM50	F0045H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 3, 5; n = 0, 1, 5 to 7)
0	Normal input buffer
1	TTL input buffer

Caution Be sure to set bits that are not mounted to their initial values.

4.3.5 Port output mode registers (POMxx)

These registers set the output mode in 1-bit units.

N-ch open-drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 pin during simplified I^{2}C communication with an external device of the same potential.

In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode is set.

Figure 4 - 30 Format of Port output mode register (44-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W
POM1	POM17	0	POM15	0	0	0	0	POM10	F0051H	00H	R/W
POM3	0	0	0	0	0	0	0	POM30	F0053H	00H	R/W
POM5	0	0	0	0	0	0	POM51	POM50	F0055H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 3, 5; n = 0, 1, 5, 7)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Caution Be sure to set bits that are not mounted to their initial values.

4.3.6 Port mode control registers 0, 12, 14 (PMCxx)

These registers set the P00, P01, P120, and P147 digital I/O/analog input in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 4 - 31 Format of Port mode control register (32-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	1	1	1	PMC01	PMC00	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 12, 14; n = 0, 1, 7)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Caution Be sure to set bits that are not mounted to their initial values.

4.3.7 A/D port configuration register (ADPC)

This register is used to switch the P20/ANI0, P21/ANI1, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 pins to digital I/O of port or analog input of A/D converter.

The ADPC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 4 - 32 Format of A/D port configuration register (ADPC)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0

ADPC3	ADPC2	ADPC1	ADPC0	Analog I/O (A)/digital I/O (D) switching							
				ANI7/ P27	ANI6/ P26	ANI5/ P25	ANI4/ P24	ANI3/ P23	ANI2/ P22	ANI1/ P21	ANI0/ P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Caution 1. Set the port to analog input by ADPC register to the input mode by using port mode register 2 (PM2).

Caution 2. Do not set the pin set by the ADPC register as digital I/O by the analog input channel specification register (ADS).

Caution 3. When using AVREFP and AVREFM, set ANI0 and ANI1 to analog input and set the port mode register to the input mode.

4.3.8 Peripheral I/O redirection register 1 (PIOR1)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

This function is used to switch ports to which alternate functions are assigned.

The PIOR1 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 4 - 33 Format of Peripheral I/O redirection register 1 (PIOR1)

Address: F0079H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR1	0	0	0	0	PIOR13	PIOR12	PIOR11	PIOR10

PIOR13	PIOR12	Timer RJ TRJO0 pin select
0	0	Double as P30/INTP3/SCK00/SCL00
0	1	Double as P50/INTP1/SI00/RxD0/TOOLRxD/SDA00
1	0	Double as P00/ANI17/TI00/TxD1/CMP0P
1	1	Setting prohibited

PIOR11	PIOR10	Timer RJ TRJIO0 pin select
0	0	Double as P01/ANI16/TO00/RxD1/PGAI
0	1	Double as P31/TI03/TO03/INTP4/PCLBUZ0
1	0	Double as P41[Note]
1	1	Setting prohibited

Note Setting is enabled in 44-pin products only. Setting is prohibited in 30-pin and 32-pin products.

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change.

The data of the output latch is cleared when a reset signal is generated. Therefore, byte data can be written to the ports used for both input and output.

4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (2.5 V or 3 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx).
I/O connection with an external device operating on 2.5 V or 3 V when the system is operating on V_{DD} = 4.0 V to 5.5 V is still possible via the serial interface by using ports 0, 1, 4, and 5.

External device	V_{DD}
3 V	4.0 V ≤ V_{DD} ≤ 5.5 V
2.5 V	3.3 V ≤ V_{DD} ≤ 4.0 V

When receiving input from an external device with a different potential (2.5 V or 3 V), set the port input mode registers 0, 1, 3, and 5 (PIM0, PIM1, PIM3, and PIM5) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.
When outputting data to an external device with a different potential (2.5 V or 3 V), set the port output mode registers 0, 1, 3, and 5 (POM0, POM1, POM3, and POM5) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

(1) Setting procedure when using input pins of UART0, UART1, and CSI00 functions for the TTL input buffer

In case of UART0:	P50
In case of UART1:	P01
In case of CSI00:	P30, P50

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> Set the corresponding bit of the PIM0, PIM1, PIM3, and PIM5 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<3> Enable the operation of the serial array unit and set the mode to the UART or simplified SPI (CSI[Note]) mode.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

(2) Setting procedure when using output pins of UART0, UART1, and CSI00 functions in N-ch open-drain output mode

In case of UART0: P51
In case of UART1: P00
In case of CSI00: P30, P51

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM1, POM3, and POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART or simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM0, PM1, PM3, and PM5 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O pins of IIC00 function with a different potential (2.5 V, 3 V)

In case of simplified IIC00: P30, P50

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM3 and POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM3 and PIM5 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
<7> Set the corresponding bit of the PM3 and PM5 registers to the output mode (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

4.5 Register Setting for Used Port and Alternate Functions

Register setting examples for used port and alternate functions are shown in Tables 4 - 18 to 4 - 23.

Caution If the output function of an alternate function is assigned to a pin that is also used as an output pin, the output of the unused alternate function must be set to its initial state.

Table 4 - 18 Setting Examples of Registers When Using P00 to P17 Pin Function (1/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P00	P00	Input	—	×	0	1	×	×	√	√	√
		Output	—	0	0	0	0/1	TXD1 = 1 (TRJO0 = 0) Note 2			
		N-ch open-drain output	—	1	0	0	0/1				
	ANI17 Note 1	Analog input	—	×	1	1	×	—	√	√	√
	CMP0P	Input	—	×	1	1	×	—	√	√	√
	TI00	Input	—	×	0	1	×	—	√	√	√
	TxD1	Output	—	0/1	0	0	1	(TRJO0 = 0) Note 2	√	√	√
	(TRJO0)	Output	PIOR13, PIOR12 = 10B	0	0	0	0	TXD1 = 1	√	√	√
P01	P01	Input	—	—	0	1	×	×	√	√	√
		Output	—	—	0	0	0/1	TO00 = 0 TRJIO0 = 0 Note 3			
	ANI16 Note 1	Analog input	—	—	1	1	×	—	√	√	√
	PGAI	Input	—	—	1	1	×	—	√	√	√
	TO00	Input	—	—	0	0	0	TRJIO0 = 0 Note 3	√	√	√
	RxD1	Output	—	—	0	1	×	—	√	√	√
	TRJIO0	Input	PIOR11, PIOR10 = 00B	—	0	1	×	—	√	√	√
		Output	PIOR11, PIOR10 = 00B	—	0	0	0	TO00 = 0	√	√	√

Remark —: Not supported
×: don't care
PIORx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

(The notes are described after the last table.)

Table 4 - 18 Setting Examples of Registers When Using P00 to P17 Pin Function (2/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P10	P10	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	TRDIOD1 = 0			
		N-ch open-drain output	—	1	—	0	0/1				
	TRDIOD1	Input	—	×	—	1	×	—	√	√	√
		Output	—	0	—	0	0	—	√	√	√
P11	P11	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TRDIOC1 = 0			
	TRDIOC1	Input	—	—	—	1	×	—	√	√	√
		Output	—	—	—	0	0	—	√	√	√
P12	P12	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TRDIOB1 = 0			
	TRDIOB1	Input	—	—	—	1	×	—	√	√	√
		Output	—	—	—	0	0	—	√	√	√
P13	P13	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TRDIOA1 = 0			
	TRDIOA1	Input	—	—	—	1	×	—	√	√	√
		Output	—	—	—	0	0	—	√	√	√
P14	P14	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TRDIOD0 = 0			
	TRDIOD0	Input	—	—	—	1	×	—	√	√	√
		Output	—	—	—	0	0	—	√	√	√
P15	P15	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	PCLBUZ1 = 0 TRDIOB0 = 0			
		N-ch open-drain output	—	1	—	0	0/1				
	PCLBUZ1	Output	—	×	—	0	0	TRDIOB0 = 0	√	√	√
	TRDIOB0	Input	—	×	—	1	×	—	√	√	√
		Output	—	0	—	0	0	—	√	√	√

Remark —: Not supported

×: don't care

PIORx: Peripheral I/O redirection register

POMxx: Port output mode register

PMCxx: Port mode control register

PMxx: Port mode register

Pxx: Port output latch

Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

(The notes are described after the last table.)

Table 4 - 18 Setting Examples of Registers When Using P00 to P17 Pin Function (3/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P16	P16	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TO01 = 0 TRDIOC0 = 0			
	TI01	Input	—	—	—	1	×	—	√	√	√
	TO01	Output	—	—	—	0	0	TRDIOC0 = 0	√	√	√
	INTP5	Input	—	—	—	1	×	—	√	√	√
	TRDIOC0	Input	—	—	—	1	×	—	√	√	√
		Output	—	—	—	0	0	TO01 = 0	√	√	√
P17	P17	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	TO02 = 0 TRDIOA0 = 0			
		N-ch open-drain output	—	1	—	0	0/1				
	TI02	Input	—	×	—	1	×	—	√	√	√
	TO02	Output	—	0	—	0	0	TRDIOA0 = 0	√	√	√
	TRDIOA0	Input	—	×	—	1	×	—	√	√	√
		Output	—	0	—	0	0	TO02 = 0	√	√	√
	TRDCLK	Input	—	×	—	1	×	—	√	√	√

Remark —: Not supported

×: don't care

PIORx: Peripheral I/O redirection register

POMxx: Port output mode register

PMCxx: Port mode control register

PMxx: Port mode register

Pxx: Port output latch

Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

Note 1. The functions of the ANI16/P01, ANI17/P00, ANI18/P147, and ANI19/P120 pins can be selected by using the port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14), analog input channel specification register (ADS), and port mode registers 0, 12, 14 (PM0, PM12, PM14).

Table 4 - 19 Settings of ANI16/P01, ANI17/P00, ANI18/P147, and ANI19/P120 Pin Function

PMC0, PMC12, PMC14 Registers	PM0, PM12, PM14 Registers	ADS Register	ANI16/P01, ANI17/P00, ANI18/P147, ANI19/P120 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog I/O selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

Note 2. "Don't care" when the setting of PIOR13 and PIOR12 is not 10B.

Note 3. "Don't care" when the setting of PIOR11 and PIOR10 is not 00B.

Table 4 - 20 Setting Examples of Registers When Using P20 to P27 Pin Function

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx	30-pin	32-pin	44-pin
	Function Name	I/O							
P20	P20	Input	ADPC = 01H	×	1	×	√	√	√
		Output	ADPC = 01H	×	0	0/1			
	ANI0[Note]	Analog input	ADPC = 00H/02H to 08H	00×0××0× 10×0××0×	1	×			
	AVREFP	Reference voltage input	ADPC = 00H/02H to 08H	01×0××0×	1	×			
P21	P21	Input	ADPC = 01H/02H	×	1	×	√	√	√
		Output	ADPC = 01H/02H	×	0	0/1			
	ANI1[Note]	Analog input	ADPC = 00H/03H to 08H	××00××0×	1	×			
	AVREFM	Reference voltage input	ADPC = 00H/03H to 08H	××10××0×	1	×			
P22	P22	Input	ADPC = 01H to 03H	×	1	×	√	√	√
		Output	ADPC = 01H to 03H	×	0	0/1			
	ANI2[Note]	Analog input	ADPC = 00H/04H to 08H	×	1	×			
P23	P23	Input	ADPC = 01H to 04H	×	1	×	√	√	√
		Output	ADPC = 01H to 04H	×	0	0/1			
	ANI3[Note]	Analog input	ADPC = 00H/05H to 08H	×	1	×			
P24	P24	Input	ADPC = 01H to 05H	×	1	×	×	×	√
		Output	ADPC = 01H to 05H	×	0	0/1			
	ANI4[Note]	Analog input	ADPC = 00H/06H to 08H	×	1	×			
P25	P25	Input	ADPC = 01H to 06H	×	1	×	×	×	√
		Output	ADPC = 01H to 06H	×	0	0/1			
	ANI5[Note]	Analog input	ADPC = 00H/07H to 08H	×	1	×			
P26	P26	Input	ADPC = 01H to 07H	×	1	×	×	×	√
		Output	ADPC = 01H to 07H	×	0	0/1			
	ANI6[Note]	Analog input	ADPC = 00H/08H	×	1	×			
P27	P27	Input	ADPC = 01H to 08H	×	1	×	×	×	√
		Output	ADPC = 01H to 08H	×	0	0/1			
	ANI7[Note]	Analog input	ADPC = 00H	×	1	×			

Note The functions of the P20/ANI0, P21/ANI1, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 pins can be selected by using the A/D port configuration register (ADPC), analog input channel specification register (ADS), and port mode registers 2 (PM2).

Table 4 - 21 Setting of P20/ANI0, P21/ANI1, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, and P27/ANI7 Pin Function

ADPC Register	PM2 Register	ADS Register	P20/ANI0, P21/ANI1, P22/ANI2, P23/ANI3, P24/ANI4, P25/ANI5, P26/ANI6, P27/ANI7 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog I/O selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

Table 4 - 22 Setting Examples of Registers When Using P30 to P147 Pin Function (1/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P30	P30	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	SCK00/SCL00 = 1 TRJO0 = $0^{\text{Note 1}}$			
		N-ch open-drain output	—	1	—	0	0/1				
	INTP3	Input	—	×	—	1	×	—	√	√	√
	SCK00	Input	—	×	—	1	×	—	√	√	√
		Output	—	0/1	—	0	1	TRJO0 = $0^{\text{Note 1}}$	√	√	√
	SCL00	Output	—	0/1	—	0	1	TRJO0 = $0^{\text{Note 1}}$	√	√	√
	TRJO0	Output	PIOR13, PIOR12 = 00B	0	—	0	0	SCK00/SCL00 = 1	√	√	√
P31	P31	Input	—	×	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	TO03 = 0 PCLBUZ0 = 0 (TRJIO0 = $0)^{\text{Note 2}}$			
	TI03	Input	—	—	—	1	×	—	√	√	√
	TO03	Output	—	—	—	0	0	PCLBUZ0 = 0 (TRJIO0 = $0)^{\text{Note 2}}$	√	√	√
	INTP4	Input	—	—	—	1	×	TRJIO0 = 0	√	√	√
	PCLBUZ0	Output	—	—	—	0	0	TO03 = 0 (TRJIO0 = $0)^{\text{Note 2}}$	√	√	√
	(TRJIO0)	Input	PIOR11, PIOR10 = 01B	—	—	1	×	—	√	√	√
		Output	PIOR11, PIOR10 = 01B	—	—	0	0	TO03 = 0 PCLBUZ0 = 0	√	√	√
P40	P40	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	×			
	TOOL0	I/O	—	—	—	×	×	—	√	√	√
P41	P41	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	(TRJIO0 = $0)^{\text{Note 3}}$			
	(TRJIO0=0)	Input	PIOR11, PIOR10 = 10B	—	—	1	×	—	×	×	√
		Output	PIOR11, PIOR10 = 10B	—	—	0	0	—	×	×	√
P50	P50	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	SDA00 = 1 (TRJO0 = $0)^{\text{Note 4}}$			
		N-ch open-drain output	—	1	—	0	0/1				
	INTP1	Input	—	×	—	1	×	—	√	√	√
	SI00	Input	—	×	—	1	×	—	√	√	√
	RxD0	Input	—	×	—	1	×	—	√	√	√
	TOOLRxD	Input	—	×	—	1	×	—	√	√	√
	SDA00	I/O	—	1	—	0	1	(TRJO0 = $0)^{\text{Note 4}}$	√	√	√
	(TRJO0=0)	Output	PIOR13, PIOR12 = 01B	0	—	0	0	SDA00 = 1	√	√	√

(The notes are described after the last table.)

Table 4 - 22 Setting Examples of Registers When Using P30 to P147 Pin Function (2/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P51	P51	Input	—	×	—	1	×	×	√	√	√
		Output	—	0	—	0	0/1	SO00/TxD0 = 1			
		N-ch open-drain output	—	1	—	0	0/1				
	INTP2	Input	—	×	—	1	×	—	√	√	√
	SO00	Output	—	0/1	—	0	1	—	√	√	√
	TxD0	Output	—	0/1	—	0	1	—	√	√	√
	TOOLTxD	Output	—	0/1	—	0	1	—	√	√	√
P60	P60	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	×			
P61	P61	Input	—	—	—	1	×	×	√	√	√
		Output	—	—	—	0	0/1	×			
P62	P62	Input	—	—	—	1	×	×	×	√	√
		Output	—	—	—	0	0/1	×			
	$\overline{\text{SSI00}}$	Input	—	—	—	1	×	×	×	√	√
P63	P63	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	×			
P70	P70	Input	—	—	—	1	×	×	×	√	√
		Output	—	—	—	0	0/1	×			
	KR0	Input	—	—	—	1	×	×	×	√	√
P71	P71	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	×			
	KR1	Input	—	—	—	1	×	×	×	×	√
P72	P72	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	×			
	KR2	Input	—	—	—	1	×	×	×	×	√
P73	P73	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	×			
	KR3	Input	—	—	—	1	×	×	×	×	√
P120	P120	Input	—	—	0	1	×	×	√	√	√
		Output	—	—	0	0	0/1	×			
	CMP1P	Input	—	—	1	1	×	×	√	√	√
	ANI19 Note 5	Input	—	—	1	1	×	×	√	√	√
P121	P121	Input	—	—	—	—	×	OSCSEL bit of CMC register = 0 EXCLK bit of CMC register = 1	√	√	√

(The notes are described after the last table.)

Table 4 - 22 Setting Examples of Registers When Using P30 to P147 Pin Function (3/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate function output	30-pin	32-pin	44-pin
	Function Name	I/O									
P122	P122	Input	—	—	—	—	×	OSCSEL bit of CMC register = 0	√	√	√
P123	P123	Input	—	—	—	—	×	×	×	×	√
P124	P124	Input	—	—	—	—	×	×	×	×	√
P137	P137	Input	—	—	—	—	×	×	√	√	√
	INTP0	Input	—	—	—	—	×	×	√	√	√
P146	P146	Input	—	—	—	1	×	×	×	×	√
		Output	—	—	—	0	0/1	×			
P147	P147	Input	—	—	0	1	×	×	√	√	√
		Output	—	—	0	0	0/1	×			
	ANI18[Note 5]	Analog input	—	—	1	1	×	×	√	√	√

Note 1. "Don't care" when the setting of PIOR13 and PIOR12 is not 00B.

Note 2. "Don't care" when the setting of PIOR11 and PIOR10 is not 01B.

Note 3. "Don't care" when the setting of PIOR11 and PIOR10 is not 10B.

Note 4. "Don't care" when the setting of PIOR13 and PIOR12 is not 01B.

Note 5. The functions of the ANI16/P01, ANI17/P00, ANI18/P147, and ANI19/P120 pins can be selected by using the port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14), analog input channel specification register (ADS), and port mode registers 0, 12, 14 (PM0, PM12, PM14).

Table 4 - 23 Settings of ANI16/P01, ANI17/P00, ANI18/P147, and ANI19/P120 Pin Function

PMC0, PMC12, PMC14 Registers	PM0, PM12, PM14 Registers	ADS Register	ANI16/P01, ANI17/P00, ANI18/P147, ANI19/P120 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog I/O selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit. Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

Example: When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78/G1G.

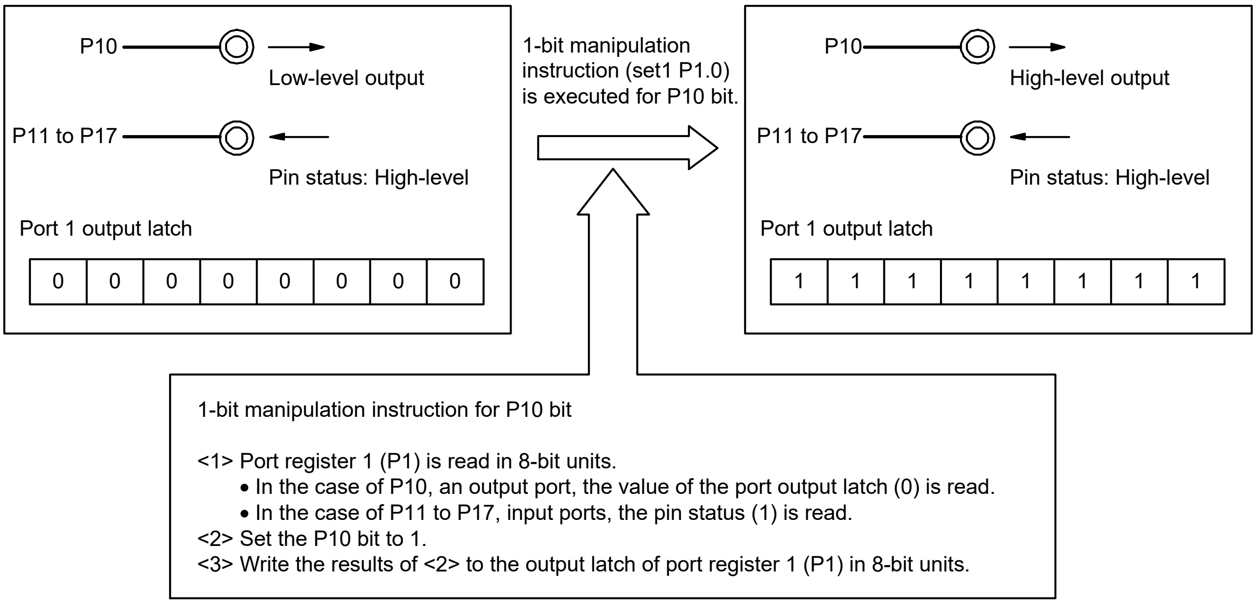
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4 - 34 Bit Manipulation Instruction (P10)



1-bit manipulation instruction for P10 bit

<1> Port register 1 (P1) is read in 8-bit units.
  - In the case of P10, an output port, the value of the port output latch (0) is read.
  - In the case of P11 to P17, input ports, the pin status (1) is read.

<2> Set the P10 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

4.6.2 Notes on specifying the pin settings

If the output function of an alternate function is assigned to a pin that is also used as an output pin, the output of the unused alternate function must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate output function, see **4.5 Register Setting for Used Port and Alternate Functions**.

No specific setting is required for input pins because the output function of their alternate functions is disabled (the buffer output is Hi-Z).

The following shows the affected units and how to handle unused alternate functions.

Table 4 - 24 Handling of Unused Alternate Functions

Affected Unit	Output or I/O Pins of Unused Alternate Functions	Handling of Unused Alternate Functions
Timer array units	TO0n	Make sure that bit 0 (TO0n) of timer output register 0 (TO0) and bit n (TOE0n) of timer output enable register 0 (TOE0) are set to their initial value (0).
Timer RJ	TRJIO0, TRJO0	Function assigned by setting peripheral I/O redirection register 1 (PIOR1). Do not assign this function when it is not used. TRJO: Make sure that bit 2 of timer RJ I/O control register 0 (TRJIOC0) is set to its initial value (0). TRJIO: Make sure that the mode select bit of timer RJ mode register 0 (TRJMR0) is set to a mode other than pulse output mode.
Timer RD	TRDIOAn, TRDIOBn, TRDIOCn, TRDIODn	Make sure that the bits (EDn, ECn, EBn, EAn) of timer RD output master enable register 1 (TRDOER1) for the corresponding pins are set to their initial value (1).
Clock/buzzer output circuit	PCLBUZn	Make sure that bit 7 (PCLOEn) of clock output select register n (CKSn) is set to its initial value (0).
Serial array units	SCK00, SO00, SCL00, SDA00	Make sure that bit n (SE0n) of serial channel enable status register 0 (SE0), bit n (SO0n) of serial output register 0 (SO0), and bit n (SOE0n) of serial output enable register 0 (SOE0) are set to their initial value (1 for SO0n and 0 for others) Note.

Note n = 0 for SCK00, SO00, SCL00, and SDA00

Example: P16/TI01/TO01/INTP5/TRDIOC0 pin of 44-pin products

(1) When the pin is used as TO01 output

P16: Specify the output mode by setting PM16 of port mode register 1 to 0.

TI01, INTP5: These are input pins, so this note does not apply.

TRDIOC0: This is an output pin, so set EC0 of output master enable register to 1.

(2) When the pin is used as TRDIOC0 output

P16: Specify the output mode by setting PM16 of port mode register 1 to 0.

TI01, INTP5: These are input pins, so this note does not apply.

TO01: This is an output pin, so set TO01 and TOE01 of timer array unit 0 to 0.

Like SCL00 when using the P30/INTP3/SCK00/SCL00 pin as the SCK00 I/O pin, changing the operation mode does not enable alternate functions assigned to pins on the same serial channel, and this note does not apply to such pins. If the simplified SPI (CSI) function is specified (MD002 = MD001 = 0), the pin does not function as a simplified I^2C pin, and therefore SCL00 output is invalid.

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended to lower power consumption.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 pin and X2 pin. Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator (High-speed OCO)

The frequency at which to oscillate can be selected from among f_{HOCO} = 48, 24, 16, 12, 8, 4, or 1 MHz (TYP.) by using the option byte (000C2H). When 48 MHz is selected as f_{HOCO}, f_{IH} is set to 24 MHz. When 24 MHz or less is selected as f_{HOCO}, f_{IH} is not divided and set to the same frequency as f_{HOCO}. After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5 - 10 Format of High-speed on-chip oscillator frequency select register (HOCODIV)**.
The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)									
	1	2	3	4	6	8	12	16	24	48
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	√

(2) Low-speed on-chip oscillator (Low-speed OCO)
This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).
The low-speed on-chip oscillator clock cannot be used as the CPU clock.
Only the following peripheral hardware runs on the low-speed on-chip oscillator clock.

- Watchdog timer
- 12-bit interval timer
- Timer RJ

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.
However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Remark f_{X}: X1 clock oscillation frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.) Note
f_{EX}: External main system clock frequency
f_{IL}: Low-speed on-chip oscillator frequency

Note f_{IH} is controlled by hardware to be set to two frequency division of f_{HOCO} when f_{HOCO} is set to 48 MHz, and the same clock frequency as f_{HOCO} when f_{HOCO} is set to 24 MHz or less. When supplying 48 MHz to timer RD, set f_{CLK} to f_{IH}.

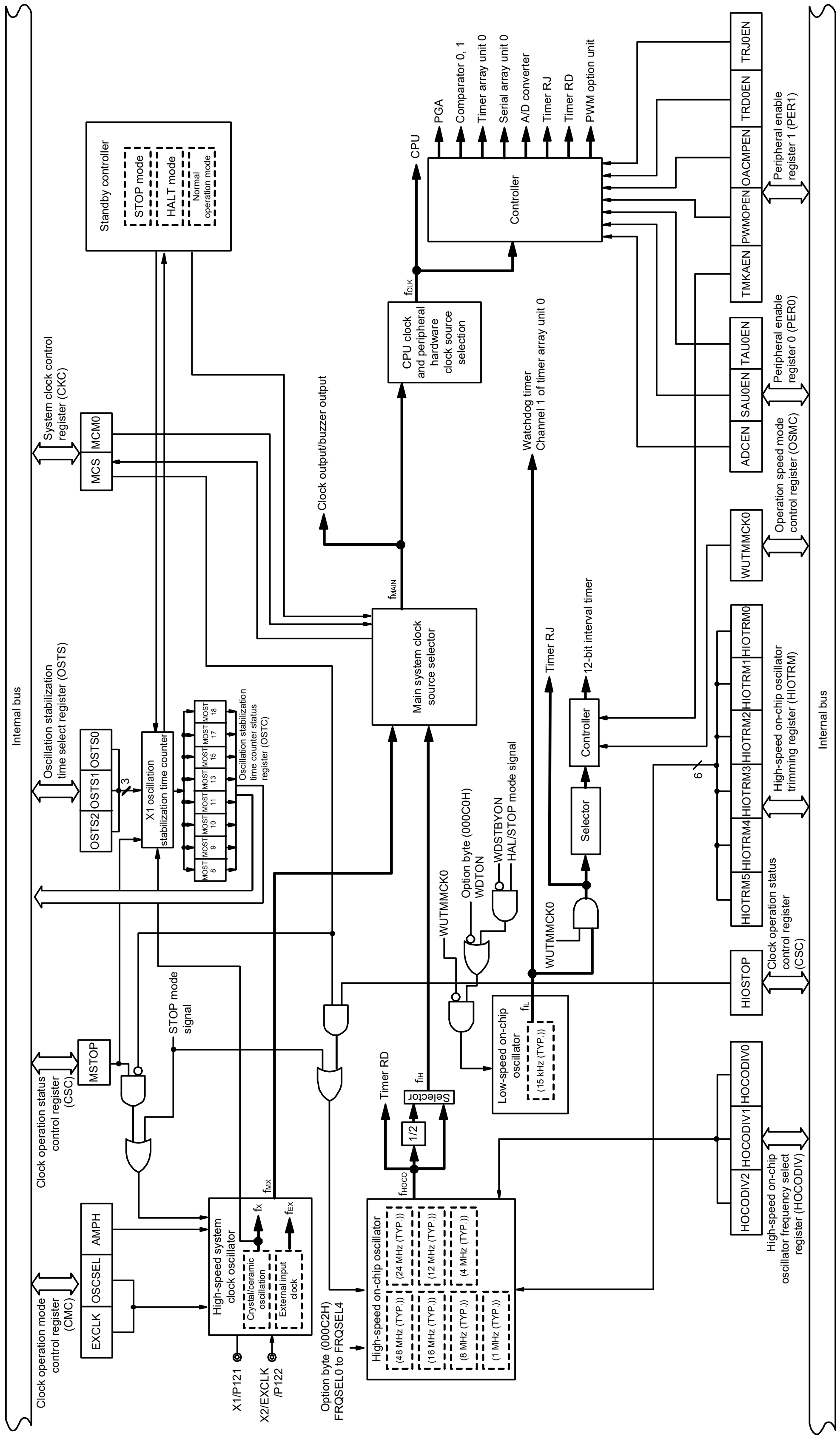
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5 - 1 Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1 (PER0, PER1) Operation speed mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator High-speed on-chip oscillator clock Low-speed on-chip oscillator clock

Figure 5 - 1 Block Diagram of Clock Generator



(**Remark** is listed on the next page after next.)

Remark f_{X}: X1 clock oscillation frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.) [Note]
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

Note f_{IH} is controlled by hardware to be set to two frequency division of f_{HOCO} when f_{HOCO} is set to 48 MHz, and the same clock frequency as f_{HOCO} when f_{HOCO} is set to 24 MHz or less. When supplying 48 MHz to timer RD, set f_{CLK} to f_{IH}.

5.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

Caution Which registers and bits are included depends on the product. Be sure to set registers and bits that are not mounted in a product to their initial values.

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 2 Format of Clock operation mode control register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ f_X ≤ 10 MHz
1	10 MHz < f_X ≤ 20 MHz

Caution 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.

Caution 2. After reset release, set the CMC register before X1 oscillation is started as set by the clock operation status control register (CSC).

Caution 3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.

Caution 4. Specify the settings for the AMPH bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

Caution 5. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Caution 6. Be sure to clear bits 1 to 5 to 0.

Remark f_X: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.
The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 5 - 3 Format of System clock control register (CKC)

Address: FFFA4H After reset: 00H R/W$^{\text{Note}}$

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{IH})
1	High-speed system clock (f_{MX})

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Note Bit 5 is read-only.

Remark f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.) $^{\text{Note}}$
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency

Note f_{IH} is controlled by hardware to be set to two frequency division of f_{HOCO} when f_{HOCO} is set to 48 MHz, and the same clock frequency as f_{HOCO} when f_{HOCO} is set to 24 MHz or less. When supplying 48 MHz to timer RD, set f_{CLK} to f_{IH}.

Caution 1. Be sure to set bits 0 to 3, 6, and 7 of the CKC register to 0.

Caution 2. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the 12-bit interval timer, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

Caution 3. When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, and high-speed on-chip oscillator clock (except the low-speed on-chip oscillator clock).

The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to C0H.

Figure 5 - 4 Format of Clock operation status control register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
CSC	MSTOP	1	0	0	0	0	0	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator operating
1	High-speed on-chip oscillator stopped

Caution 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.

Caution 2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

Caution 3. To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

Caution 4. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 5. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5 - 2. Check the condition before stopping clock before stopping the clock.

Table 5 - 2 Condition Before Stopping Clock Oscillation and Flag Setting

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (MCS = 0)	MSTOP = 1
External main system clock		
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock. (MCS = 1)	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case,

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.
The generation of reset signal, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 5 - 5 Format of Oscillation stabilization time counter status register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.2 ms min.	13.1 ms min.

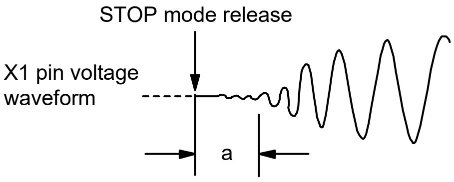
Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time.

When the X1 clock is selected by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

When the CPU clock is switched from the high-speed on-chip oscillator clock to the X1 clock, or when STOP mode is entered while the high-speed on-chip oscillator is used as the CPU clock and the X1 clock is also oscillating, and then STOP mode is released, use the oscillation stabilization time counter status register (OSTC) to confirm that the oscillation stabilization time has elapsed.

The OSTS register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the OSTS register to 07H.

Figure 5 - 6 Format of Oscillation stabilization time select register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

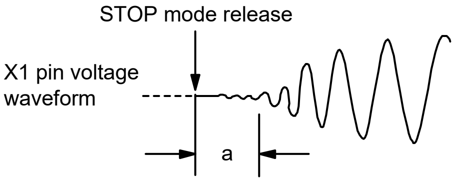
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

5.3.6 Peripheral enable registers 0, 1 (PER0, PER1)

These registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- 12-bit interval timer
- A/D converter
- Serial array unit 0
- Timer array unit 0
- Timer RD
- Timer RJ
- Comparator 0
- Comparator 1
- Programmable gain amplifier (PGA)
- PWM option unit

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 5 - 7 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PER0	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 0 cannot be written. • The serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 0 can be read and written.

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. • SFR used by timer array unit 0 cannot be written. • Timer array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by timer array unit 0 can be read and written.

Caution Be sure to set bits 1, 3, 4, 6, 7 of the PER0 register to 0.

Figure 5 - 8 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
PER1	TMKAEN[Note 1]	PWMOPEN	OACMPEN	TRD0EN[Note 2]	0	0	0	TRJ0EN

TMKAEN[Note 1]	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by 12-bit interval timer cannot be written. • 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by 12-bit interval timer can be read and written.

PWMOPEN	Control of PWM option unit input clock supply
0	Stops input clock supply. • SFR used by PWM option unit cannot be written. • PWM option unit is in the reset status.
1	Enables input clock supply. • SFR used by PWM option unit can be read and written.

OACMPEN	Control of input clock supply for comparators 0 and 1 and the programmable gain amplifier
0	Stops input clock supply. • SFR used by comparators 0 and 1 and the programmable gain amplifier cannot be written. • Comparators 0 and 1 and the programmable gain amplifier are in the reset state.
1	Enables input clock supply. • SFR used by comparators 0 and 1 and the programmable gain amplifier cannot be read and written.

TRD0EN[Note 2]	Control of timer RD input clock supply
0	Stops input clock supply. • SFR used by timer RD cannot be written. • Timer RD is in the reset status.
1	Enables input clock supply. • SFR used by timer RD can be read and written.

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

Note 1. When using the 12-bit interval timer, set bit 4 (WUTMMCK0) in the operation speed mode control register (OSMC) to 1 and start oscillating the low-speed on-chip oscillator clock. Then wait until the low-speed on-chip oscillator stabilizes before setting the TMKAEN bit to 1.

Note 2. When FRQSEL4 = 1 in the user option byte (000C2H), set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Caution **Be sure to set bits 1 to 3 of the PER1 register to 0.**

5.3.7 Operation speed mode control register (OSMC)

The OSMC register can be used to select the low-speed on-chip oscillator as the 12-bit interval timer operating clock or the timer RJ count source.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 9 Format of Operation speed mode control register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0 Notes 1, 2	0	0	0	0

WUTMMCK0 Notes 1, 2	Selection of low-speed on-chip oscillator as 12-bit interval timer operating clock or timer RJ count source
0	• The low-speed on-chip oscillator cannot be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator cannot be selected as the timer RJ count source.
1	• The low-speed on-chip oscillator can be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator can be selected as the timer RJ count source.

Note 1. When using the 12-bit interval timer, be sure to set the WUTMMCK0 bit to 1.

Note 2. When the 12-bit interval timer is operating, do not set the WUTMMCK0 bit to 0.

Caution Be sure to clear bits 0 to 3 and bits 5 to 7 to 0.

5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV)

The frequency of the high-speed on-chip oscillator which is set by an option byte (000C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL4 and FRQSEL3 bits of the option byte (000C2H).

The HOCODIV register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 5 - 10 Format of High-speed on-chip oscillator frequency select register (HOCODIV)

Address: F00A8H After reset: the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24 MHz f_{HOCO} = 48 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited
Other than above			Setting prohibited		

Caution 1. Both before and after the frequency change, set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H).

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 to 8 MHz	2.7 to 5.5 V
1	1	HS (high-speed main) mode	1 to 24 MHz	2.7 to 5.5 V
Setting prohibited		Other than above		

Caution 2. Set the HOCODIV register while the high-speed on-chip oscillator clock (f_{IH}) is selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 3. After the frequency has been changed using the HOCODIV register and the following transition time has elapsed, the frequency is switched.

- **Operation for up to three clocks at the pre-change frequency**
- **The CPU/peripheral hardware clock waits for a maximum of 3 clocks at the frequency after the frequency has been changed.**

5.3.9 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input, and so on, the accuracy can be adjusted.

The HIOTRM register can be set by an 8-bit memory manipulation instruction.

Caution The frequency will vary if the temperature and V_{DD} pin voltage change after accuracy adjustment. When the temperature and V_{DD} voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5 - 11 Format of High-speed on-chip oscillator trimming register (HIOTRM)

Address: F00A0H After reset: Note R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	↑
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
⋮						
1	1	1	1	1	0	↓
1	1	1	1	1	1	Maximum speed

Note The value after reset is the value adjusted at shipment.

Remark 1. The HIOTRM register can be used to adjust the high-speed on-chip oscillator clock to an accuracy within about 0.05%.

Remark 2. For the usage example of the HIOTRM register, see the application note for RL78 MCU Series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464).

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

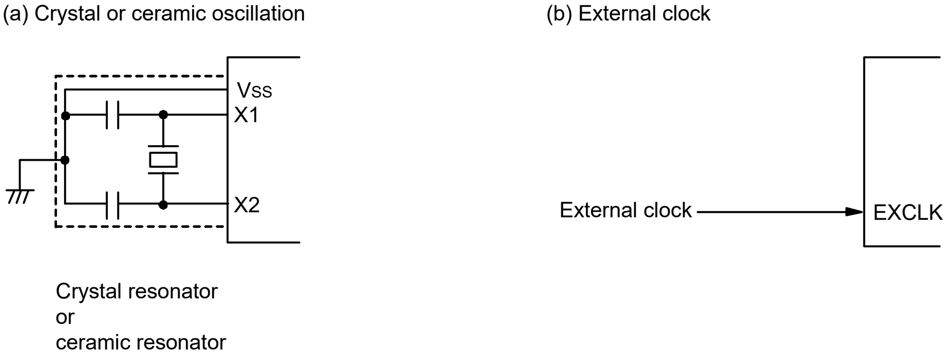
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2 - 3 Connection of Unused Pins**.

Figure 5 - 12 shows an example of the external circuit of the X1 oscillator.

Figure 5 - 12 Example of External Circuit of X1 Oscillator

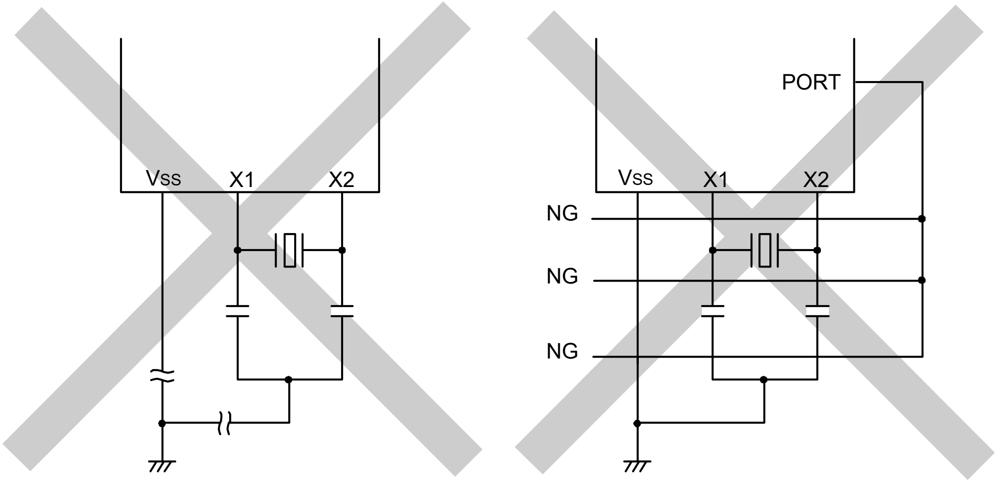


Note When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5 - 12 to avoid an adverse effect from wiring capacitance.

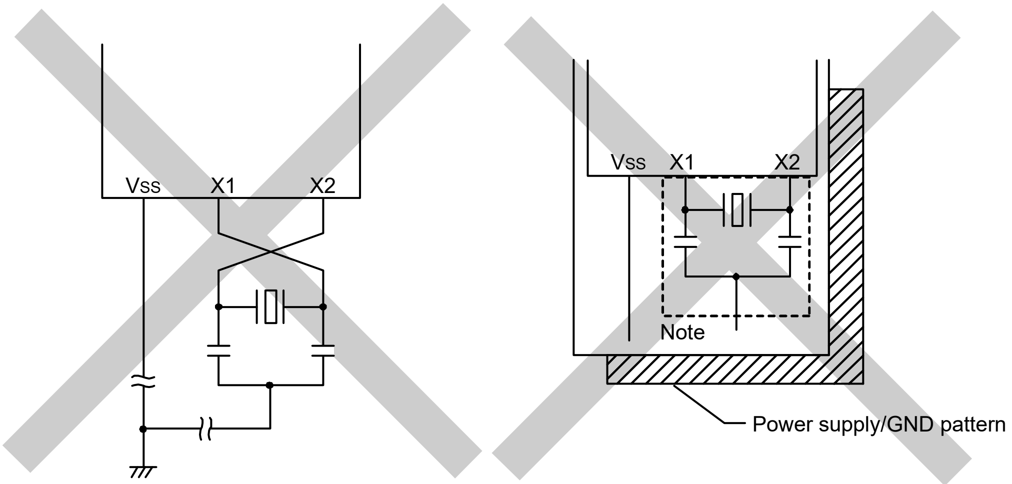
- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Figure 5 - 13 shows examples of incorrect resonator connection.

Figure 5 - 13 Examples of Incorrect Resonator Connection (1/2)


(a) Too long wiring

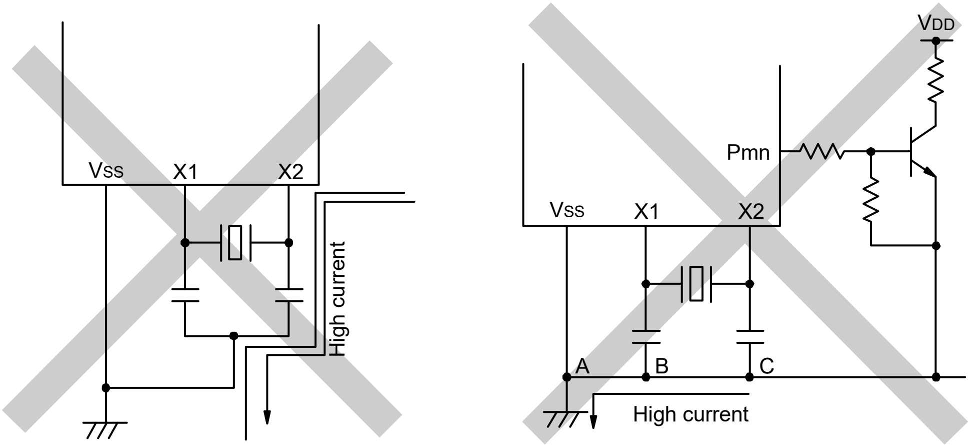
(b) Crossed signal line


(c) The X1 and X2 signal line wires cross.

(d) A power supply/GND pattern exists under the X1 and X2 wires.

Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

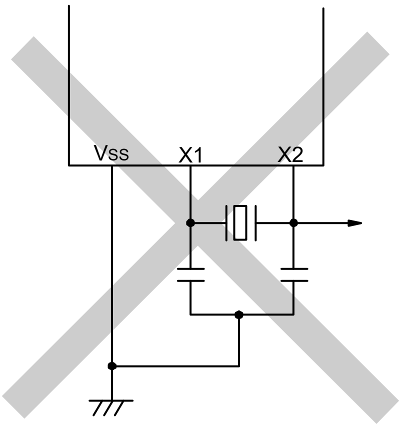
Figure 5 - 14 Examples of Incorrect Resonator Connection (2/2)



(e) Wiring near high alternating current

(f) Current flowing through ground line of oscillator
(potential at points A, B, and C fluctuates)

(g) Signals are fetched



5.4.2 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G1G. The frequency can be selected from among 48, 24, 16, 12, 8, 4, or 1 MHz by using the option byte (000C2H). When 48 MHz is selected, the two frequency division of the selected clock is supplied to CPU clock. Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G1G.

The low-speed on-chip oscillator clock is used only as the watchdog timer, 12-bit interval timer, and timer RJ clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. Note that only when the watchdog timer is operating and the WUTMMCK0 bit is 0, oscillation of the low-speed on-chip oscillator will stop while the WDSTBYON bit is 0 and operation is in the HALT, STOP, or SNOOZE mode. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

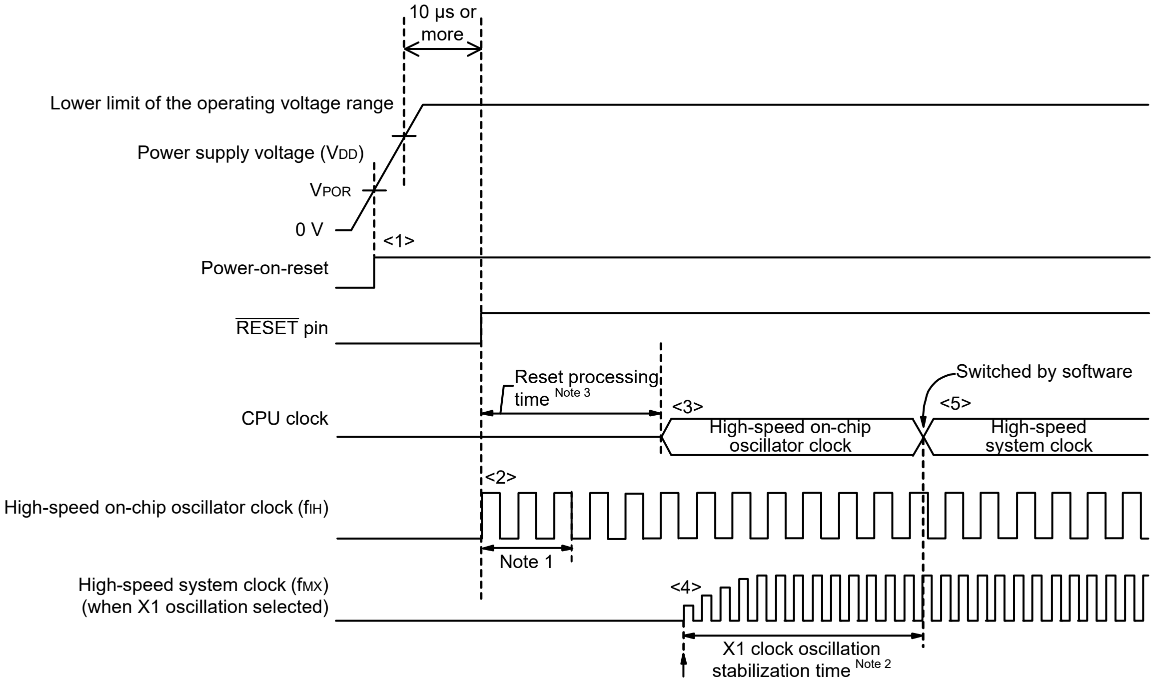
5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5 - 1**).

- Main system clock f_{MAIN}
    - High-speed system clock f_{MX}
      X1 clock f_X
      External main system clock f_{EX}
    - High-speed on-chip oscillator clock f_{IH}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/G1G.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 5 - 15.

Figure 5 - 15 Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit.
Note that the reset state is maintained after a reset by the voltage detection circuit or an external reset until the voltage reaches the range of operating voltage described in **29.5 AC Characteristics** (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 via software (see **5.6.2 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1, wait for the clock oscillation to stabilize, and then set switching via software (see **5.6.2 Example of setting X1 oscillation clock**).

Note 1. The reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

Note 3. For the reset processing time, see **CHAPTER 20 POWER-ON-RESET CIRCUIT**.

Caution **It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.**

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 48, 24, 16, 12, 8, 4, and 1 MHz by using FRQSEL0 to FRQSEL4 of the option byte (000C2H). In addition, Oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	Setting of flash operation mode	
1	0	LS (low speed main) mode	V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 8 MHz
1	1	HS (high speed main) mode	V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited	

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of high-speed on-chip oscillation circuit	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	1	0	1	1	4 MHz	4 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24 MHz f_{HOCO} = 48 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited
Other than above			Setting prohibited		

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases where the fx is equal to or more than 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	0	0	0	0	0	AMPH 1

AMPH bit: Set this bit to 0 if the X1 oscillation clock is 10 MHz or less.

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.
Example: Setting values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2 0	OSTS1 1	OSTS0 0

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	1	0	0	0	0	0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.
Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	0	0	MCS 0	MCM0 1	0	0	0	0

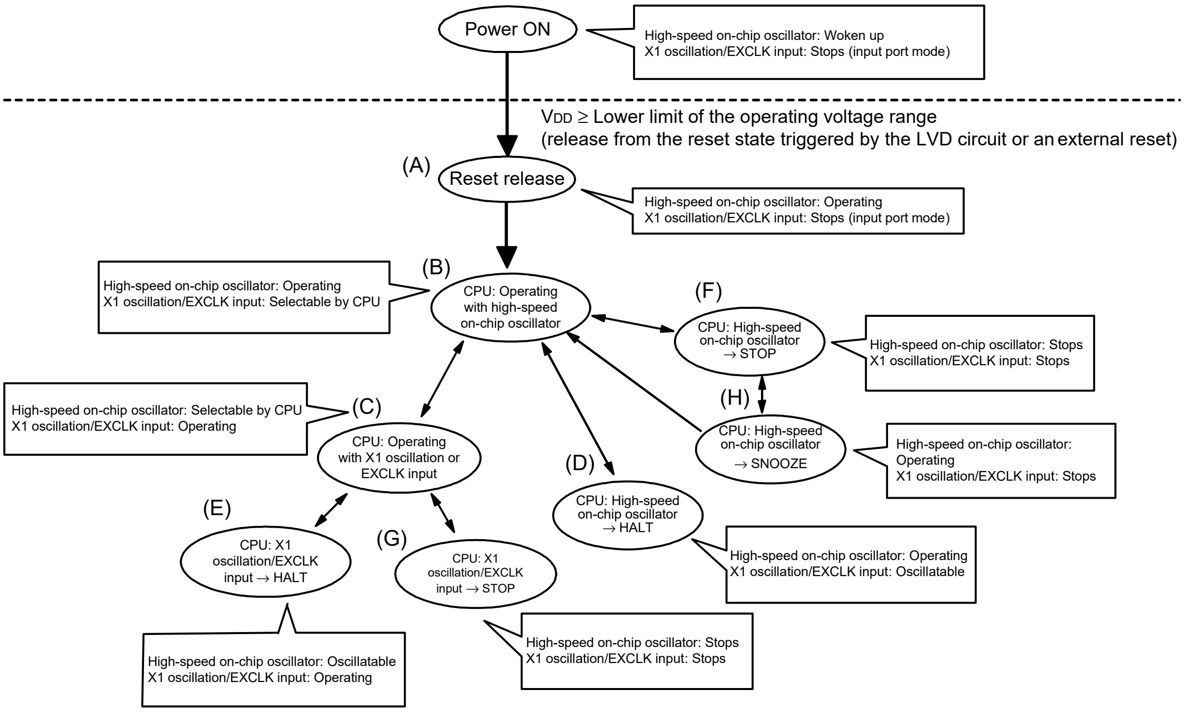
Caution Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Value of option byte (000C2H)		Flash operation mode	Operating frequency range	Operating voltage range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 MHz to 8 MHz	2.7 V to 5.5 V
1	1	HS (high-speed main) mode	1 MHz to 24 MHz	2.7 V to 5.5 V
Other than above		Setting prohibited		

5.6.3 CPU clock status transition diagram

Figure 5 - 16 shows the CPU clock status transition diagram of this product.

Figure 5 - 16 CPU Clock Status Transition Diagram



Tables 5 - 3 to 5 - 5 show transition of the CPU clock and examples of setting the SFR registers.

Table 5 - 3 CPU Clock Transition and SFR Register Setting Examples (1/3)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)
(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	Note 2	0	Need not be checked	1

Note 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

Note 2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution **Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).**

Remark 1. ×: don’t care

Remark 2. (A) to (H) in Tables 5 - 3 to 5 - 5 correspond to (A) to (H) in Figure 5 - 16.

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (2/3)

(3) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	Note 2	0	Need not be checked	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Note 1. The clock operation mode control register (CMC) can be changed only once after reset release. This setting is not necessary if it has already been set.

Note 2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution **Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).**

Remark 1. ×: don't care

Remark 2. (A) to (H) in Tables 5 - 3 to 5 - 5 correspond to (A) to (H) in Figure 5 - 16.

(4) CPU clock changing from high-speed system clock (C) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	Note	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

Note When FRQSEL4 = 0: 18 μs to 65 μs
When FRQSEL4 = 1: 18 μs to 135 μs

Remark 1. (A) to (H) in Tables 5 - 3 to 5 - 5 correspond to (A) to (H) in Figure 5 - 16.

Remark 2. The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

Table 5 - 5 CPU Clock Transition and SFR Register Setting Examples (3/3)

(5) • HALT mode (D) set while CPU is operating with high-speed on-chip oscillator clock (B)
• HALT mode (E) set while CPU is operating with high-speed system clock (C)

Status Transition	Setting
(B) → (D) (C) → (E)	Executing HALT instruction

Remark (A) to (H) in Tables 5 - 3 to 5 - 5 correspond to (A) to (H) in Figure 5 - 16.

(6) • STOP mode (F) set while CPU is operating with high-speed on-chip oscillator clock (B)
• STOP mode (G) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (F)		Stopping peripheral functions that cannot operate in STOP mode	—	Executing STOP instruction
(C) → (G)	In X1 oscillation		Sets the OSTS register	
	External main system clock		—	

(7) CPU changing from STOP mode (F) to SNOOZE mode (H)
For details about the setting for switching from the STOP mode to the SNOOZE mode, see **12.8 SNOOZE Mode Function**, **14.5.7 SNOOZE mode function**, and **14.7.3 SNOOZE mode function**.

Remark (A) to (H) in Tables 5 - 3 to 5 - 5 correspond to (A) to (H) in Figure 5 - 16.

5.6.4 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5 - 6 Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible	—
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, external main system clock input can be disabled (MSTOP = 1).
	X1 clock	• Transition not possible	—

5.6.5 Time required for switchover of CPU clock and main system clock

By setting bits 4 (MCM0) of the system clock control register (CKC) and main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Tables 5 - 7** and **5 - 8**).

Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 5 - 7 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See **Table 5 - 8**

Table 5 - 8 Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 (f_{MAIN} = f_{IH})	1 (f_{MAIN} = f_{MX})
0 (f_{MAIN} = f_{IH})	$f_{MX} \geq f_{IH}$		2 clock
	$f_{MX} < f_{IH}$		1 + f_{IH}/f_{MX} clock
1 (f_{MAIN} = f_{IH})	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	2 clock	

Remark 1. The number of clocks listed in Table 5 - 8 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Table 5 - 8 by rounding up the number after the decimal position.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

$2f_{MX}/f_{IH} = 2\ (10/8) = 2.5 \rightarrow 3$ clocks

5.6.6 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Check the condition before stopping clock before stopping the clock.

Table 5 - 9 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		

5.7 Resonator and Oscillator Constants

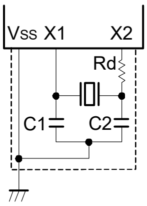
The resonators for which the operation is verified and their oscillator constants are shown below.

Caution 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

Caution 2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5 - 17 Example of External Circuit

(a) X1 oscillation



(1) X1 oscillation

As of Apr 2024

Manufacturer	Resonator	SMD/ Lead	Part Number	Frequency (MHz)	Flash Operation Mode$^{\text{Note 1}}$	Circuit Constants$^{\text{Note 2}}$ (Reference)			Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd.$^{\text{Note 3}}$	Ceramic resonator	SMD	CSTCC2M00G56-R0	2.0	LS	(47)	(47)	0	2.7	5.5
		SMD	CSTCR4M00G55-R0	4.0		(39)	(39)	0		
		Lead	CSTLS4M00G53-B0			(15)	(15)	0		
		SMD	CSACN4M00G530000R0			15	15	0		
		SMD	CSTCE8M00G52-R0	8.0		(10)	(10)	0		
		Lead	CSTLS8M00G53-B0			(15)	(15)	0		
		SMD	CSTCC2M00G56-R0	2.0	HS	(47)	(47)	0		
		SMD	CSTCR4M00G55-R0	4.0		(39)	(39)	0		
		Lead	CSTLS4M00G53-B0			(15)	(15)	0		
		SMD	CSACN4M00G530000R0			15	15	0		
		SMD	CSTCE8M00G52-R0	8.0		(10)	(10)	0		
		Lead	CSTLS8M00G53-B0			(15)	(15)	0		
		SMD	CSACM8M00G530005R0			10	10	0		
		SMD	CSTCE10M0G52-R0	10.0		(10)	(10)	0		
		Lead	CSTLS10M0G53-B0			(15)	(15)	0		
		SMD	CSACM10M0G530005R0			10	10	0		
		SMD	CSTCE12M0G52-R0	12.0		(10)	(10)	0		
		SMD	CSACM12M0G530005R0			10	10	0		
		SMD	CSTCE16M0V53-R0	16.0		(15)	(15)	0		
		Lead	CSTLS16M0X51-B0			(5)	(5)	0		
		SMD	CSTCE20M0V51-R0	20.0		(5)	(5)	0		
		Lead	CSTLS20M0X51-B0			(5)	(5)	0		
Kyocera Crystal Device Co., Ltd.$^{\text{Note 4}}$	Crystal oscillator	SMD	CX8045GB04000D0PPS01	4.0	LS	7	7	0	2.7	5.5
		SMD	CX8045GB08000D0PPS01	8.0		6	6	0		
		SMD	CX8045GB04000D0PPS01	4.0	HS	7	7	0		
		SMD	CX8045GB08000D0PPS01	8.0		6	6	0		
		SMD	CX3225SB12000D0PPSC1	12.0		6	6	0		
		SMD	CX3225SB16000D0PPSC1	16.0		6	6	0		
		SMD	CX3225SB20000D0PPSC1	20.0		6	6	0		
Nihon Dempa Kogyo Co., Ltd.$^{\text{Note 5}}$	Crystal oscillator	SMD	NX3225GA-20.000M-CHP-CRG-12	20.0	HS	3	3	0	2.7	5.5

Note 1. Set the flash operation mode by using the CMODE1 and CMODE0 bits of the option byte (000C2H).

Note 2. Values in parentheses in the C1 and C2 columns indicate an internal capacitance.

Note 3. When using this resonator, for details about the matching, contact Murata Manufacturing Co., Ltd. (http://www.murata.com/index.html).

Note 4. When using this resonator, for details about the matching, contact Kyocera Crystal Device Co., Ltd. (http://www.kyocera-crystal.jp/eng/index.html, http://global.kyocera.com).

Note 5. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (High-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
LS (Low-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz

CHAPTER 6 TIMER ARRAY UNIT

The number of units or channels of the timer array unit differs, depending on the product.

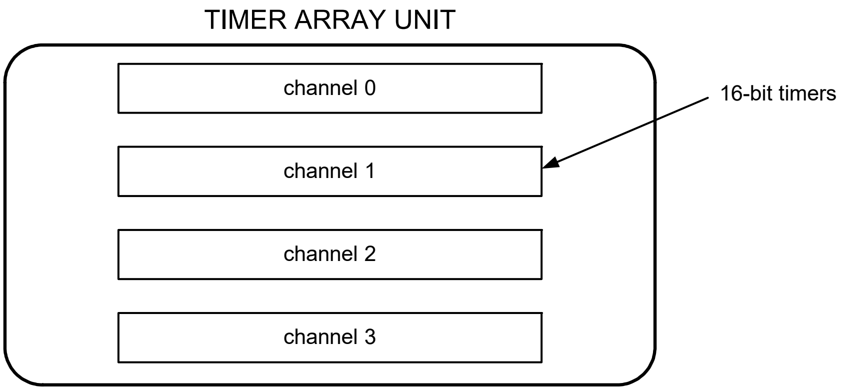
Units	Channels	30, 32, 44-pin
Unit 0	Channel 0	√
	Channel 1	√
	Channel 2	√
	Channel 3	√

Caution 1. The presence or absence of timer I/O pins depends on the product. See Table 6 - 2 Timer I/O Pins Provided in Each Product for details.

Caution 2. Most of the following descriptions in this chapter use the 44-pin products as an example.

The timer array unit has four 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **6.8.1**) • Square wave output (→ refer to **6.8.1**) • External event counter (→ refer to **6.8.2**) • Divider function Note (→ refer to **6.8.3**) • Input pulse interval measurement (→ refer to **6.8.4**) • Measurement of high-/low-level width of input signal (→ refer to **6.8.5**) • Delay counter (→ refer to **6.8.6**)	• One-shot pulse output (→ refer to **6.9.1**) • PWM output (→ refer to **6.9.2**) • Multiple PWM output (→ refer to **6.9.3**)

Note Only channel 0 of timer array unit 0.

It is possible to use the 16-bit timer of channels 1 and 3 of the unit 0 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

6.1 Functions of Timer Array Unit

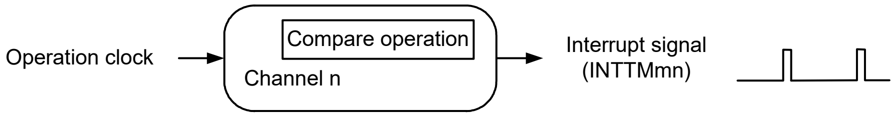
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

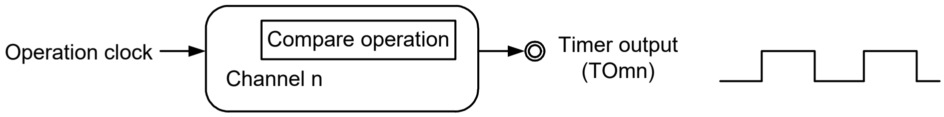
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



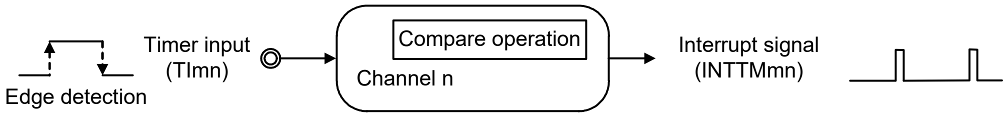
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



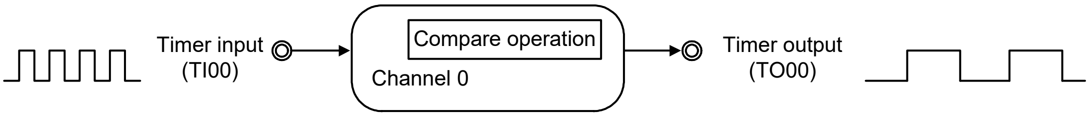
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



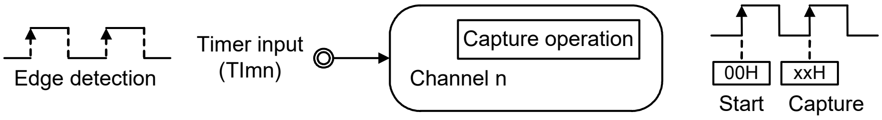
(4) Divider function (channel 0 only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TOm0).



(5) Input pulse interval measurement

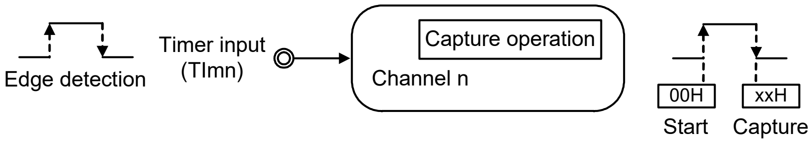
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(**Remark** is listed on the next page.)

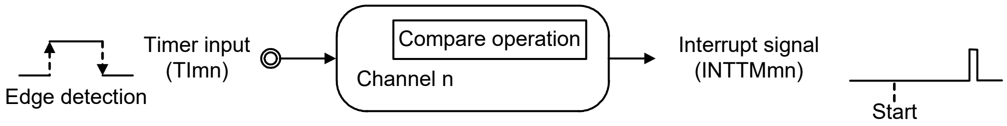
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

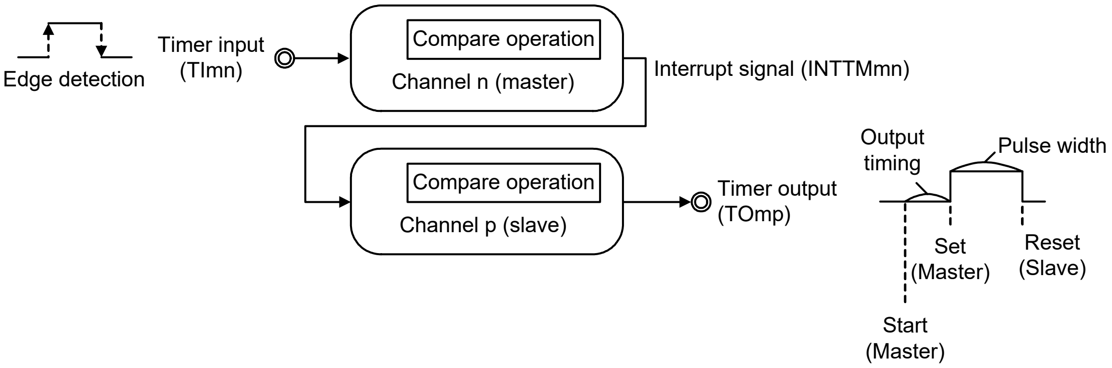
Remark 2. The presence or absence of timer I/O pins of channel 0 to 3 depends on the product. See **Table 6 - 2 Timer I/O Pins Provided in Each Product**.

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

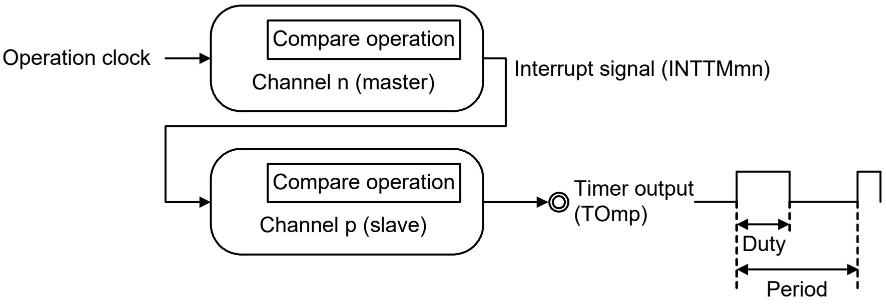
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



(2) PWM (Pulse Width Modulation) output

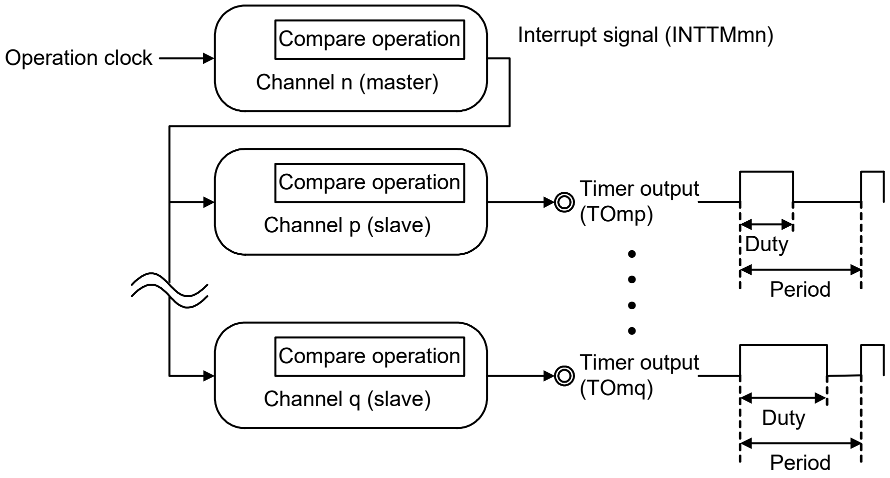
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(**Caution** and **Remark** are listed on the next page.)

(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to three types of PWM signals that have a specific period and a specified duty factor can be generated.



Caution **The following rules apply when using multiple channels simultaneously.**
For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3),
p, q: Slave channel number (n < p < q ≤ 3)

6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution **There are several rules for using 8-bit timer operation function.**
For details, see 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI03 Note 1
Timer output	TO00 to TO03 Note 1, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx) Note 2 • Port mode register (PMxx) Note 2 • Port register (Pxx) Note 2

Note 1. See **Table 6 - 2 Timer I/O Pins Provided in Each Product** for details.

Note 2. The port mode control register (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **4.5 Register Setting for Used Port and Alternate Functions**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

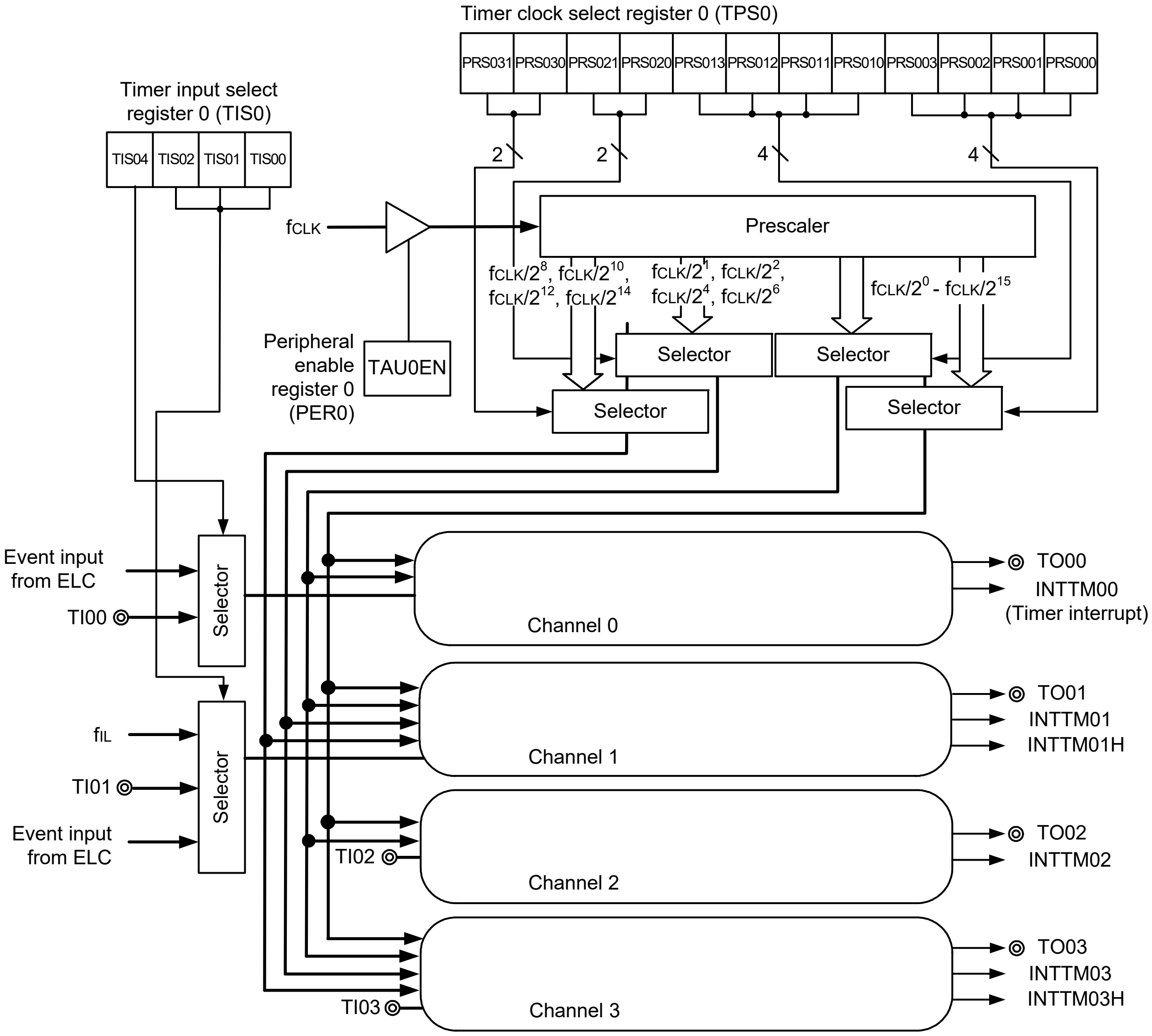
Table 6 - 2 Timer I/O Pins Provided in Each Product

Timer array unit channels		I/O Pins of Each Product
		30, 32, 44-pin
Unit 0	Channel 0	P00/TI00, P01/TO00
	Channel 1	P16/TI01/TO01
	Channel 2	P17/TI02/TO02
	Channel 3	P31/TI03/TO03

Remark When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Figures 6 - 1 to 6 - 5 show the block diagrams of the timer array unit.

Figure 6 - 1 Entire Configuration of Timer Array Unit 0



Remark fIL: Low-speed on-chip oscillator clock frequency

Figure 6 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

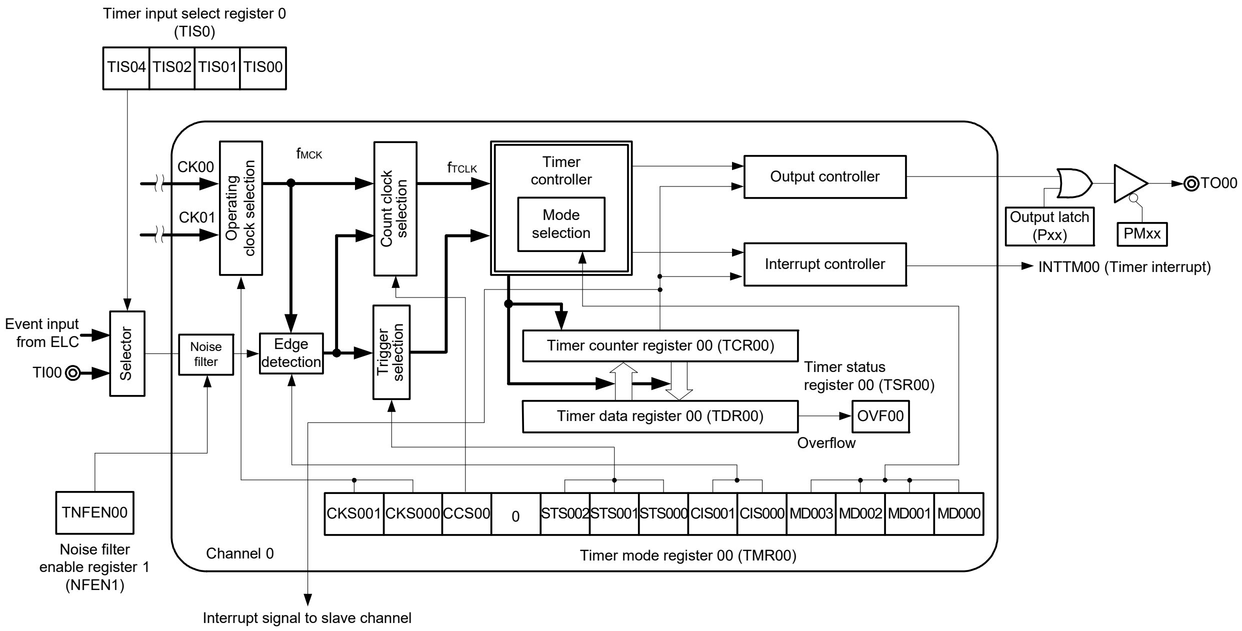


Figure 6 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

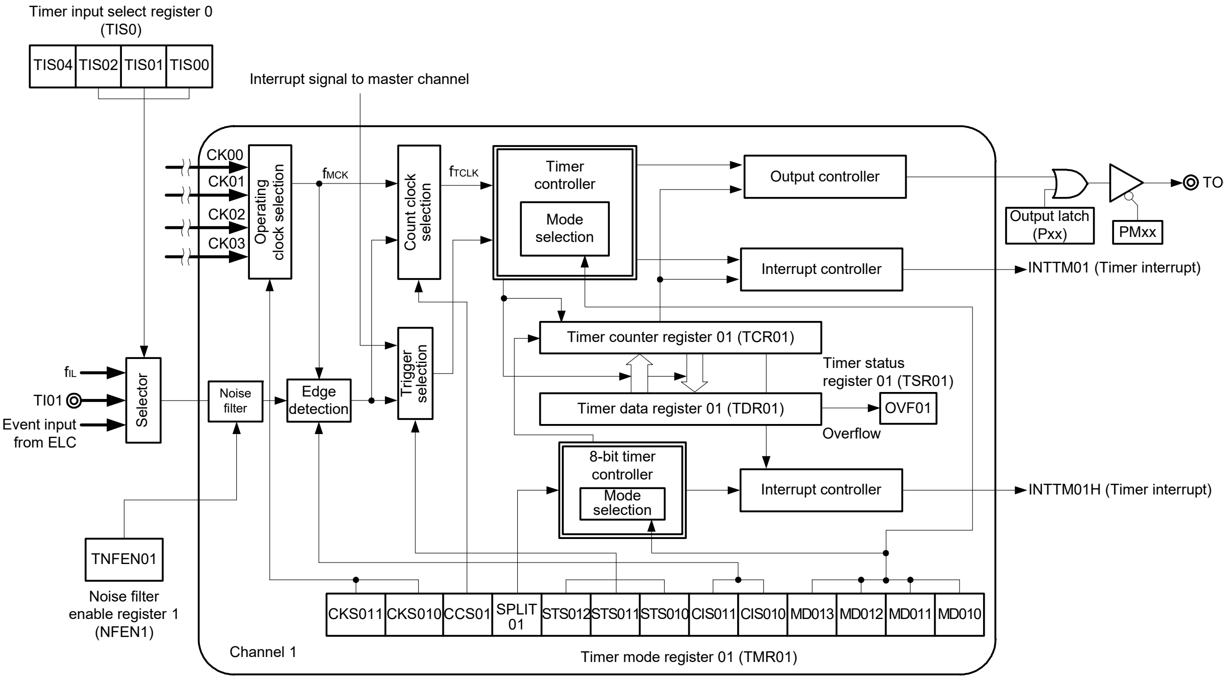


Figure 6 - 4 Internal Block Diagram of Channel 2 of Timer Array Unit 0

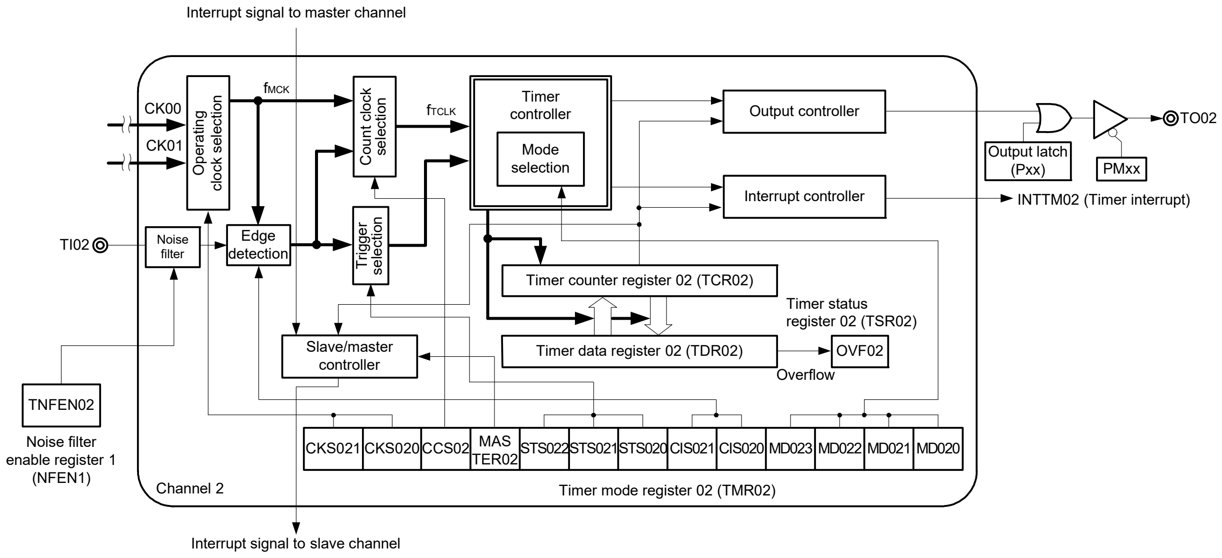
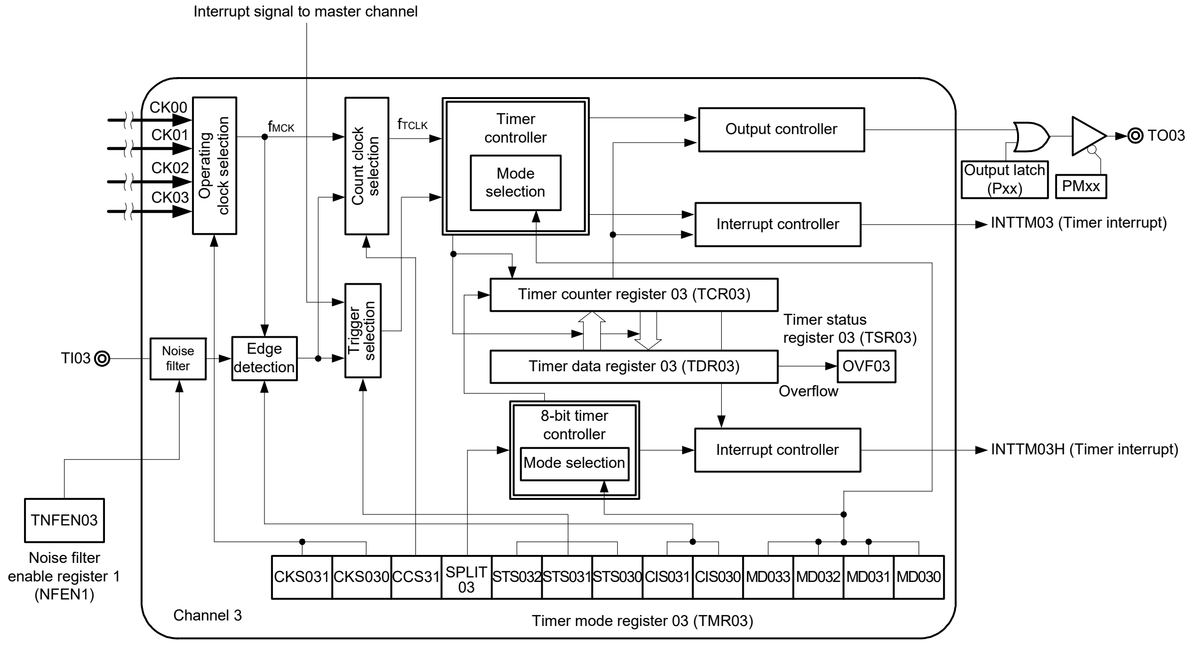


Figure 6 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0



6.2.1 Timer count register mn (TCRmn)

The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

Figure 6 - 6 Format of Timer count register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03)　　After reset: FFFFH　　R

	F0181H (TCR00)								F0180H (TCR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The count value can be read by reading timer count register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 6 - 3 Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer Count Register mn (TCRmn) Read Value Note			
		Value if the operation mode was changed after releasing reset	Value if the operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	—
Capture mode	Count up	0000H	Value if stop	Undefined	—
Event counter mode	Count down	FFFFH	Value if stop	Undefined	—
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6 - 7 Format of Timer data register mn (TDRmn) (n = 0, 2)

Figure 6 - 8 Format of Timer data register mn (TDRmn) (n = 1, 3)

(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port register (Pxx)

Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 9 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PER0	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array 0 unit input clock
0	Stops supply of input clock. • SFR used by the timer array unit 0 cannot be written. • The timer array unit 0 is in the reset status.
1	Supplies input clock. • SFR used by the timer array unit 0 can be read/written.

Caution 1. When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 0, writing to a control register of timer array unit is ignored, and all read values are default values (except for the timer input select register 0 (TIS0), noise filter enable register 1 (NFEN1), port mode registers 0, 12, 14 (PM0, PM12, PM14), and port registers 0, 1, 3 (P0, P1, P3)).

- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 4, 6, 7

6.3.2 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel from external prescaler. CKm1 is selected by using bits 7 to 4 of the TPSm register, and CKm0 is selected by using bits 3 to 0. In addition, for channel 1 and 3, CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12.
Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 3):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).
If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 6 - 10 Format of Timer clock select register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Note (k = 0, 1)				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution 1. Be sure to clear bits 15, 14, 11, 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (m = 0 to 3), interrupt requests output from timer array units are not detected.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPS0 register which becomes high level for one period of f_{CLK} from its rising edge. For details, see **6.5.1 Count clock (f_{TCLK})**.

Figure 6 - 11 Format of Timer clock select register m (TPSm) (2/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS m21	PRS m20	Selection of operation clock (CKm2) [Note]				
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	$f_{\mathrm{CLK}}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	1	$f_{\mathrm{CLK}}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{\mathrm{CLK}}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	1	$f_{\mathrm{CLK}}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz

PRS m31	PRS m30	Selection of operation clock (CKm3) [Note]				
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	$f_{\mathrm{CLK}}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
0	1	$f_{\mathrm{CLK}}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	$f_{\mathrm{CLK}}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	$f_{\mathrm{CLK}}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, 10 to “0”.

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6 - 4 can be achieved by using the interval timer function.

Table 6 - 4 Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time[Note] (f_{CLK} = 20 MHz)			
		16 μs	160 μs	1.6 ms	16 ms
CKm2	$f_{\mathrm{CLK}}/2$	√	—	—	—
	$f_{\mathrm{CLK}}/2^2$	√	—	—	—
	$f_{\mathrm{CLK}}/2^4$	√	√	—	—
	$f_{\mathrm{CLK}}/2^6$	√	√	—	—
CKm3	$f_{\mathrm{CLK}}/2^8$	—	√	√	—
	$f_{\mathrm{CLK}}/2^{10}$	—	√	√	—
	$f_{\mathrm{CLK}}/2^{12}$	—	—	√	√
	$f_{\mathrm{CLK}}/2^{14}$	—	—	√	√

Note The margin is within 5%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of $f_{\mathrm{CLK}}/2^r$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK})**.

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.
TMRm2: MASTERmn bit (n = 2)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0: Fixed to 0

Figure 6 - 12 Format of Timer mode register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 0, valid edge of input signal selected by TIS0 In channel 1, valid edge of input signal selected by TIS0 In channel 3, valid edge of input signal selected by ISC
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Caution 1. Be sure to clear bits 13, 5, and 4 to “0”.

Caution 2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 13 Format of Timer mode register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only channel 2 can be set as a master channel (MASTERmn = 1). Channel 0 is fixed to 0 (channel 0 always operates as master regardless of the bit setting, because it is the highest channel). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 14 Format of Timer mode register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CISmn1	CISmn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

MDmn3	MDmn2	MDmn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation of the MDmn0 bit varies depending on each operation mode (see **Figure 6 - 15**).					

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 15 Format of Timer mode register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see **Figure 6 - 14**))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated.
Other than above		Setting prohibited

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Note 2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

Note 3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialized, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.4 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 6 - 5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6 - 16 Format of Timer status register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Table 6 - 5 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	— (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 17 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H (TE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	0	0	0	0	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 18 Format of Timer channel start register m (TSm)

Address: F01B2H, F01B3H (TS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	0	0	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6 - 6** in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

(Cautions and Remarks are listed on the next page.)

Caution 1. Be sure to clear bits 15 to 12, 10, 8 to 4 to “0”

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.

When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})

When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.7 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TTHm1, TTHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 19 Format of Timer channel stop register m (TTm)

Address: F01B4H, F01B5H (TT0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 to 4 of the TTm register to “0”.

Remark 1. When the TTm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 0 and 1 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 20 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

Caution 1. Input 1/f_{MCK} + 10 ns or more for the high-level and low-level widths of the timer input to be selected.

Caution 2. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), set the noise filter of the corresponding pin to “OFF” using noise filter enable register 1 (NFEN1), and select f_{CLK} using timer clock select register 0 (TPS0).

6.3.9 Timer output enable register m (TOEm)

The TOEm register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOmn bit of timer output register m (TOm) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOmn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 21 Format of Timer output enable register m (TOEm)

Address: F01BAH, F01BBH (TOE0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	0	0	0	0	TOEm 3	TOEm 2	TOEm 1	TOEm 0

TOE mn	Timer output enable/disable of channel n
0	Timer output is disabled. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution Be sure to clear bits 15 to 4 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.10 Timer output register m (TOm)

The TOm register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

The TOmn bit oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0).

When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P00/TI00, P01/TO00, P16/TI01/TO01, P17/TI02/TO02, P31/TI03/TO03 pins as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 22 Format of Timer output register m (TOm)

Address: F01B8H, F01B9H (TO0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	0	0	0	0	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution **Be sure to clear bits 15 to 4 to “0”.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.11 Timer output level register m (TOLm)

The TOLm register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 23 Format of Timer output level register m (TOLm)

Address: F01BCH, F01BDH (TOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOLm 3	TOLm 2	TOLm 1	0

TOL mn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution **Be sure to clear bits 15 to 4, and 0 to “0”.**

Remark 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.12 Timer output mode register m (TOMm)

The TOMm register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 24 Format of Timer output mode register m (TOMm)

Address: F01BEH, F01BFH (TOM0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOMm 3	TOMm 2	TOMm 1	0

TOM mn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear bits 15 to 4, and 0 to “0”.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0, p = 1, 2, 3
n = 2, p = 3
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**)

6.3.13 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.
Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.
When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{MCK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}) Note.
The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

Figure 6 - 25 Format of Noise filter enable register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

6.3.14 Port mode registers 0, 1, 3 (PM0, PM1, PM3)

These registers set input/output of ports 0, 1, 3 in 1-bit units.

When using the ports (such as P01/TO00 and P17/TO02/TI02) to be shared with the timer output pin for timer output, set the port mode control register (PMCxx) bit, port mode register (PMxx) bit, and port register (Pxx) bit corresponding to each port to 0.

Example When using P17/TO02/TI02 for timer output.
Set the PM17 bit of port mode register 1 to 0.
Set the P17 bit of port register 1 to 0.

When using the ports (such as P00/TI00 and P17/TO02/TI02) to be shared with the timer output pin for timer input, set the port mode control register (PMCxx) bit and port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example When using P17/TO02/TI02 for timer input.
Set the PM17 bit of port mode register 1 to 1.
P17 bit of port register may be 0 or 1.

The PM0, PM1, PM3 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Figure 6 - 26 Format of Port mode registers 0, 1, 3 (PM0, PM1, PM3) (44-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	1	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2) can be set as a master channel.

(2) Any channel, except channel 0, can be set as a slave channel.

(3) The slave channel must be lower than the master channel.

Example If channel 0 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.

(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.

Example If channels 0 and 2 are set as master channels, channels 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channel of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.

(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.

(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.

(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.

(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.

(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.

(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.

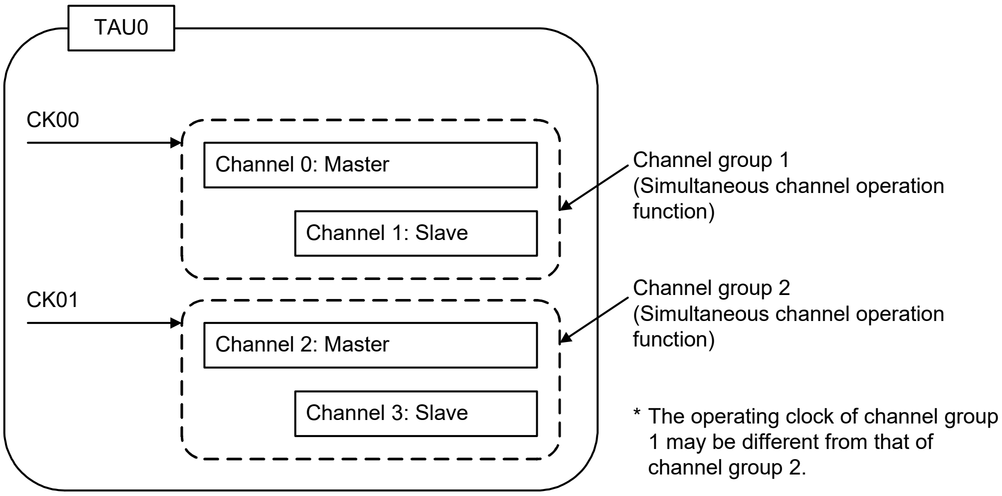
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.

(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

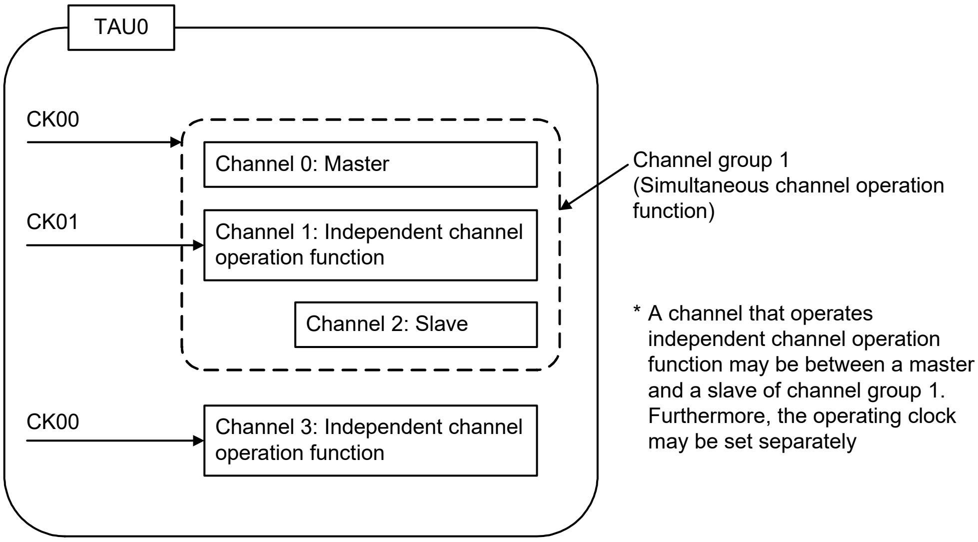
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Example 1



* The operating clock of channel group 1 may be different from that of channel group 2.

Example 2



* A channel that operates independent channel operation function may be between a master and a slave of channel group 1. Furthermore, the operating clock may be set separately

6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.

(2) When using 8-bit timers, set the SPLITmn bit of timer mode register mn (TMRmn) to 1.

(3) The higher 8 bits can be operated as the interval timer function.

(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).

(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.

(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.

(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:

- Interval timer function/square wave function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.

(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.

(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3)

6.5 Operation Timing of Counter

6.5.1 Count clock (fTCLK)

The count clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

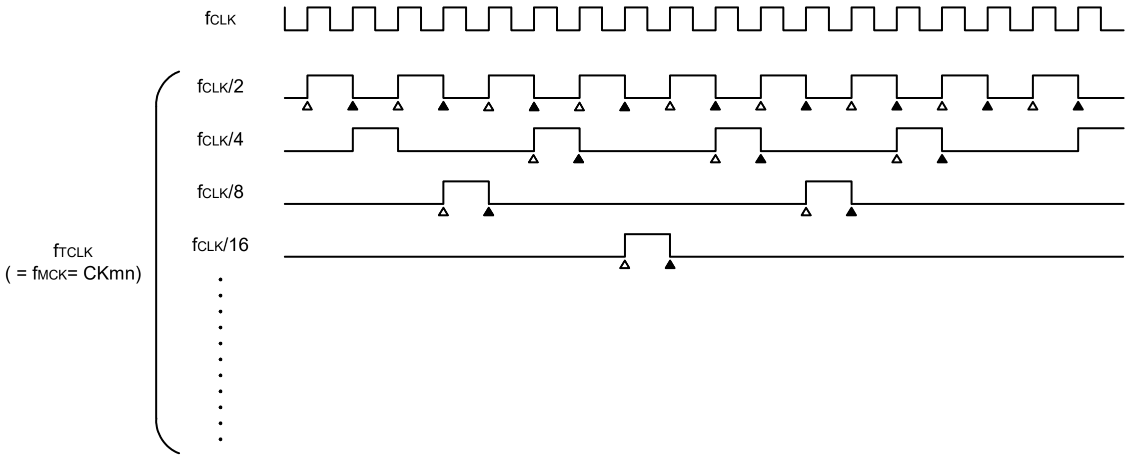
Because the timer array unit is designed to operate in synchronization with fCLK, the timings of the count clock (fTCLK) are shown below.

(1) When operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (fTCLK) is between fCLK to fCLK /2^{15} by setting of timer clock select register m (TPSm). When a divided fCLK is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of fCLK from its rising edge. When a fCLK is selected, fixed to high level.

Counting of timer count register mn (TCRmn) delayed by one period of fCLK from rising edge of the count clock, because of synchronization with fCLK. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6 - 27 Timing of fCLK and Count Clock (fTCLK) (When CCSmn = 0)



Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

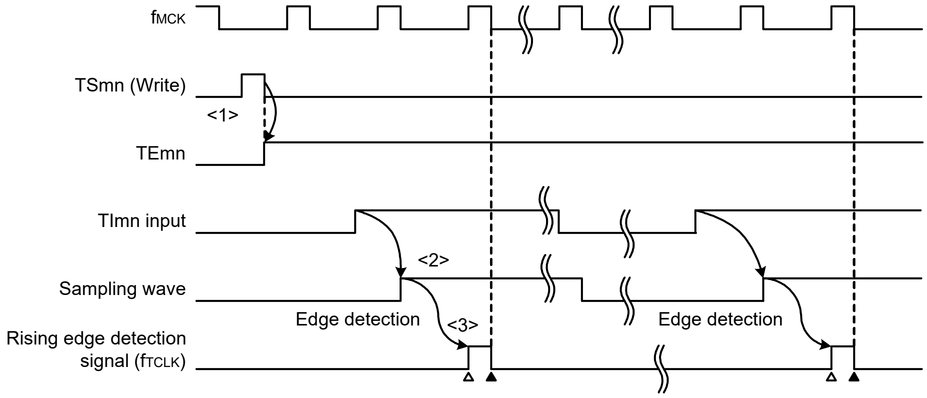
Remark 2. fCLK: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at valid edge of input signal via the TImn pin", as a matter of convenience.

Figure 6 - 28 Timing of fCLK and Count Clock (fTCLK) (When CCSmn = 1, Noise Filter Unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.

<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remark 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock

f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, and the one-shot pulse output are the same as that shown in Figure 6 - 28.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 6 - 6.

Table 6 - 6 Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation (see **6.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

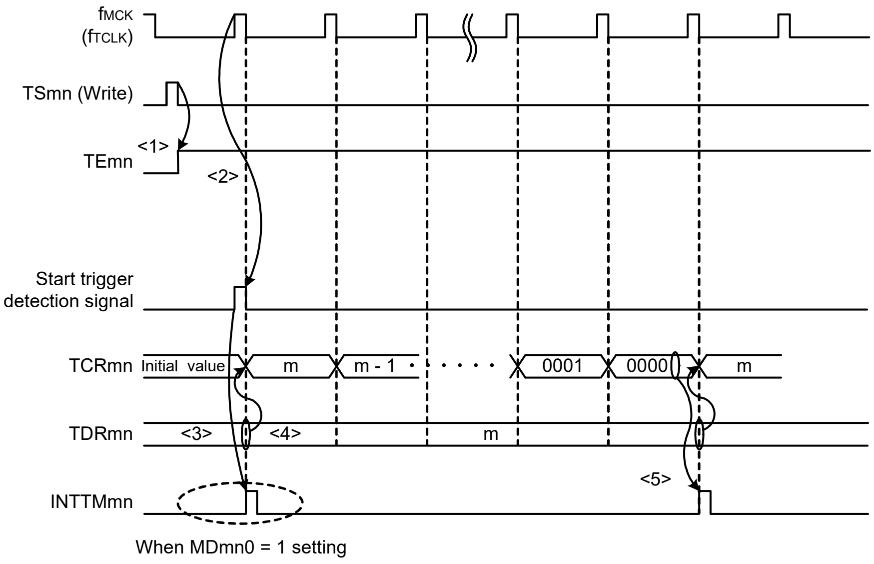
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6 - 29 Operation Timing (Interval Timer Mode)



Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

(2) Operation of event counter mode

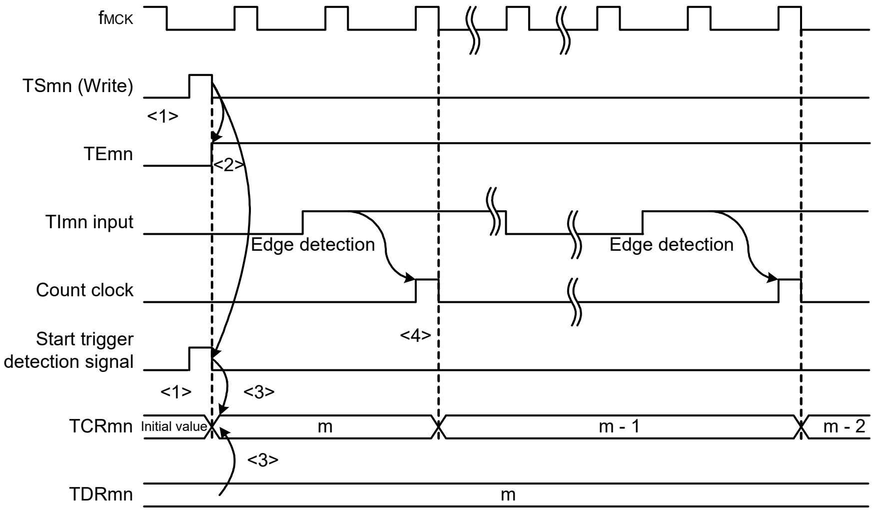
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.

<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 6 - 30 Operation Timing (Event Counter Mode)



Remark Figure 6-30 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

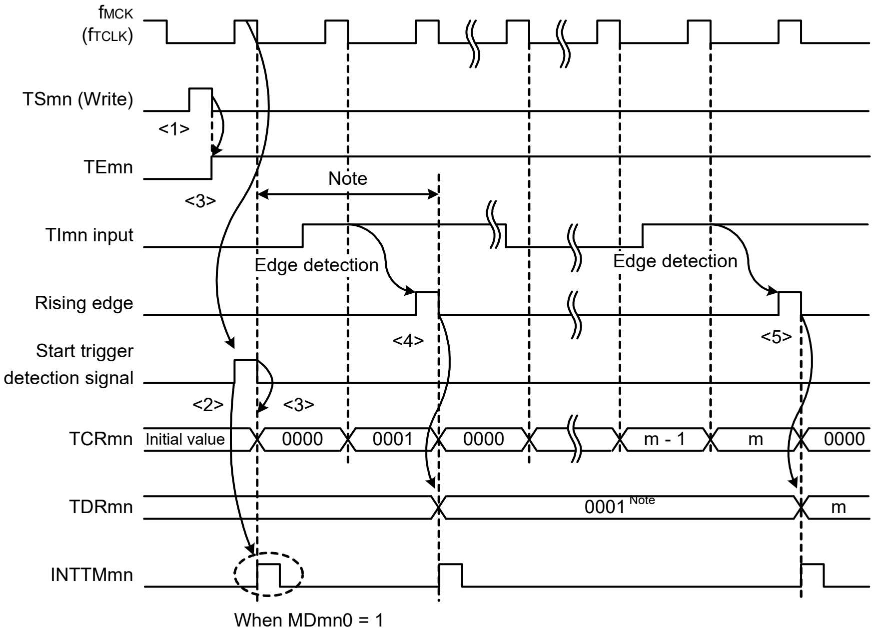
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)

<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 31 Operation Timing (Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark Figure 6-31 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

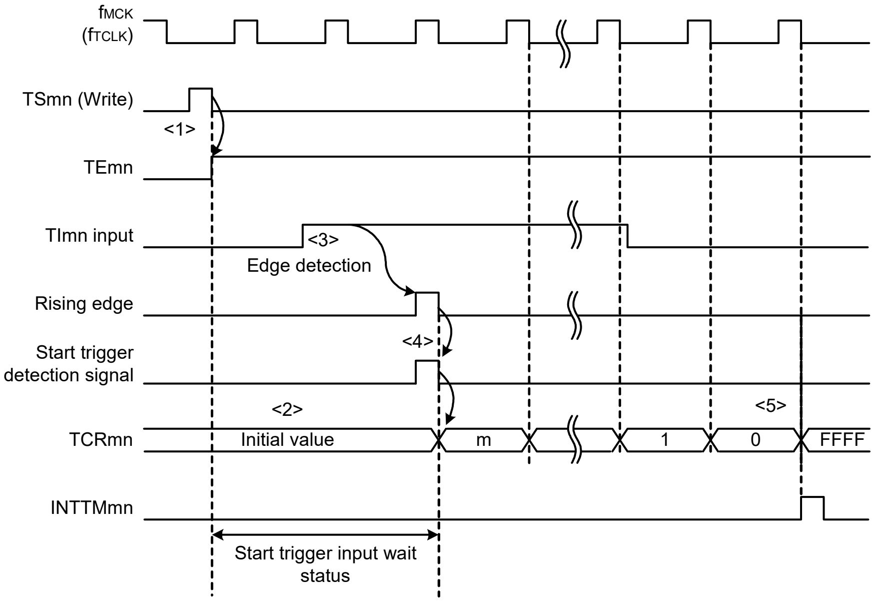
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 6 - 32 Operation Timing (One-count Mode)



Remark Figure 6-32 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

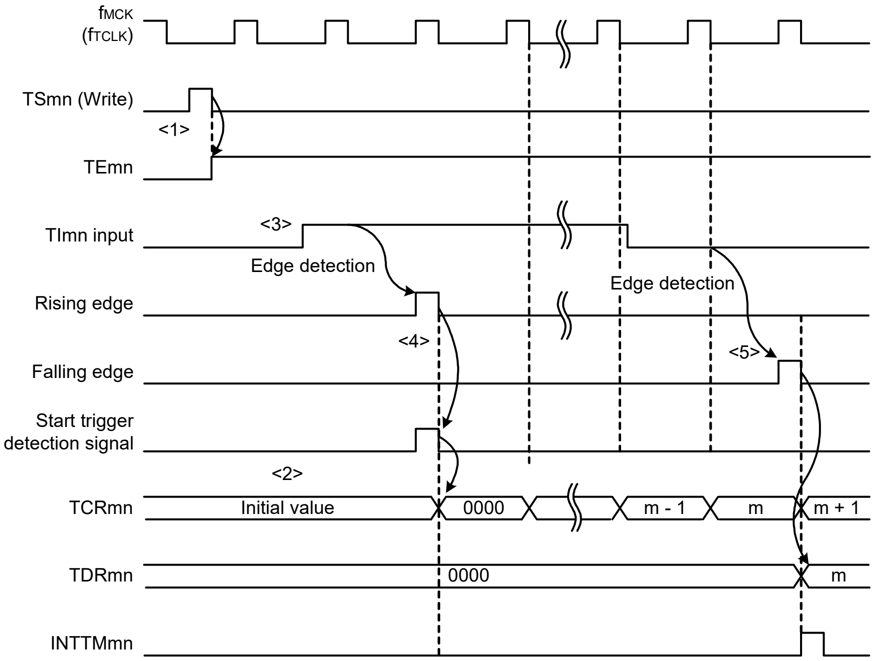
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.

<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 33 Operation Timing (in Capture & One-count Mode: High-level Width Measurement)

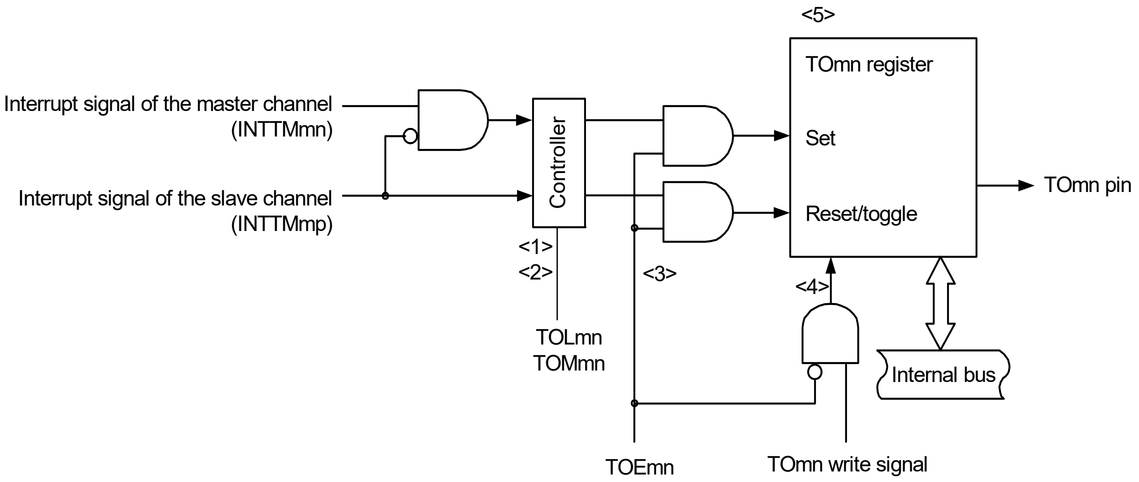


Remark Figure 6-33 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6 - 34 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

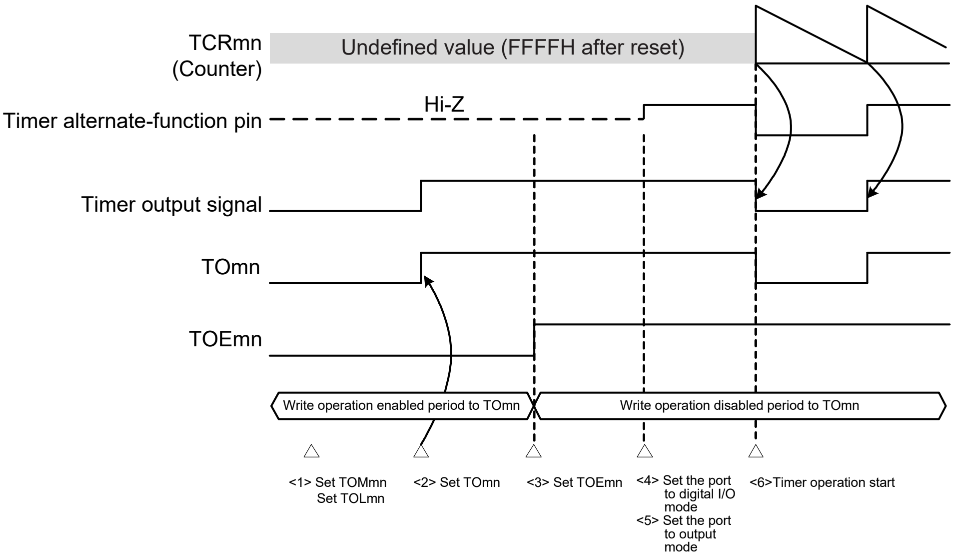
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0: p = 1, 2, 3
n = 2: p = 3

6.6.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6 - 35 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3.14 Port mode registers 0, 1, 3 (PM0, PM1, PM3)**).

<5> The port I/O setting is set to output (see **6.3.14 Port mode registers 0, 1, 3 (PM0, PM1, PM3)**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TOm, TOEm, TOLm, and TOMm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), timer output level register m (TOLm), and timer output mode register m (TOMm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.8 and 6.9.

When the values set to the TOEm, TOLm, and TOMm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

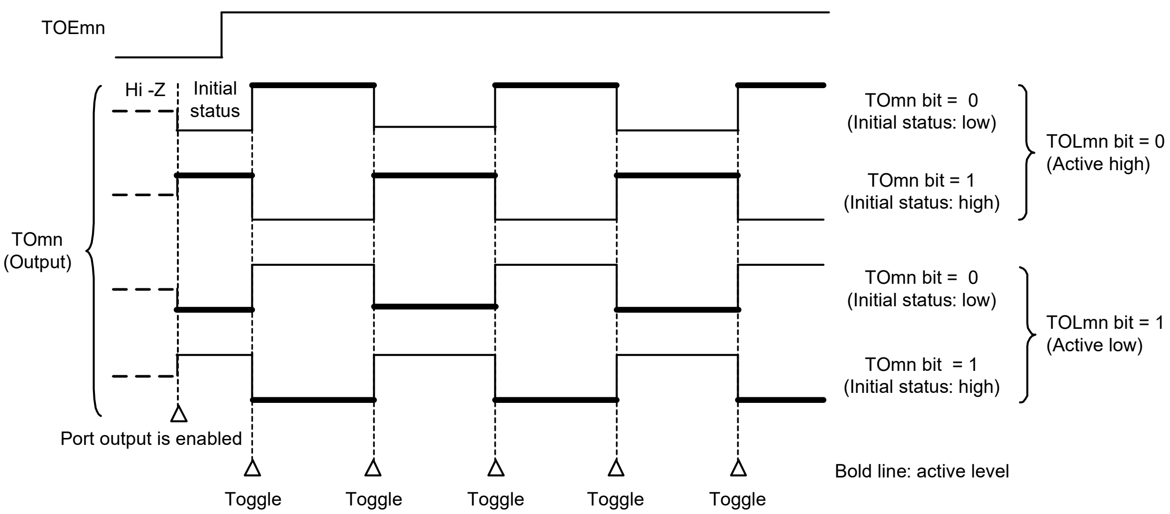
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 6 - 36 TOmn Pin Output Status at Toggle Output (TOMmn = 0)

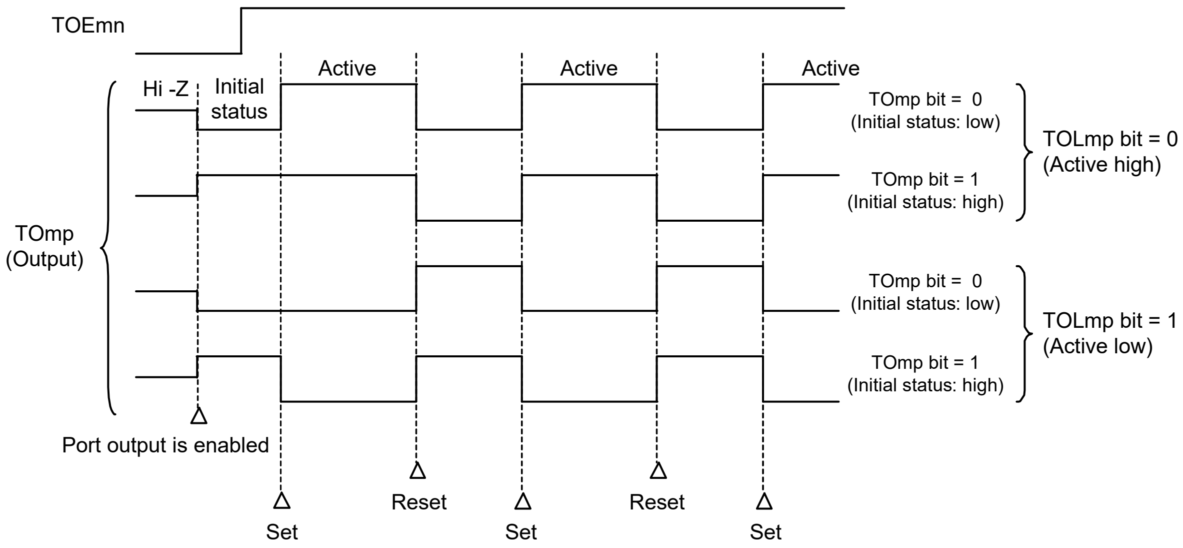


Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))
When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6 - 37 TOmn Pin Output Status at PWM Output (TOMmn = 1)



Remark 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

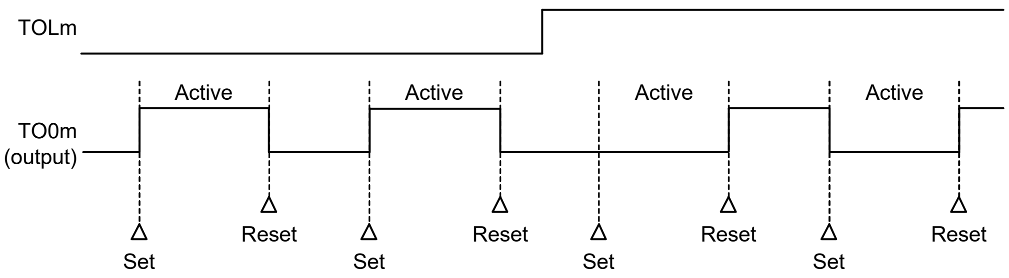
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6 - 38 Operation When TOLm Register Has Been Changed during Timer Operation



Remark 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

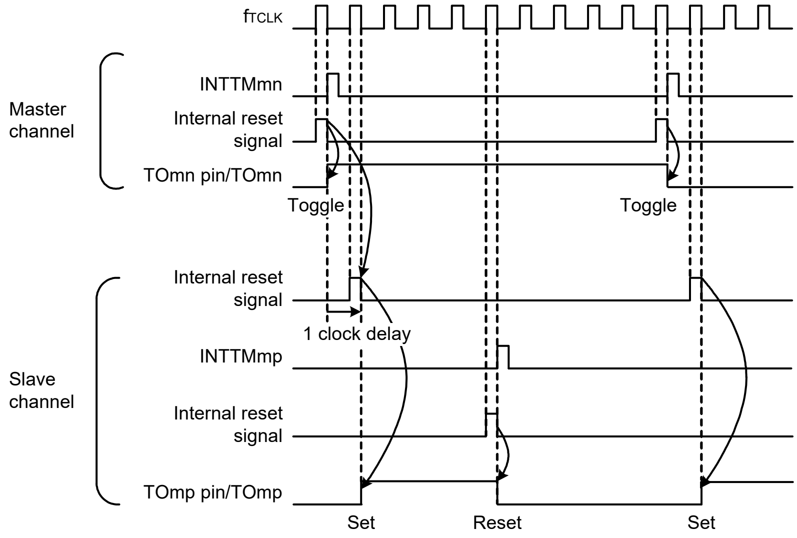
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6 - 39 shows the set/reset operating statuses where the master/slave channels are set as follows.

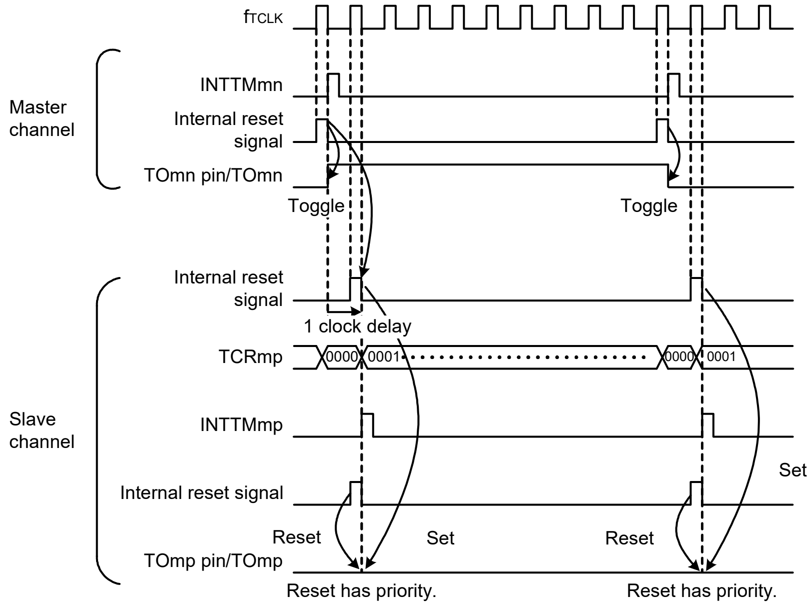
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6 - 39 Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remark 1. Internal reset signal:TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

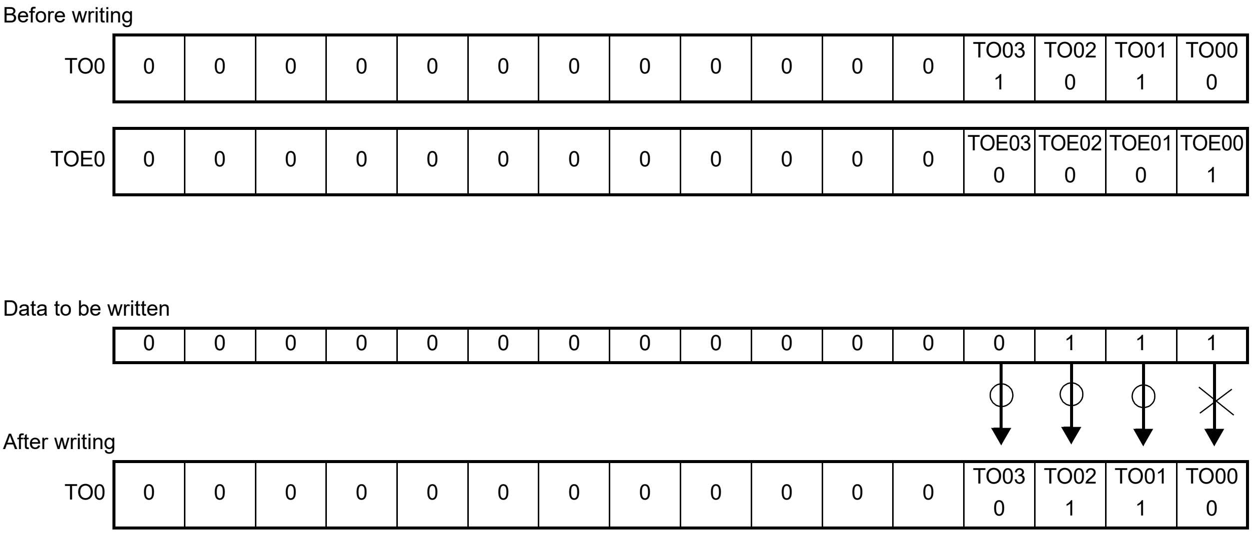
Remark 2. m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0: p = 1, 2, 3
n = 2: p = 3

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 6 - 40 Example of TO0n Bit Collective Manipulation



Before writing

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 1	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE03 0	TOE02 0	TOE01 0	TOE00 1

Data to be written

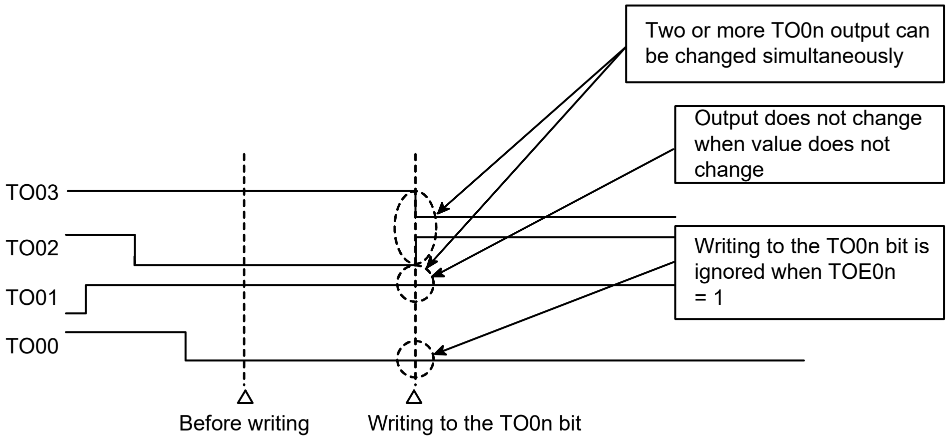
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

After writing

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 0	TO02 1	TO01 1	TO00 0

Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored. TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Figure 6 - 41 TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

6.6.5 Timer interrupt and TOmn pin output at operation start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

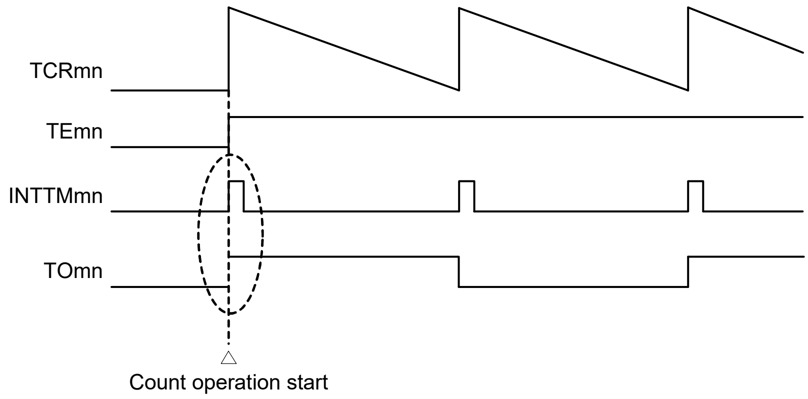
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

Figure 6 - 42 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

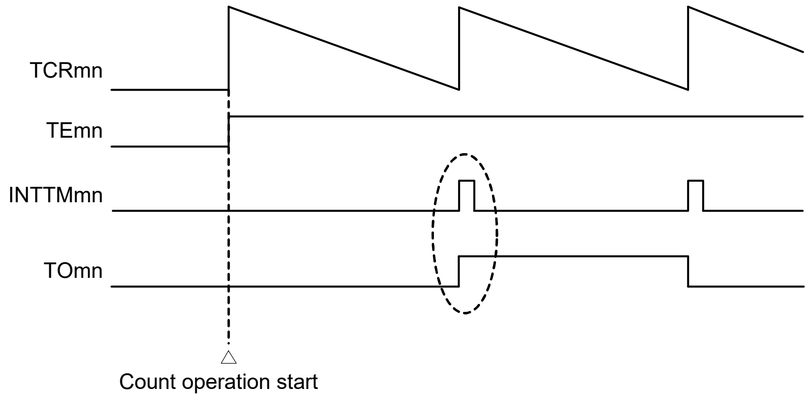
Figure 6 - 42 Operation Examples of Timer Interrupt at Count Operation Start and TOmn Output

(a) When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

(b) When MDmn0 is set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

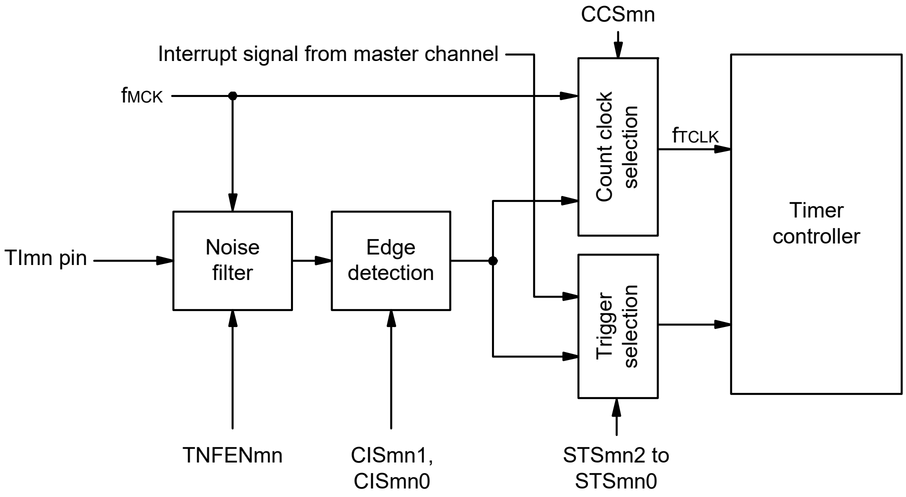
6.7 Timer Input (TImn) Control

6.7.1 TImn input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller.

Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

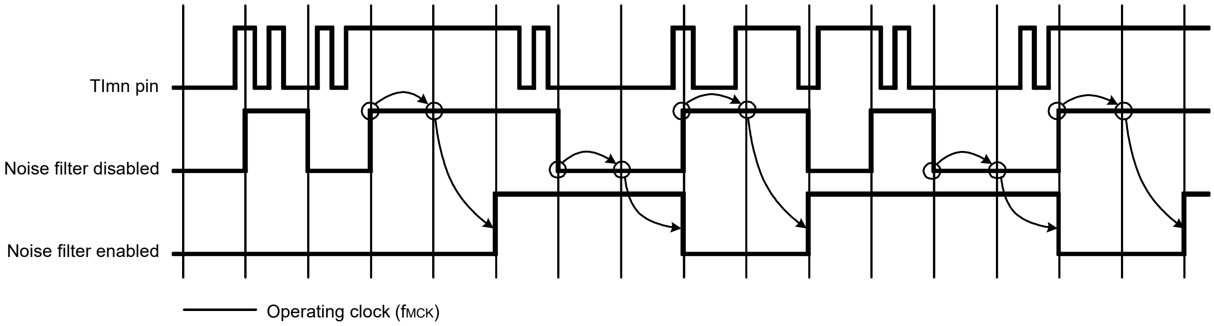
Figure 6 - 43 Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6 - 44 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution **The input waveforms to the TImn pin are shown to explain the operation when the noise filter is enabled or disabled. When actually inputting waveforms, input them according to the TImn input high-level and low-level widths listed in 29.5 AC Characteristics.**

6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

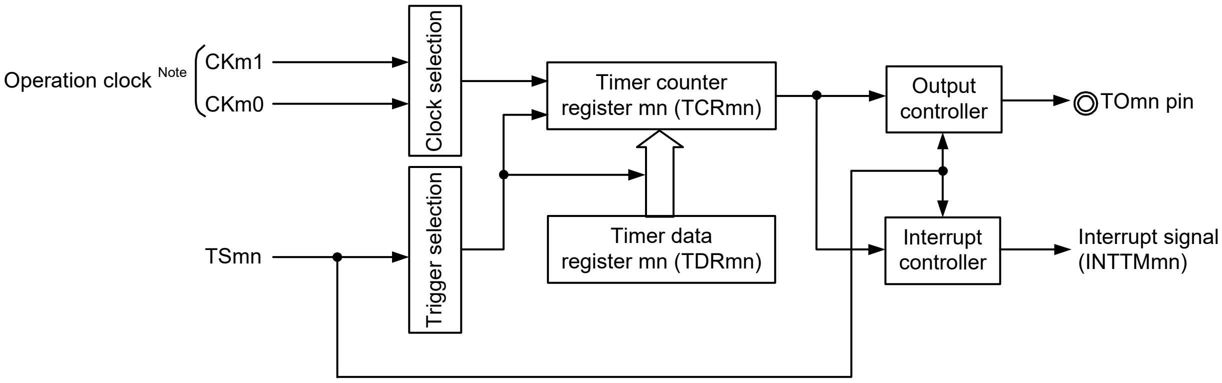
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

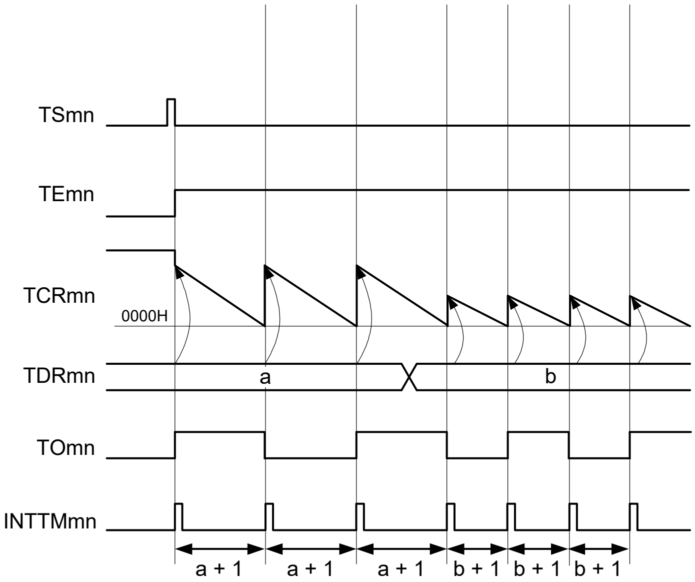
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 45 Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 46 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 6 - 47 Example of Set Contents of Registers during Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (fMCK).

Operation clock (fMCK) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 48 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Input clock for timer array unit 0 is supplied. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output.	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn.	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit.	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

(**Remark** is listed on the next page.)

Figure 6 - 49 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

Timer count register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn) of timer channel start register m (TSm) to 1.

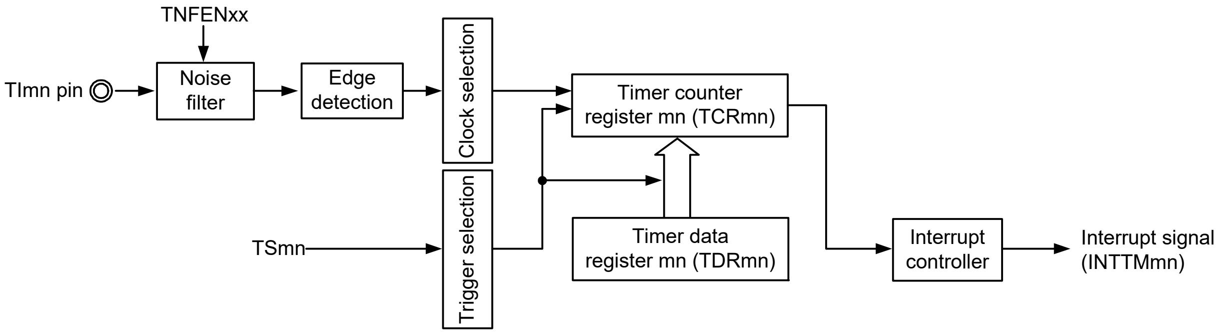
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

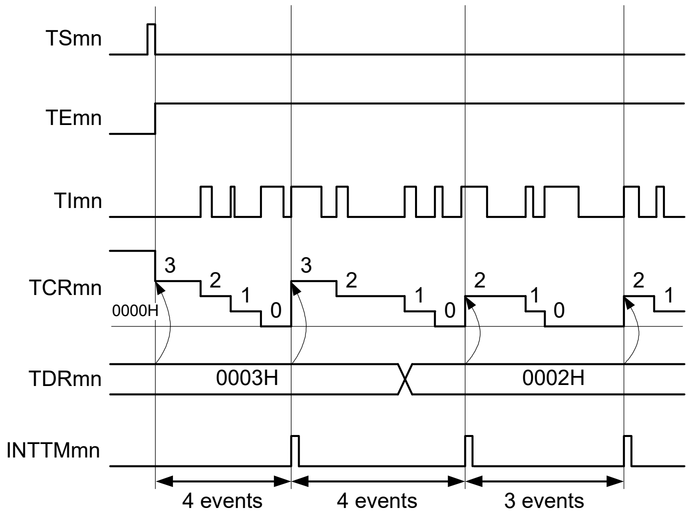
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6 - 50 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 51 Example of Basic Timing of Operation as External Event Counter

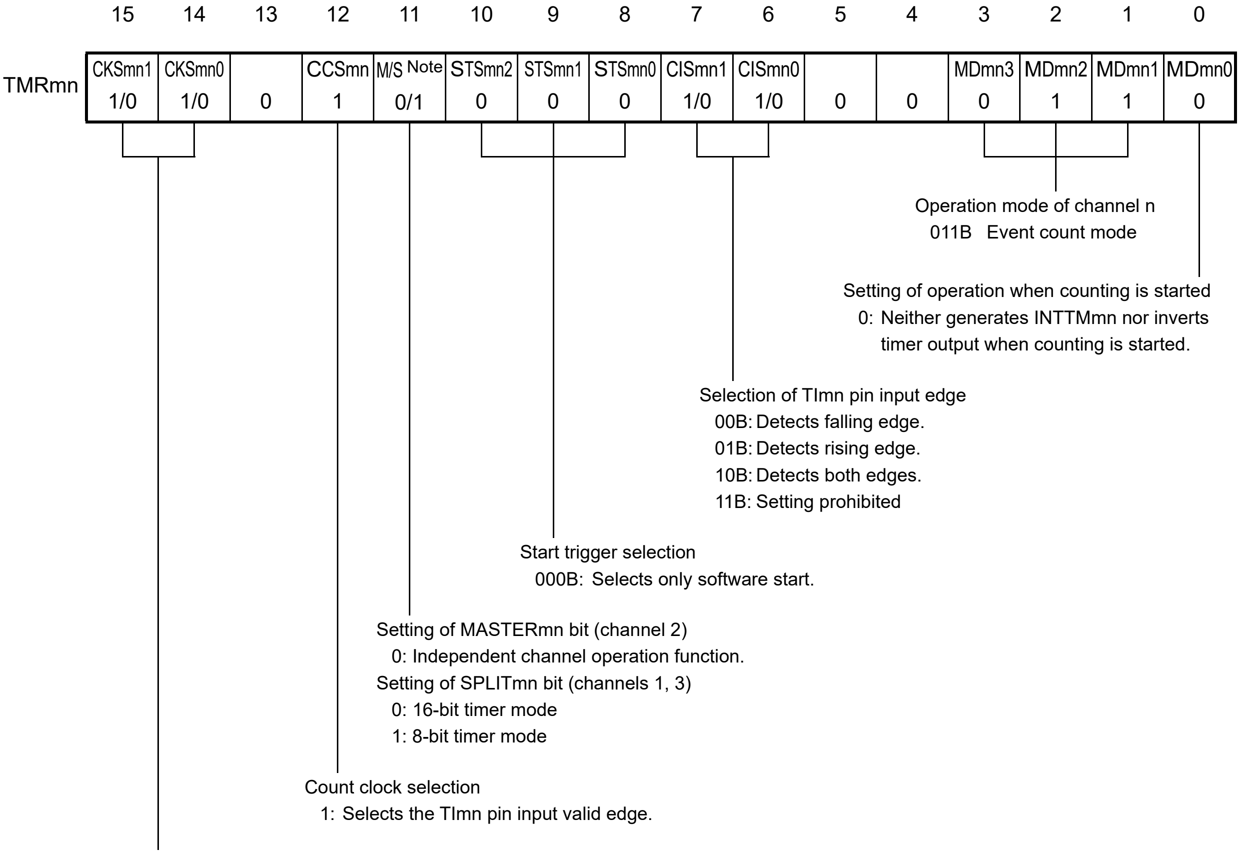


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6 - 52 Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Note
TMRm2: MASTERmn bit
TMRm1, TMRm3 SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 53 Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Input clock supply for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0.	Input clock for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.3 Operation as frequency divider (channel 0 of unit 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

Timer count register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 6 - 54 Block Diagram of Operation as Frequency Divider

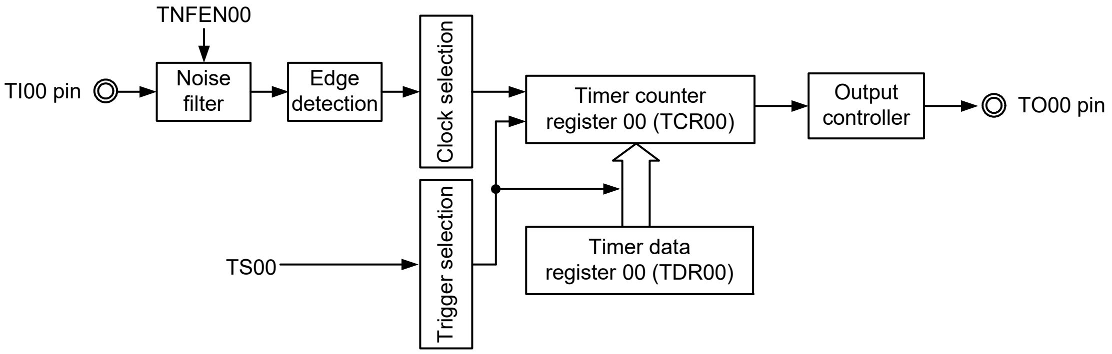
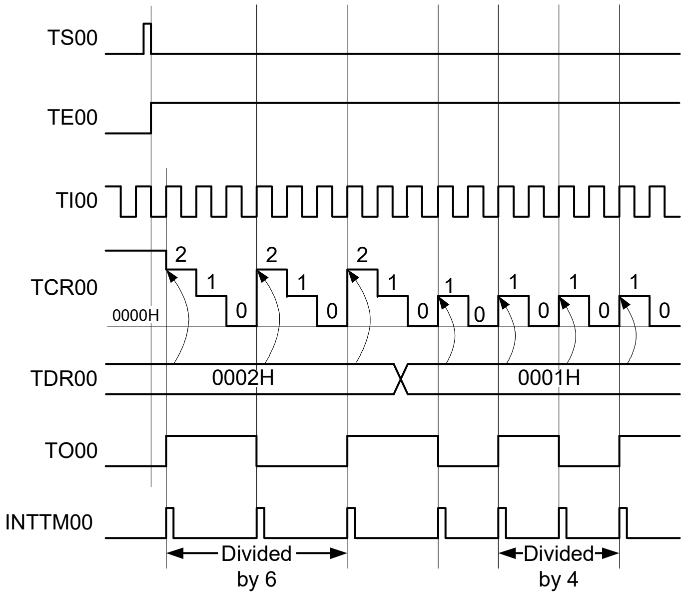


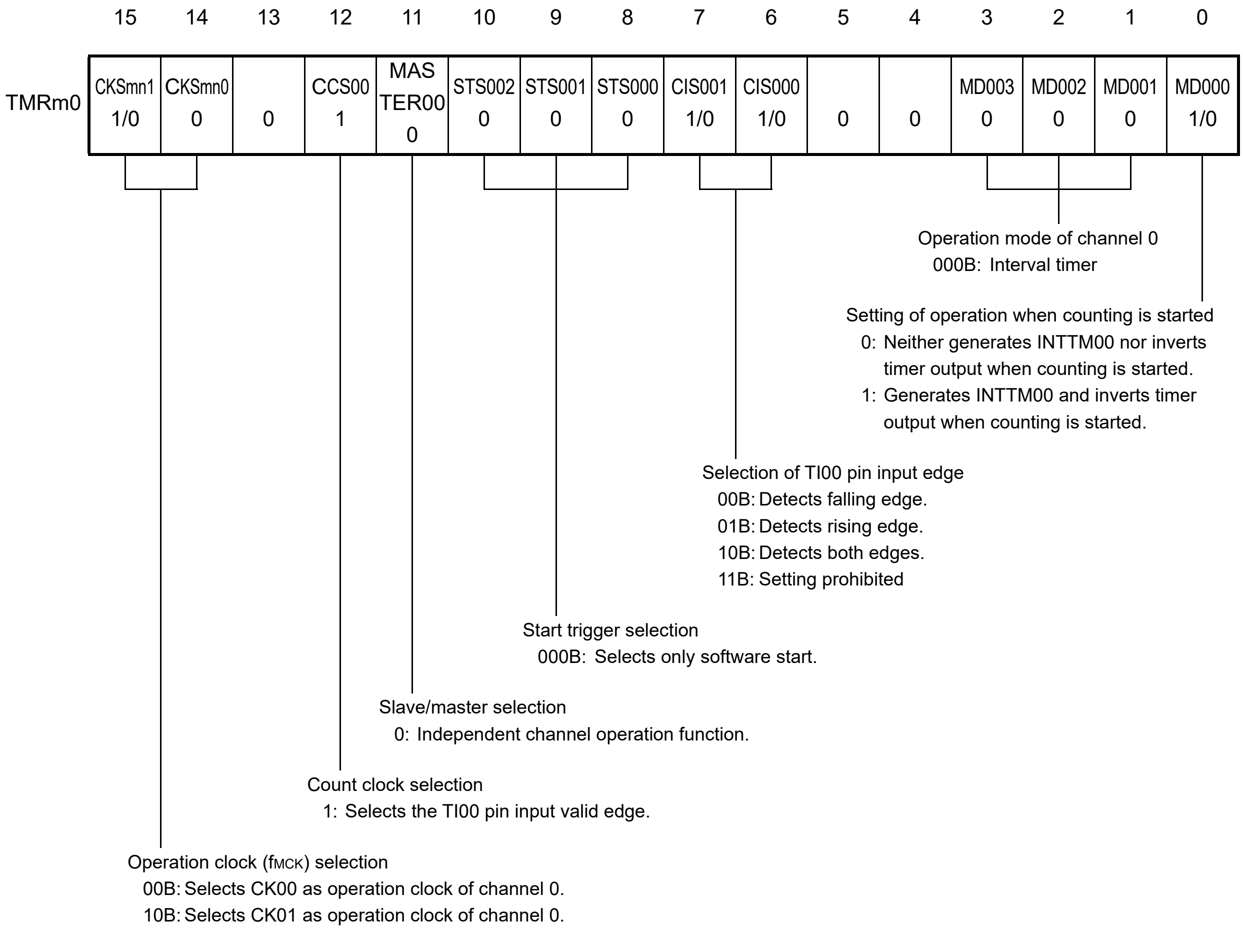
Figure 6 - 55 Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



Remark TS00: Bit n of timer channel start register 0 (TS0)
TE00: Bit n of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer count register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Figure 6 - 56 Example of Set Contents of Registers during Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)



Operation mode of channel 0
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM00 nor inverts timer output when counting is started.
1: Generates INTTM00 and inverts timer output when counting is started.

Selection of TI00 pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Independent channel operation function.

Count clock selection
1: Selects the TI00 pin input valid edge.

Operation clock (fMCK) selection
00B: Selects CK00 as operation clock of channel 0.
10B: Selects CK01 as operation clock of channel 0.

(b) Timer output register 0 (TO0)

	Bit 0
TO0	TOmn 0

0: Outputs 0 from TO00.
1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEmn 0

0: Stops the TO00 output operation by counting operation.
1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0
TOL0	TOLmn 0

0: Cleared to 0 when TOM00 = 0 (master channel output mode)

(e) Timer output mode register 0 (TOM0)

	Bit 0
TOM0	TOMmn 0

0: Sets master channel output mode.

Figure 6 - 57 Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 00 (TMR00) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 bit to 1 and enables operation of TO00. →	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). → Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. → The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit. →	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. → When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

6.8.4 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.

The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

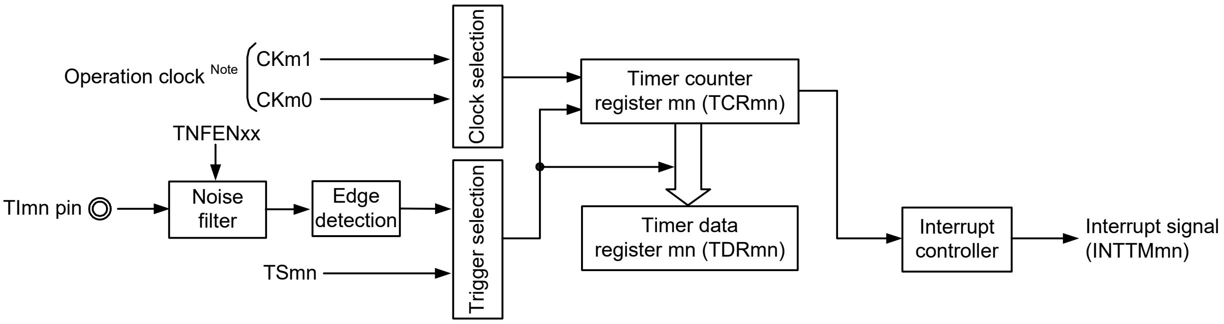
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

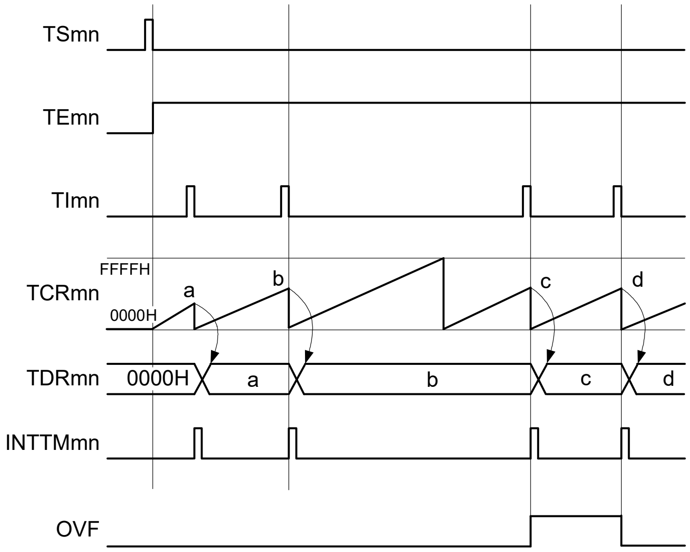
Figure 6 - 58 Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 59 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

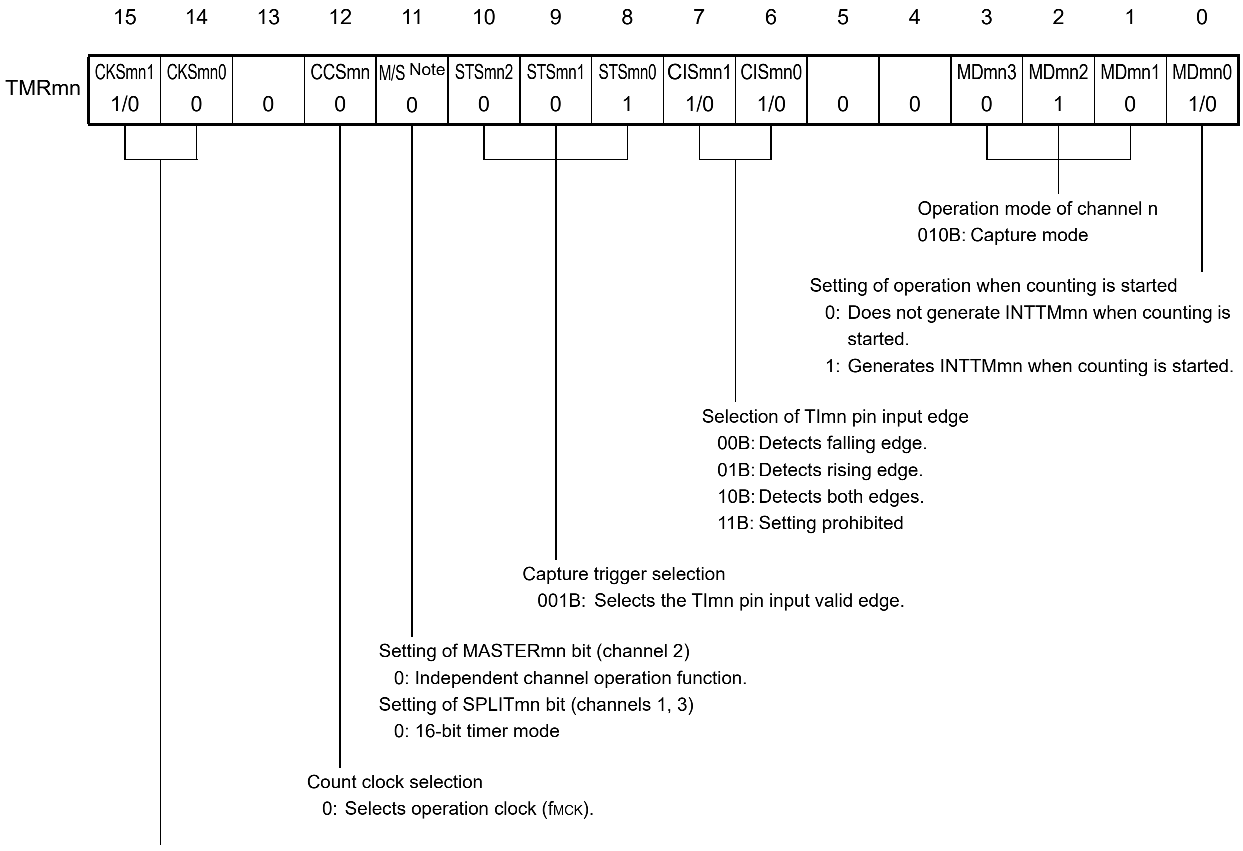


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 60 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	M/S Note 0	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 0	MDmn2 1	MDmn1 0	MDmn0 1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 61 Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0.	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

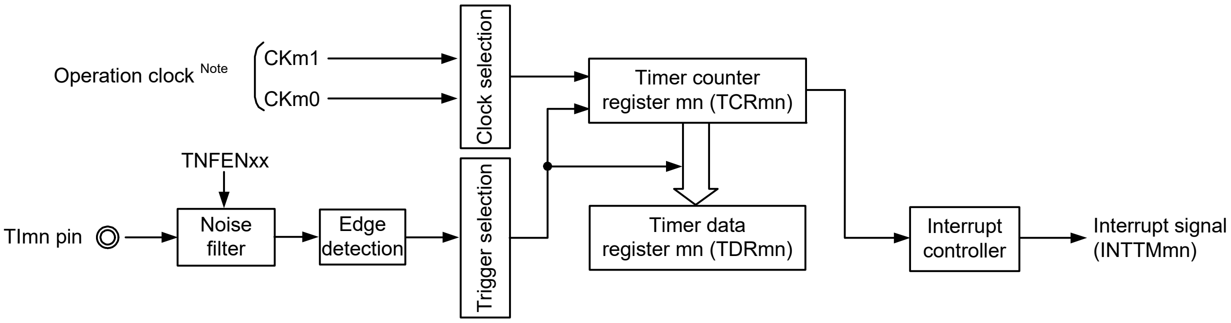
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.
When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.
Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

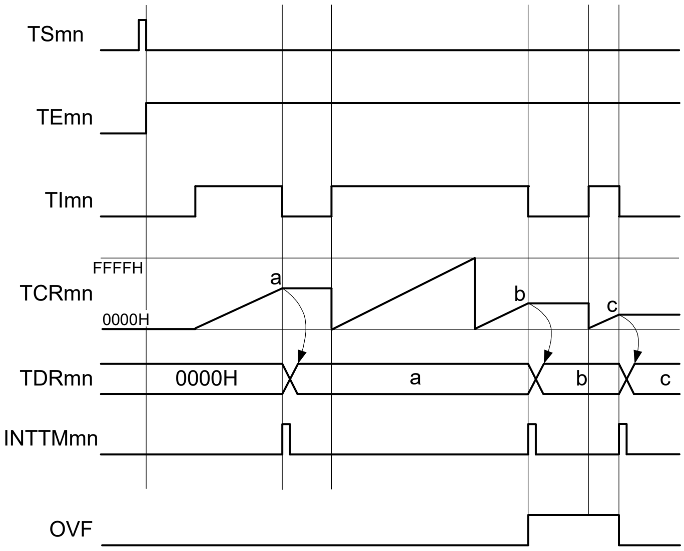
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6 - 62 Block Diagram of Operation as Input Signal High-/Low-level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 63 Example of Basic Timing of Operation as Input Signal High-/Low-level Width Measurement



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TImn: TImn pin input signal

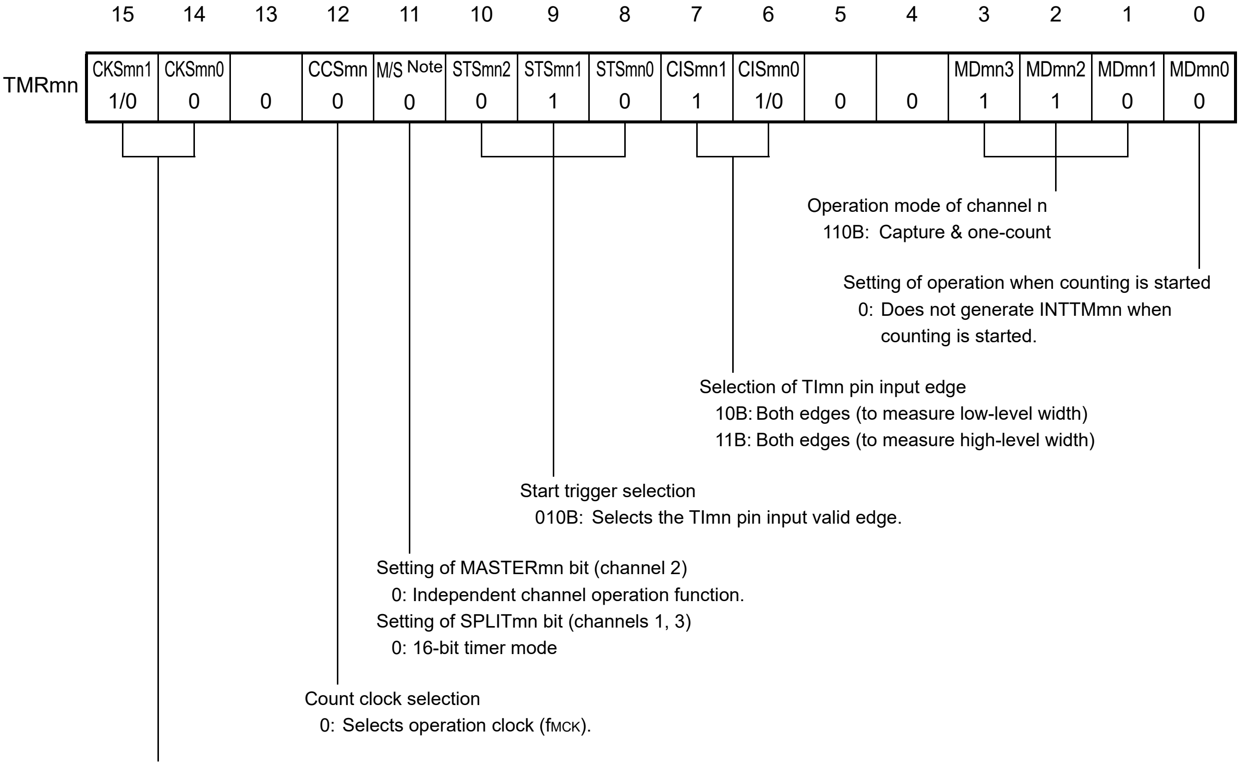
TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 64 Example of Set Contents of Registers to Measure Input Signal High-/Low-level Width

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	M/S Note 0	STSmn2 0	STSmn1 1	STSmn0 0	CISmn1 1	CISmn0 1/0	0	0	MDmn3 1	MDmn2 1	MDmn1 0	MDmn0 0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note		
	TMRm2:	MASTERmn bit
	TMRm1, TMRm3:	SPLITmn bit
	TMRm0:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 65 Operation Procedure When Input Signal High-/Low-level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge. →	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

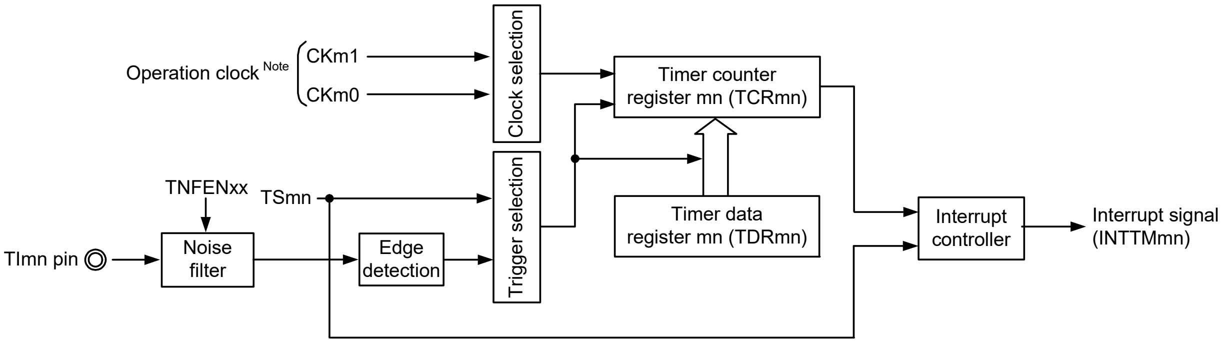
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

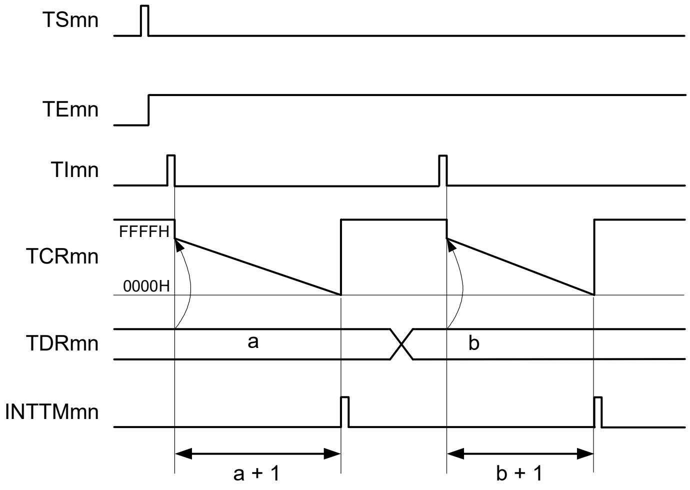
Figure 6 - 66 Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 67 Example of Basic Timing of Operation as Delay Counter



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 6 - 68 Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 1/0	 0	CCSmn 0	M/S Note 0/1	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	 0	 0	MDmn3 1	MDmn2 0	MDmn1 0	MDmn0 0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6 - 69 Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status is set.
	The counter starts counting down by the next start trigger detection. • Detects the TImn pin input valid edge. • Sets the TSmn bit to 1 by the software. →	Value of the TDRmn register is loaded to the timer count register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

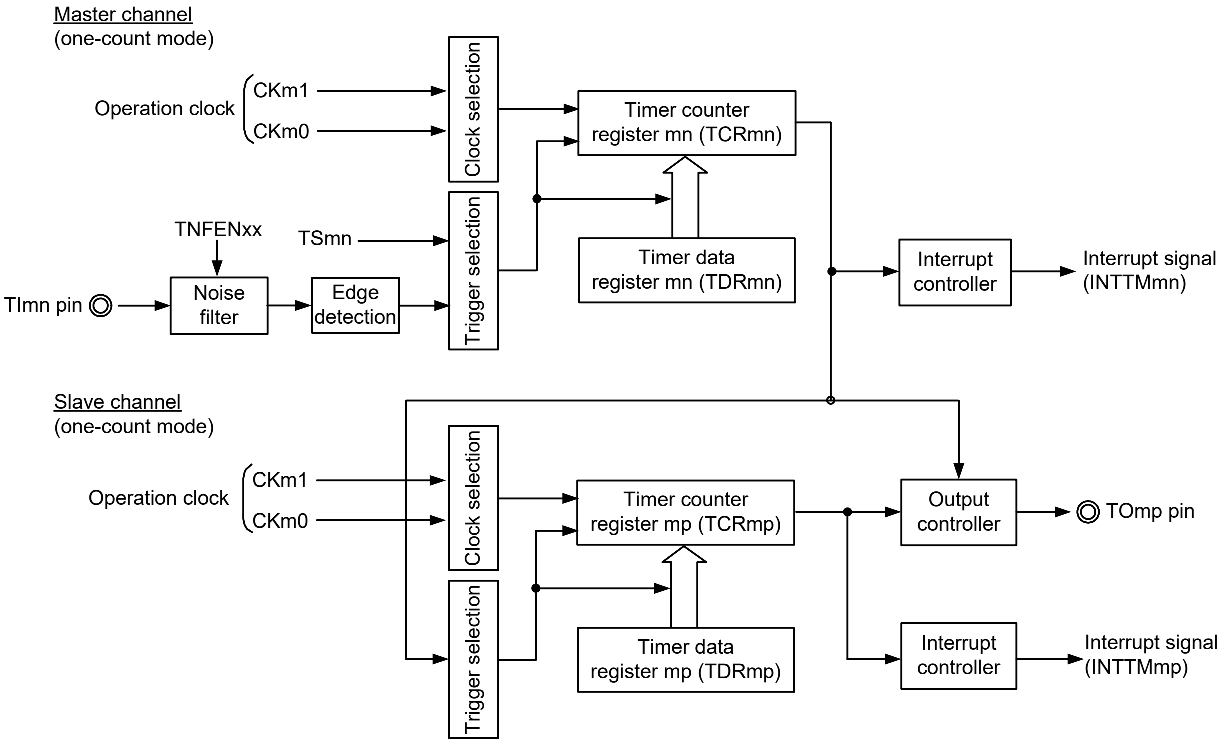
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution **The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform may be output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

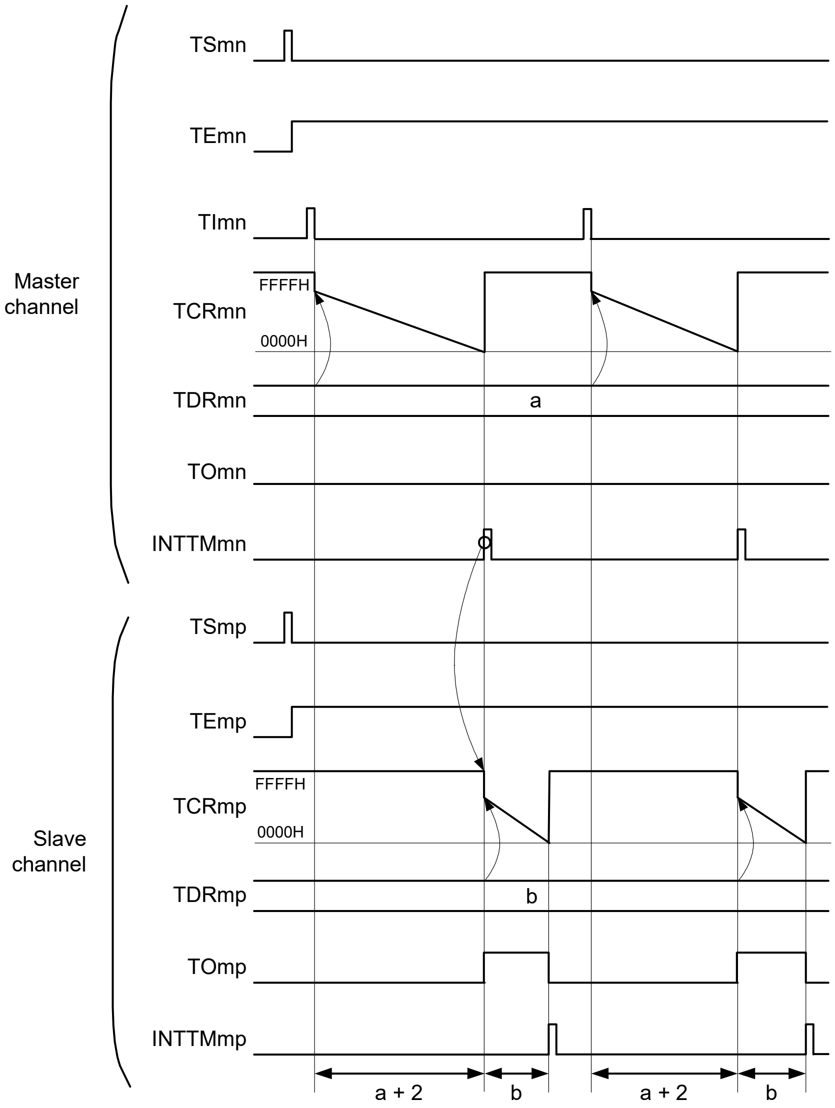
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 70 Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 71 Example of Basic Timing of Operation as One-Shot Pulse Output Function

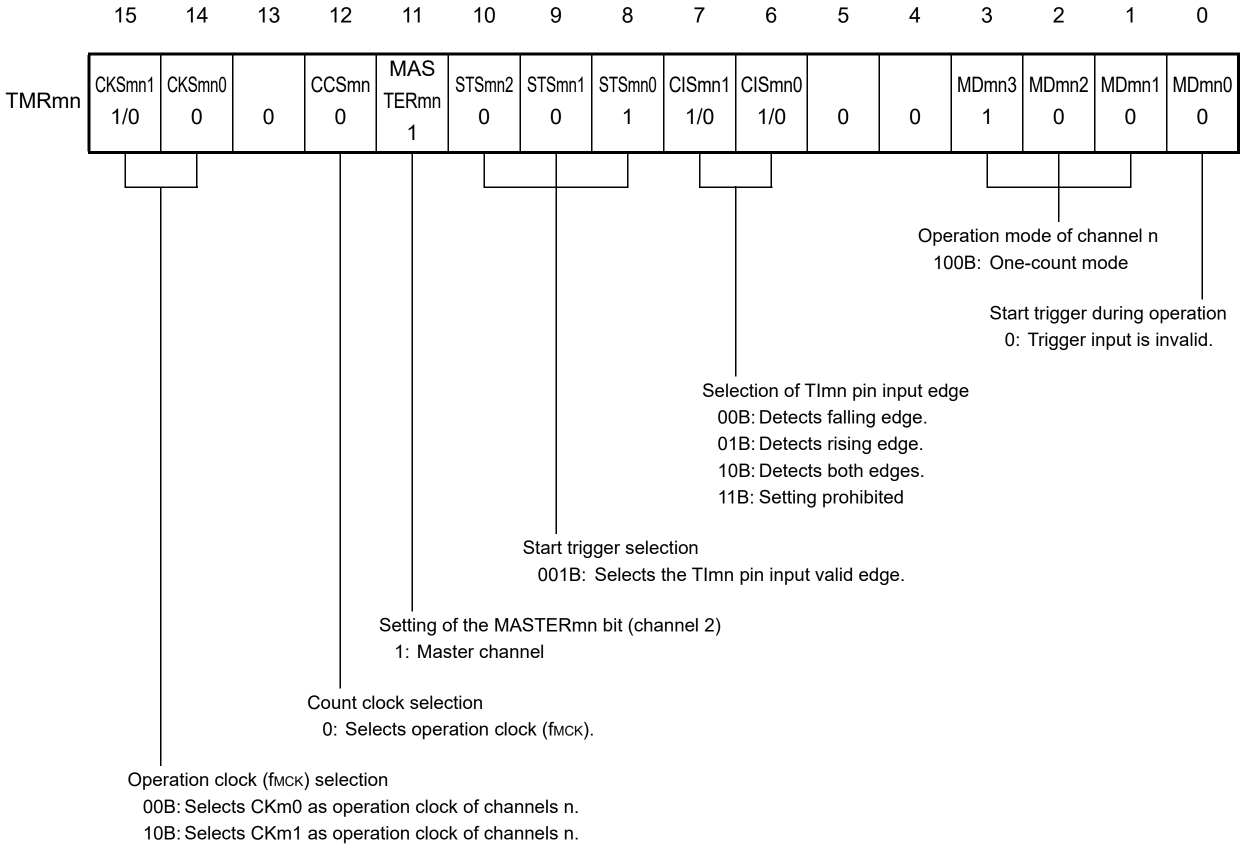


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 72 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MASTERmn 1	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 1	MDmn2 0	MDmn1 0	MDmn0 0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of the MASTERmn bit (channel 2)
1: Master channel

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

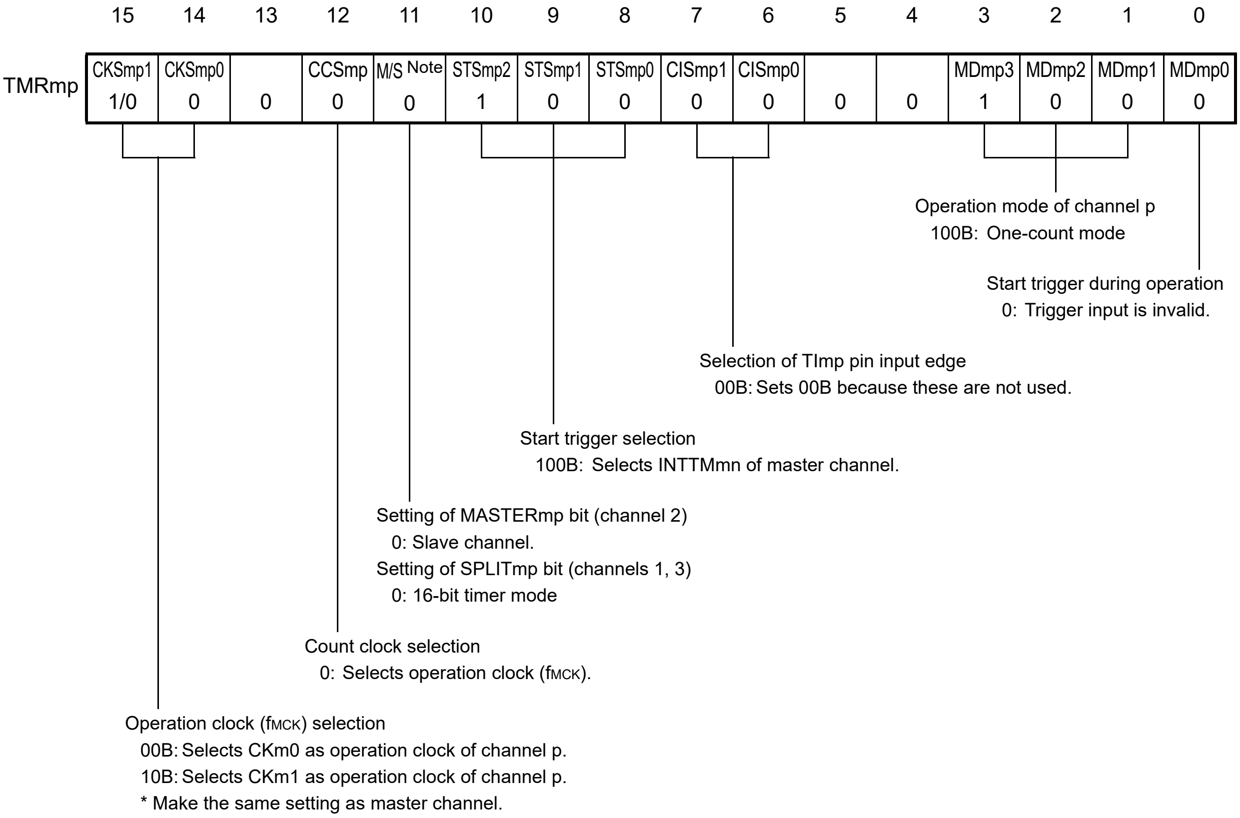
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 6 - 73 Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp bit (channel 2)
0: Slave channel.
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp
	1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp
	1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp
	1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 74 Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. → Sets the TOEmp bit to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

(**Remark** is listed on the next page.)

Figure 6 - 75 Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software[Note]	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) when the TImn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn (master) and TTmp (slave) bits are set to 1 at the same time.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

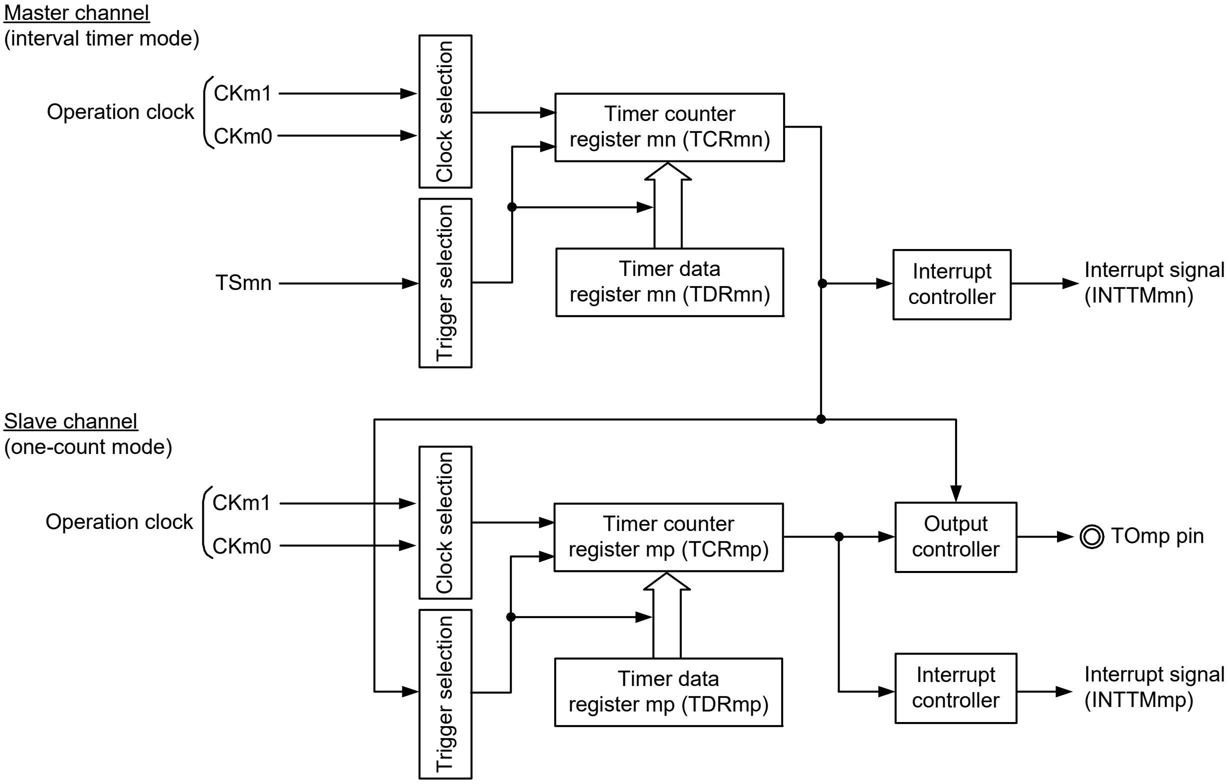
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

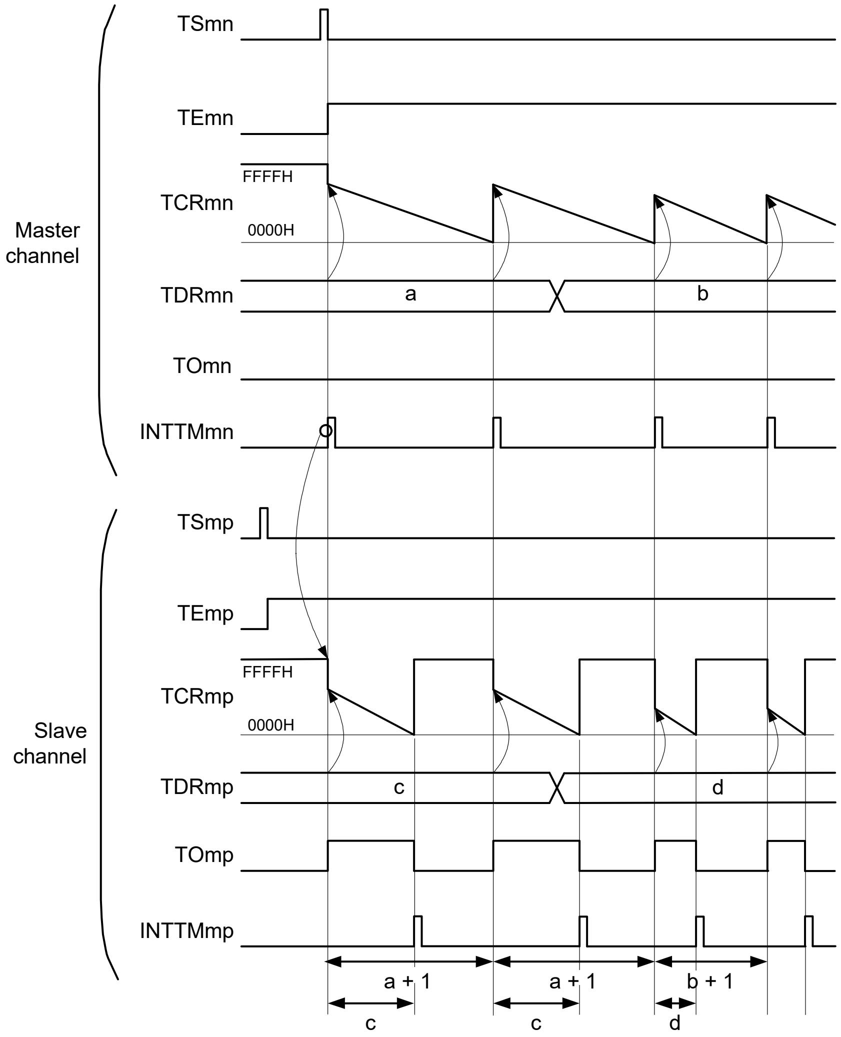
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 76 Block Diagram of Operation as PWM Function



Remark	m: Unit number (m = 0), n: Channel number (n = 0, 2) p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 77 Example of Basic Timing of Operation as PWM Function



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 78 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MASTERmn 1	STSmn2 0	STSmn1 0	STSmn0 0	CISmn1 0	CISmn0 0	0	0	MDmn3 0	MDmn2 0	MDmn1 0	MDmn0 1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of the MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

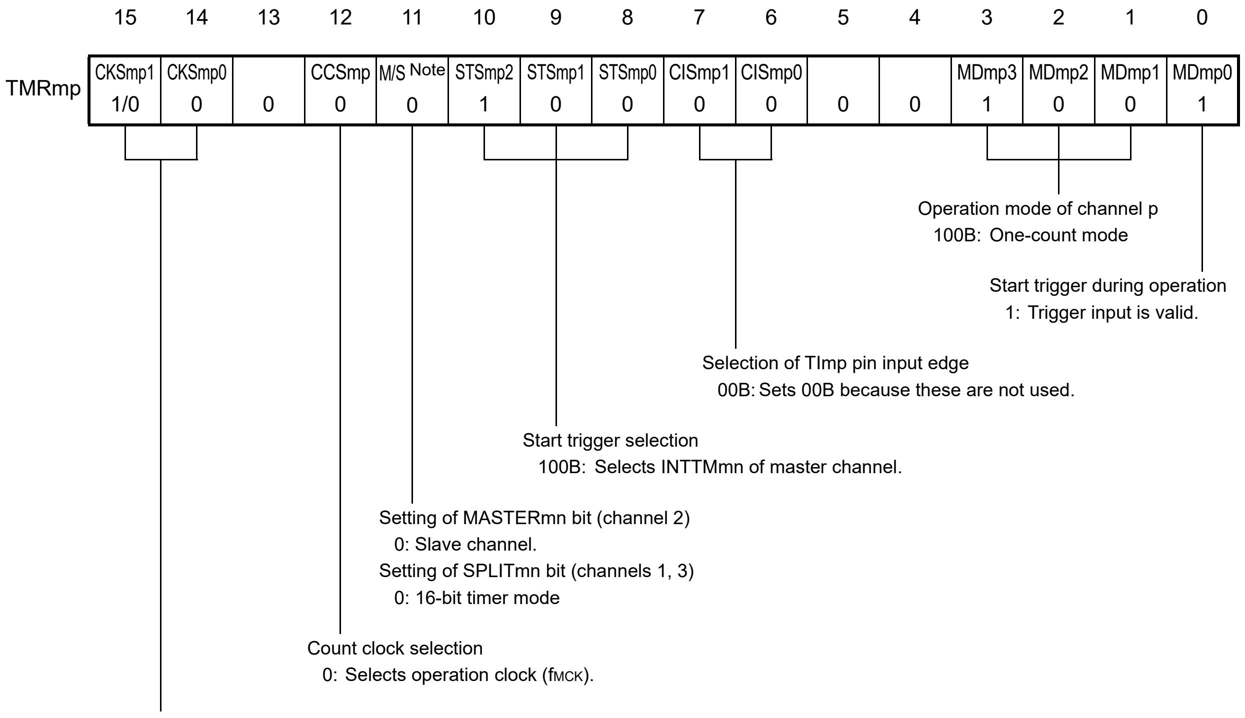
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 6 - 79 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channel 2)
0: Slave channel.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the slave channel output mode.

Note TMRm2: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 6 - 80 Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

(**Remark** is listed on the next page.)

Figure 6 - 81 Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0.	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.
For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

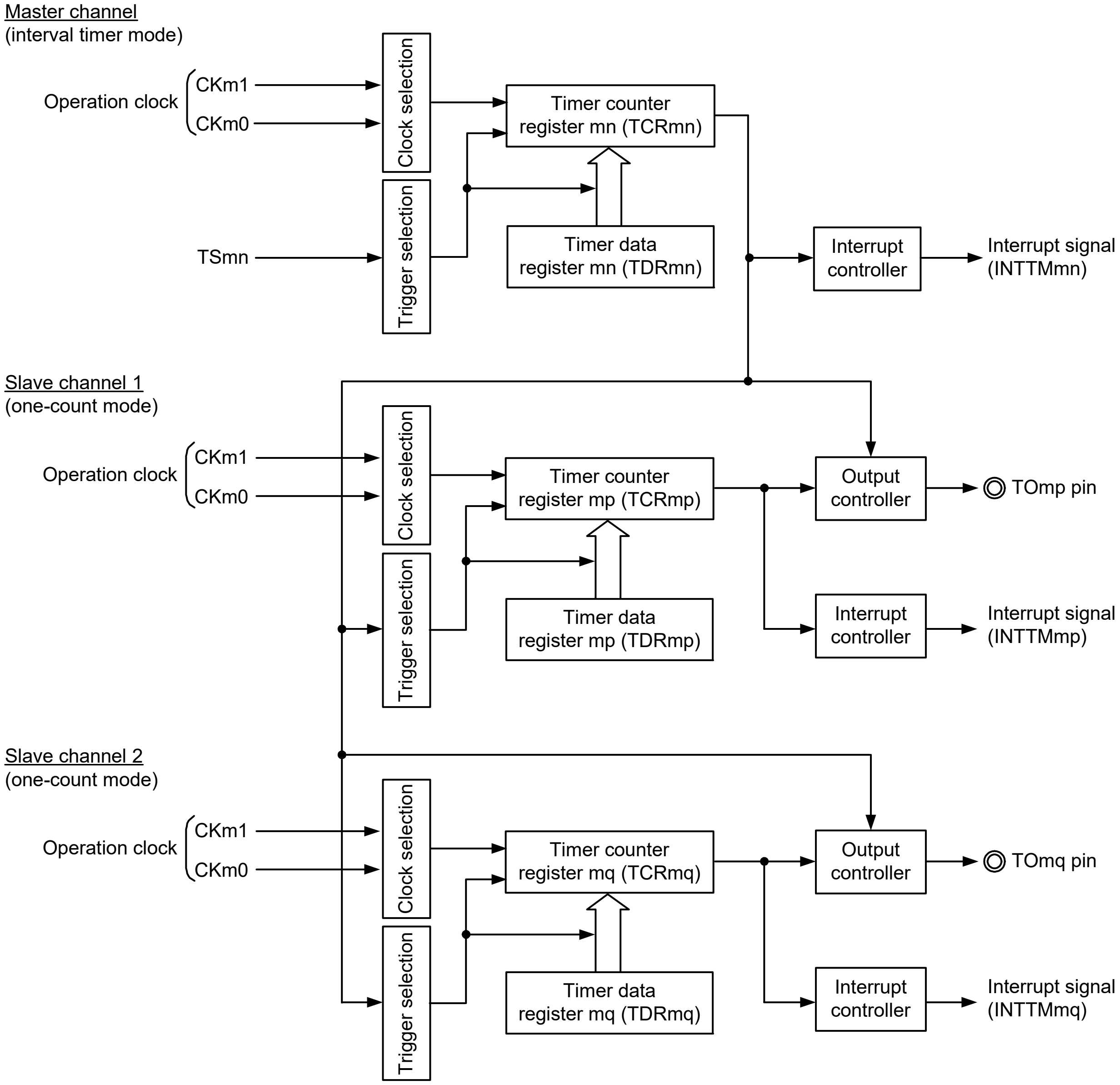
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.
The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

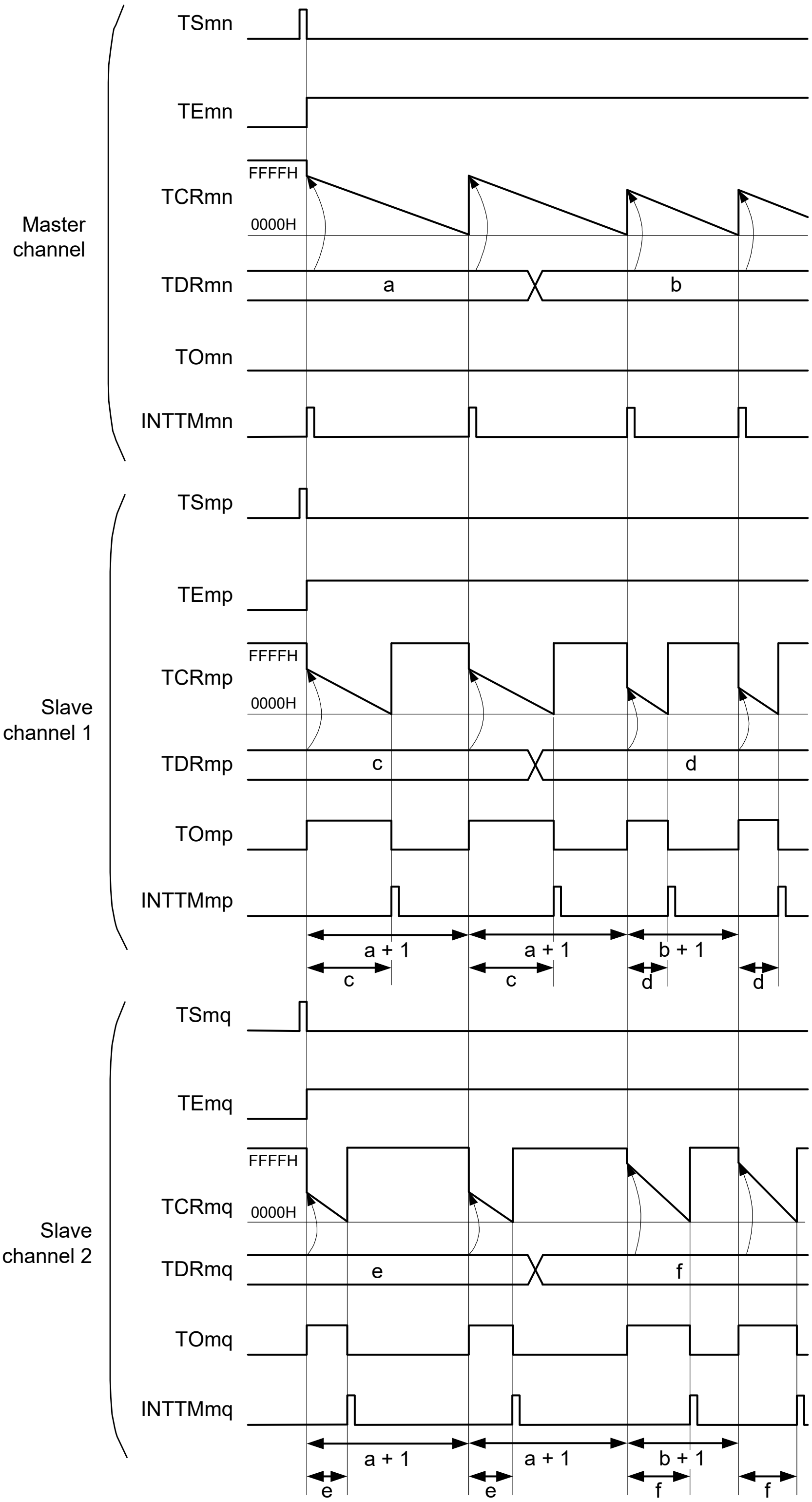
Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 6 - 82 Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Figure 6 - 83 Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



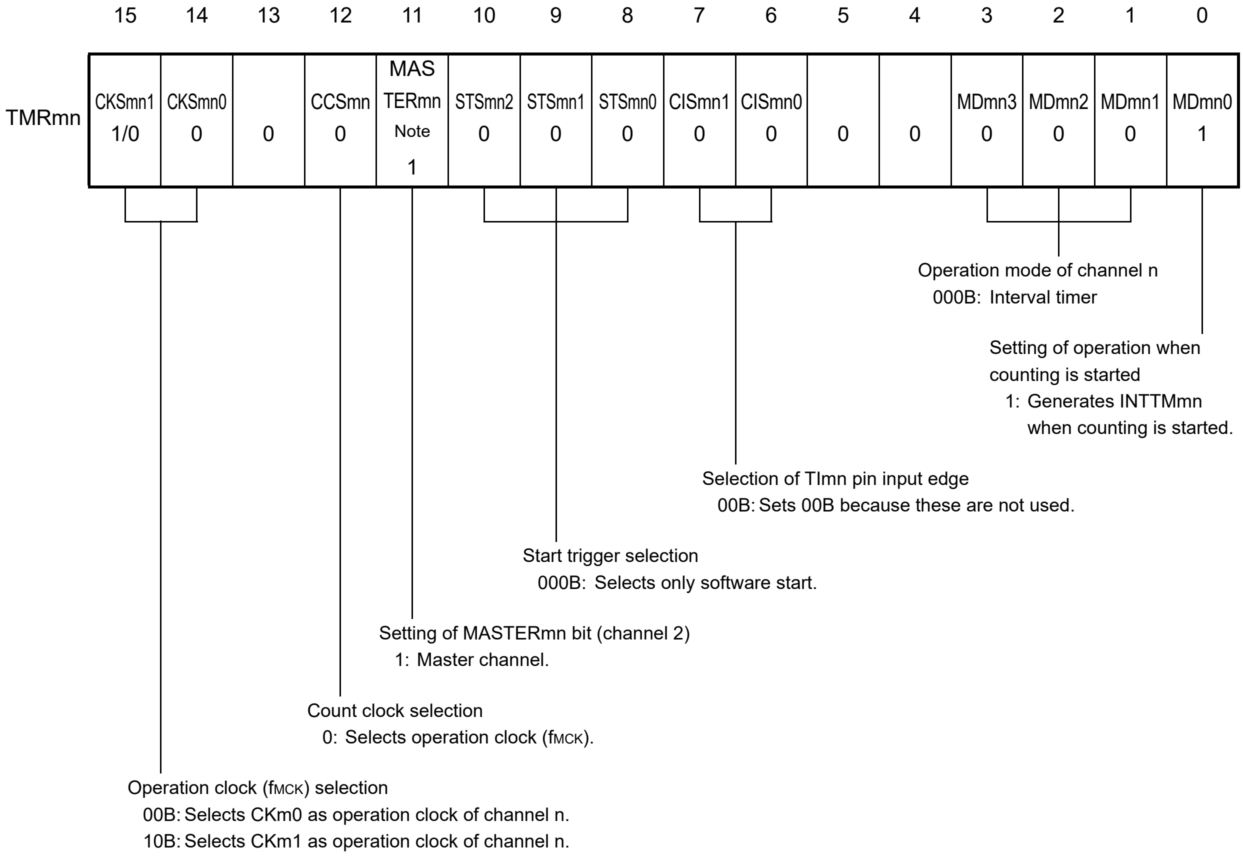
(**Remarks** are listed on the next page.)

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Remark 2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 6 - 84 Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MASTERmn Note 1	STSmn2 0	STSmn1 0	STSmn0 0	CISmn1 0	CISmn0 0	0	0	MDmn3 0	MDmn2 0	MDmn1 0	MDmn0 1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (fMCK).

Operation clock (fMCK) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0)

**Figure 6 - 85 Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs)**

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1 1/0	CKSmp0 0	0	CCSmp 0	M/S Note 0	STSmp2 1	STSmp1 0	STSmp0 0	CISmp1 0	CISmp0 0	0	0	MDmp3 1	MDmp2 0	MDmp1 0	MDmp0 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1 1/0	CKSmq0 0	0	CCSmq 0	M/S Note 0	STSmq2 1	STSmq1 0	STSmq0 0	CISmq1 0	CISmq0 0	0	0	MDmq3 1	MDmq2 0	MDmq1 0	MDmq0 1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTER0p, MASTER0q bit (channel 2)
0: Slave channel.
Setting of SPLIT0p, SPLIT0q bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq 1/0	TOmp 1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq 1/0	TOEmp 1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq 1/0	TOLmp 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq 1	TOMmp 1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLITmq bit

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 6 - 86 Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock for timer array unit 0 is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0.	The TOmp and TOmq pins go into Hi-Z output state.
	Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

(**Remark** is listed on the next page.)

Figure 6 - 87 Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. → The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. → When holding the TOmp and TOmq pin output levels are not necessary Setting not required	The TOmp and TOmq pin output levels are held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit 0 is stopped. All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 3$ (Where p and q are integer greater than n)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions when using timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

(1) Using TO03 output assigned to the P31 for 30 to 32-pin products

So that the alternated PCLBUZ0 output becomes 0, not only set the port mode register (the PM31 bit) and the port register (the P31 bit) to 0, but also use the bit 7 of the clock output select register 0 (CKS0) with the same setting as the initial status.

CHAPTER 7 TIMER RJ

7.1 Functions of Timer RJ

Timer RJ is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events.

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and they can be accessed by accessing the TRJ0 register.

Table 7 - 1 lists the Timer RJ Specifications. Figure 7 - 1 shows the Timer RJ Block Diagram.

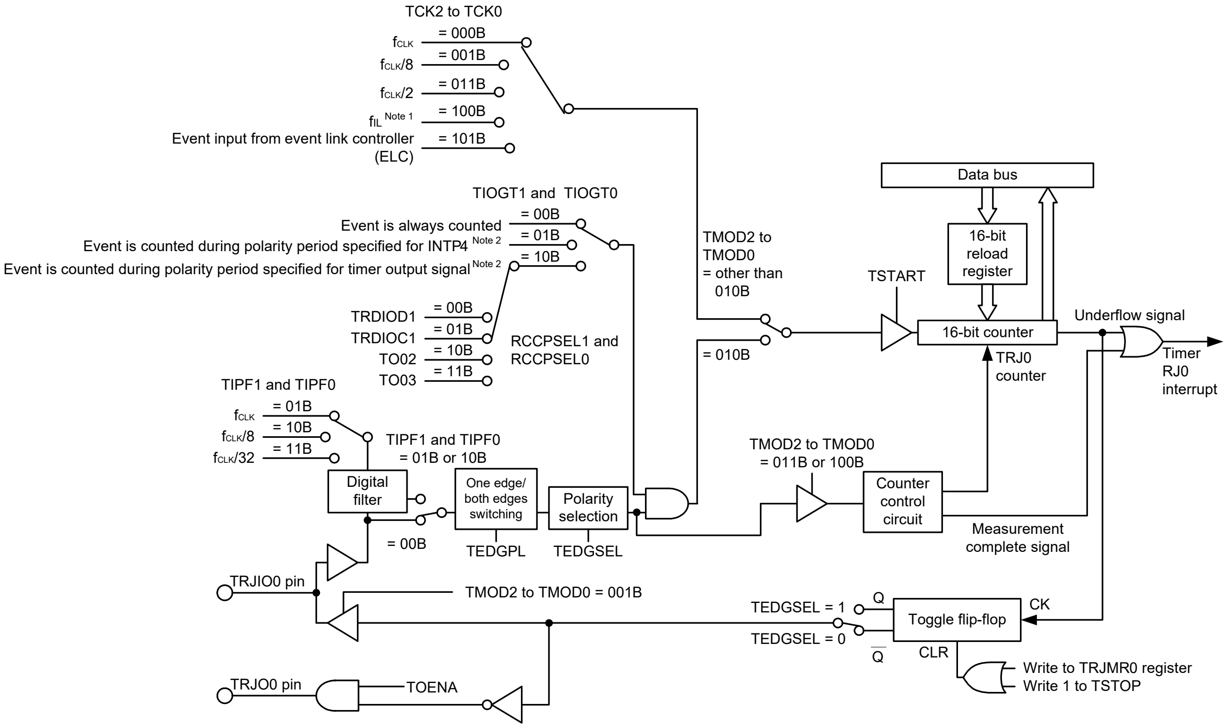
Table 7 - 1 Timer RJ Specifications

Item		Description
Operating modes	Timer mode	The count source is counted.
	Pulse output mode	The count source is counted and the output is inverted at each underflow of the timer.
	Event counter mode	An external event is counted. Operation is possible in STOP mode.
	Pulse width measurement mode	An external pulse width is measured.
	Pulse period measurement mode	An external pulse period is measured.
Count source (Operating clock)		f_{CLK}, $f_{CLK}/2$, $f_{CLK}/8$, f_{IL}, or event input from the event link controller (ELC) selectable
Interrupt		• When the counter underflows. • When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode. • When the set edge of the external input (TRJIO0) is input in pulse period measurement mode.
Selectable functions		• Coordination with the event link controller (ELC). Event input from the ELC is selectable as a count source.

7.2 Configuration of Timer RJ

Figure 7 - 1 shows the Timer RJ Block Diagram and Table 7 - 2 lists the Timer RJ Pin Configuration.

Figure 7 - 1 Timer RJ Block Diagram



TSTART, TSTOP: Bits in TRJCR0 register
TEDGSEL, TOENA, TIPF0, TIPF1, TIOGT0, TIOGT1: Bits in TRJIOC0 register
TMOD0 to TMOD2, TEDGPL, TCK0 to TCK2: Bits in TRJMR0 register
RCCPSEL0, RCCPSEL1: Bits in TRJISR0 register

Note 1. When selecting fIL as the count source, set the WUTMMCK0 bit in the operation speed mode control register (OSMC) to 1.

Note 2. The polarity can be selected by the RCCPSEL2 bit in the TRJISR0 register.

Table 7 - 2 Timer RJ Pin Configuration

Pin Name	I/O	Function
INTP4	Input	External input for timer RJ
TRJIO0 Note	Input/output	External event input and pulse output for timer RJ
TRJO0 Note	Output	Pulse output for timer RJ

Note The assignment of the TRJIO0 pin is selected by bits PIOR12 and PIOR13 in the PIOR1 register. The assignment of the TRJO0 pin is selected by bits PIOR10 and PIOR11 in the PIOR1 register. Refer to **CHAPTER 4 PORT FUNCTIONS** for details.

7.3 Registers

Table 7 - 3 lists the Timer RJ Register Configuration.

Table 7 - 3 Timer RJ Register Configuration

Register Name	Symbol	After Reset	Address	Access Size
Peripheral I/O redirection register 1	PIOR1	00H	F0079H	8
Peripheral enable register 1	PER1	00H	F007AH	8
Operation speed mode control register	OSMC	00H	F00F3H	8
Timer RJ counter register 0 Note	TRJ0	FFFFH	F0500H	16
Timer RJ control register 0	TRJCR0	00H	F0240H	8
Timer RJ I/O control register 0	TRJIOC0	00H	F0241H	8
Timer RJ mode register 0	TRJMR0	00H	F0242H	8
Timer RJ event pin select register 0	TRJISR0	00H	F0243H	8
Port register 0	P0	00H	FFF00H	8
Port register 3	P3	00H	FFF03H	8
Port register 4	P4	00H	FFF04H	8
Port register 5	P5	00H	FFF05H	8
Port mode register 0	PM0	FFH	FFF20H	8
Port mode register 3	PM3	FFH	FFF23H	8
Port mode register 4	PM4	FFH	FFF24H	8
Port mode register 5	PM5	FFH	FFF25H	8

Note When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

7.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
To use Timer RJ, be sure to set bit 0 (TRJ0EN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
PER1	TMKAEN	PWMOPEN	OACMPEN	TRD0EN	0	0	0	TRJ0EN

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

Caution 1. When setting timer RJ, be sure to set the TRJ0EN bit to 1 first. If TRJ0EN = 0, writing to a control register of timer RJ is ignored, and all read values are default values (except for port mode registers 0, 3, 4, 5 (PM0, PM3, PM4, PM5), and port registers 0, 3, 4, 5 (P0, P3, P4, P5)).

Caution 2. Be sure to set the following bits to 0:
Bits 1 to 3

7.3.2 Operation speed mode control register (OSMC)

The OSMC register can be used to select the low-speed on-chip oscillator as the 12-bit interval timer operating clock or the timer RJ count source.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 3 Format of Operation speed mode control register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0 Notes 1, 2	0	0	0	0

WUTMMCK0 Notes 1, 2	Selection of low-speed on-chip oscillator as 12-bit interval timer operating clock or timer RJ count source
0	• The low-speed on-chip oscillator cannot be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator cannot be selected as the timer RJ count source.
1	• The low-speed on-chip oscillator can be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator can be selected as the timer RJ count source.

Note 1. When using the 12-bit interval timer, be sure to set the WUTMMCK0 bit to 1.

Note 2. When the 12-bit interval timer is operating, do not set the WUTMMCK0 bit to 0.

Caution Be sure to clear bits 0 to 3 and bits 5 to 7 to 0.

7.3.3 Timer RJ counter register 0 (TRJ0)

TRJ0 is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter are changed depending on the TSTART bit in the TRJCR0 register. For details, see **7.4.1 Reload register and counter rewrite operation**.

The TRJ0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to FFFFH.

Figure 7 - 4 Format of Timer RJ counter register 0 (TRJ0)

Address: F0500H After Reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRJ0																

—	Function	Setting Range
Bits 15 to 0	16-bit counter Notes 1, 2	0000H to FFFFH

Note 1. When 1 is written to the TSTOP bit in the TRJCR0 register, the 16-bit counter is forcibly stopped and set to FFFFH.

Note 2. When the setting of bits TCK2 to TCK0 in the TRJMR0 register is other than 001B (f_{CLK}/8) or 011B (f_{CLK}/2), if the TRJ0 register is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. However, the TRJO0 and TRJIO0 output is toggled.
When the TRJ0 register is set to 0000H in event counter mode, regardless of the value of bits TCK2 to TCK0, a request signal to the ELC is generated only once immediately after the count starts.
In addition, the TRJO0 output is toggled even during a period other than the specified count period.
When the TRJ0 register is set to 0000H or a higher value, a request signal is generated each time TRJ underflows.

Caution **When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.**

7.3.4 Timer RJ control register 0 (TRJCR0)

The TRJCR0 register starts or stops count operation and indicates the status of timer RJ.

The TRJCR0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 5 Format of Timer RJ control register 0 (TRJCR0)

Address: F0240H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJCR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

TUNDF	Timer RJ underflow flag
0	No underflow
1	Underflow
[Condition for setting to 0] • When 0 is written to this bit by a program. [Condition for setting to 1] • When the counter underflows.	

TEDGF	Active edge judgement flag
0	No active edge received
1	Active edge received
[Condition for setting to 0] • When 0 is written to this bit by a program. [Conditions for setting to 1] • When the measurement of the active width of the external input (TRJIO) is completed in pulse width measurement mode. • The set edge of the external input (TRJIO) is input in pulse period measurement mode.	

TSTOP	Timer RJ count forced stop Note 1
When 1 is written to this bit, the count is forcibly stopped. The read value is 0.	

TCSTF	Timer RJ count status flag Note 2
0	Count stops
1	Count in progress
[Conditions for setting to 0] • When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source). • When 1 is written to the TSTOP bit. [Condition for setting to 1] • When 1 is written to the TSTART bit (the TCSTF bit is set to 1 in synchronization with the count source).	

TSTART	Timer RJ count start Note 2
0	Count stops
1	Count starts
Count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF bit is set to 0 (count stops) in synchronization with the count source. For details, see **7.5.1 Count operation start and stop control**.	

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, bits TSTART and TCSTF are initialized at the same time. The pulse output level is also initialized.

Note 2. For notes on using bits TSTART and TCSTF, see **7.5.1 Count operation start and stop control**.

7.3.5 Timer RJ I/O control register 0 (TRJIOC0)

The TRJIOC0 register sets the input/output of timer RJ.

The TRJIOC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 6 Format of Timer RJ I/O control register 0 (TRJIOC0)

Address: F0241H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	TRJIO count control Notes 1, 2
0	0	Event is always counted
0	1	Event is counted during polarity period specified for INTP4
1	0	Event is counted during polarity period specified for timer output signal
Other than above		Setting prohibited

TIPF1	TIPF0	TRJIO input filter select
0	0	No filter
0	1	Filter sampled at f_{CLK}
1	0	Filter sampled at $f_{CLK}/8$
1	1	Filter sampled at $f_{CLK}/32$
These bits are used to specify the sampling frequency of the filter for the TRJIO input. If the input to the TRJIO0 pin is sampled and the value matches three successive times, that value is taken as the input value.		

TOENA	TRJO output enable
0	TRJO output disabled (port)
1	TRJO output enabled

TEDGSEL	I/O polarity switch
Function varies depending on the operating mode (see **Tables 7 - 4** and **7 - 5**).	

Note 1. When INTP4 or the timer output signal is used, the polarity to count an event can be selected by the RCCPSEL2 bit in the TRJISR0 register.

Note 2. Bits TIOGT0 and TIOGT1 are enabled only in event counter mode.

Table 7 - 4 TRJIO I/O Edge and Polarity Switching

Operating Mode	Function
Timer mode	Not used (I/O port)
Pulse output mode	0: Output is started at high (Initialization level: High) 1: Output is started at low (Initialization level: Low)
Event counter mode	0: Count at rising edge 1: Count at falling edge
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge

Table 7 - 5 TRJO Output Polarity Switching

Operating Mode	Function
All modes	0: Output is started at low (Initialization level: Low) 1: Output is started at high (Initialization level: High)

7.3.6 Timer RJ mode register 0 (TRJMR0)

The TRJMR0 register sets the operating mode of timer RJ.

The TRJMR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 7 Format of Timer RJ mode register 0 (TRJMR0)

Address: F0242H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJMR0	0	TCK2	TCK1	TCK0	TEDGPL	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	Timer RJ count source select Notes 1, 2
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	f_{IL} Note 4
1	0	1	Event input from ELC
Other than above			Setting prohibited

TEDGPL	TRJIO edge polarity select Note 5
0	One edge
1	Both edges

TMOD2	TMOD1	TMOD0	Timer RJ operating mode select Note 3
0	0	0	Timer mode
0	0	1	Pulse output mode
0	1	0	Event counter mode
0	1	1	Pulse width measurement mode
1	0	0	Pulse period measurement mode
Other than above			Setting prohibited

Note 1. When event counter mode is selected, the external input (TRJIO) is selected as the count source regardless of the setting of bits TCK0 to TCK2.

Note 2. Do not switch count sources during count operation. Count sources should be switched when both the TSTART and TCSTF bits in the TRJCR0 register are set to 0 (count stops).

Note 3. The operating mode can be changed only when the count is stopped while both the bits TSTART and TCSTF in the TRJCR0 register are set to 0 (count stops). Do not change the operating mode during count operation.

Note 4. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the operation speed mode register (OSMC) to 1.

Note 5. The TEDGPL bit is enabled only in event counter mode.

Caution Write access to the TRJMR0 register initializes the output from pins TRJO0 and TRJIO0 of timer RJ. For details on the output level at initialization, refer to the description shown below Figure 7 - 6 Format of Timer RJ I/O control register 0 (TRJIOC0).

7.3.7 Timer RJ event pin select register 0 (TRJISR0)

The TRJISR0 register selects the timer for controlling the event count period and sets the polarity in event counter mode.

The TRJISR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 8 Format of Timer RJ event pin select register 0 (TRJISR0)

Address: F0243H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJISR0	0	0	0	0	0	RCCPSEL2 Note	RCCPSEL1 Note	RCCPSEL0 Note

RCCPSEL2 Note	Timer output signal and INTP4 polarity selection
0	An event is counted during the low-level period
1	An event is counted during the high-level period

RCCPSEL1 Note	RCCPSEL0 Note	Timer output signal selection
0	0	TRDIOD1
0	1	TRDIOC1
1	0	TO02
1	1	TO03

Note Bits RCCPSEL0 to RCCPSEL2 are enabled only in event counter mode.

7.3.8 Port mode registers 0, 3, 4, 5 (PM0, PM3, PM4, PM5)

These registers set input/output of ports 0, 3, 4, 5 in 1-bit units.
When using the ports (P01/TRJIO0, P30/TRJO0, etc.) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P01/TRJIO0 for timer output
Set the PM01 bit of port mode register 0 to 0.
Set the P01 bit of port register 0 to 0.

When using the ports (P01/TRJIO0, etc.) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P01/TRJIO0 for timer input
Set the PM01 bit of port mode register 0 to 1.
Set the P01 bit of port register 0 to 0 or 1.

The PM0, PM3, PM4, PM5 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Figure 7 - 9 Format of Port mode registers 0, 3, 4, 5 (PM0, PM3, PM4, PM5)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	1	PM01	PM00

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	1	PM41	PM40

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	1	1	1	1	PM51	PM50

PMmn	Pmn pin I/O mode selection (m = 0, 3; n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

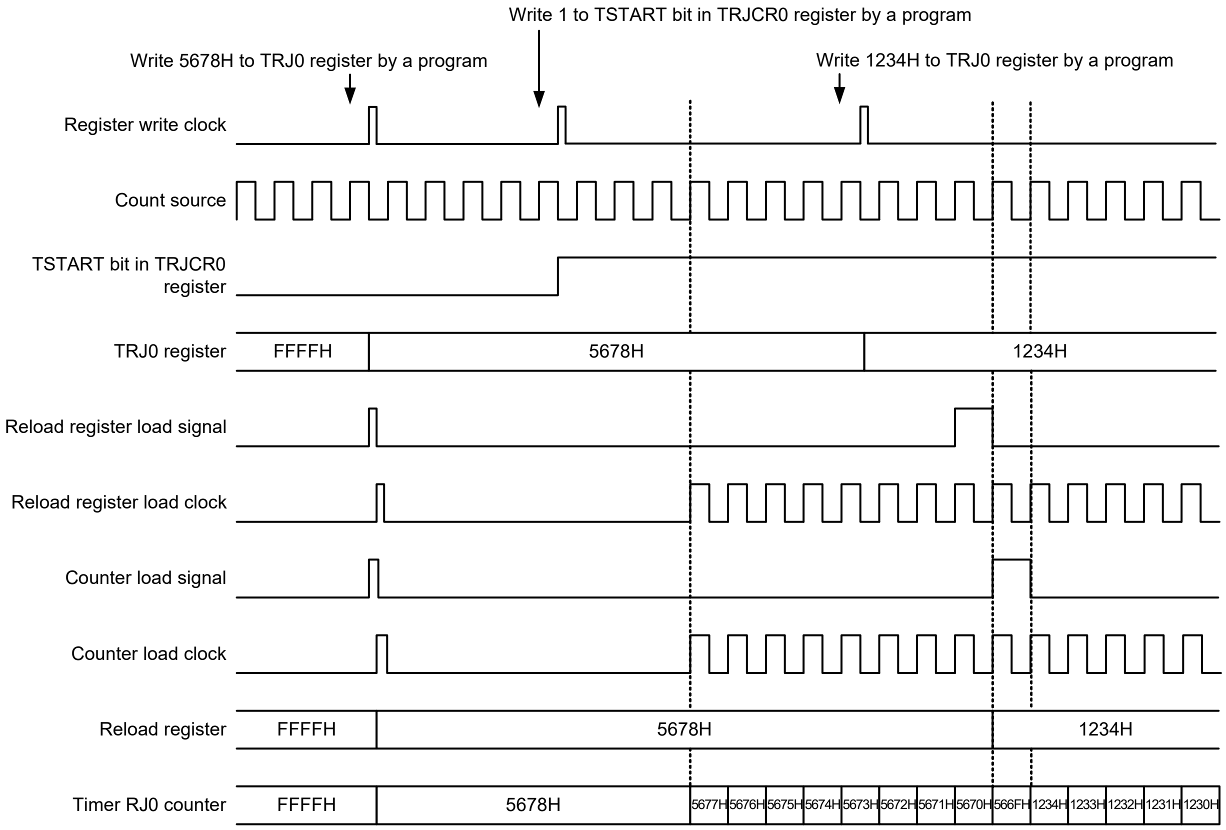
7.4 Operation

7.4.1 Reload register and counter rewrite operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value in the TSTART bit in the TRJCR0 register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source.

Figure 7 - 10 shows the Timing of Rewrite Operation with TSTART Bit Value.

Figure 7 - 10 Timing of Rewrite Operation with TSTART Bit Value



Write 1 to TSTART bit in TRJCR0 register by a program

Write 5678H to TRJ0 register by a program

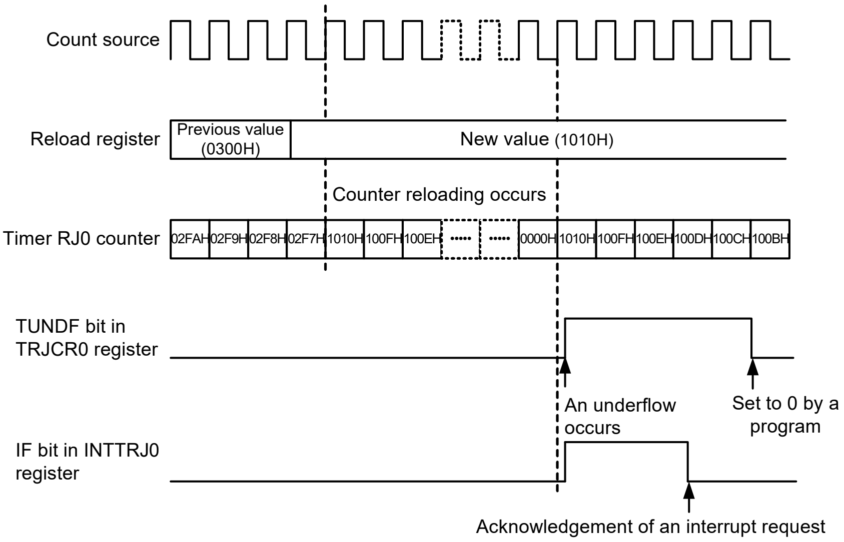
Write 1234H to TRJ0 register by a program

7.4.2 Timer mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register.

In timer mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated. Figure 7 - 11 shows the Operation Example in Timer Mode.

Figure 7 - 11 Operation Example in Timer Mode



7.4.3 Pulse output mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register, and the output level of pins TRJIO and TRJO pin is inverted each time an underflow occurs.

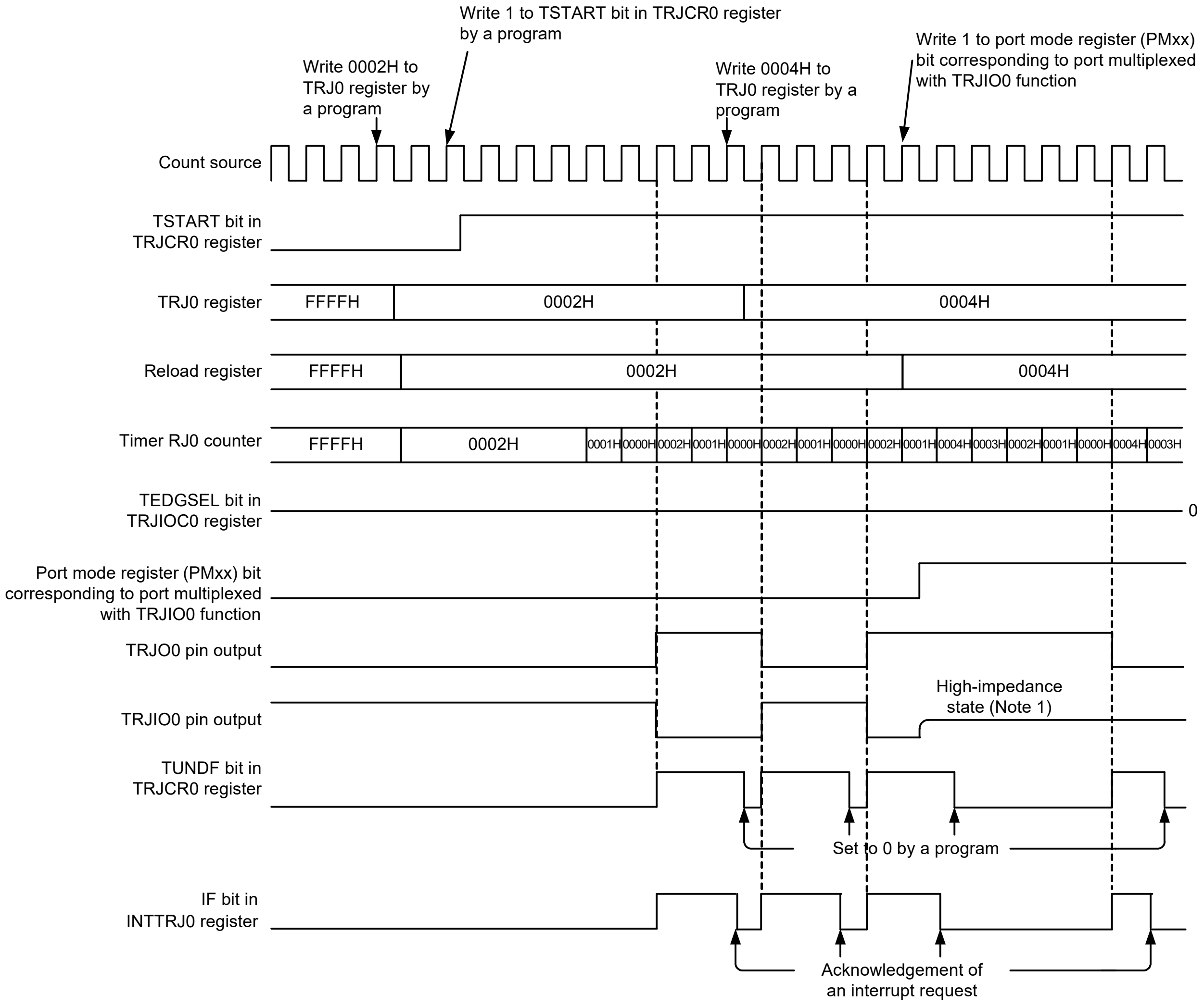
In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

In addition, a pulse can be output from pins TRJIO0 and TRJO0. The output level is inverted each time an underflow occurs. The pulse output from the TRJO0 pin can be stopped by the TOENA bit in the TRJIOC0 register.

Also, the output level can be selected by the TEDGSEL bit in the TRJIOC0 register.

Figure 7 - 12 shows the Operation Example in Pulse Output Mode.

Figure 7 - 12 Operation Example in Pulse Output Mode



Note 1: The TRJIO0 pin becomes high impedance by output enable control on the port selected as the TRJIO function

7.4.4 Event counter mode

In this mode, the counter is decremented by an external event signal (count source) input to the TRJIO0 pin. Various periods for counting events can be set by bits TIOGT0 and TIOGT1 in the TRJIOC0 register and the TRJISR0 register. In addition, the filter function for the TRJIO0 input can be specified by bits TIPF0 and TIPF1 in the TRJIOC0 register.

Also, the output from the TRJO0 pin can be toggled even in event counter mode.

When event counter mode is used, see **7.5.5 Procedure for setting pins TRJO0 and TRJIO0**.

Figure 7 - 13 shows the Operation Example 1 in Event Counter Mode.

Figure 7 - 13 Operation Example 1 in Event Counter Mode

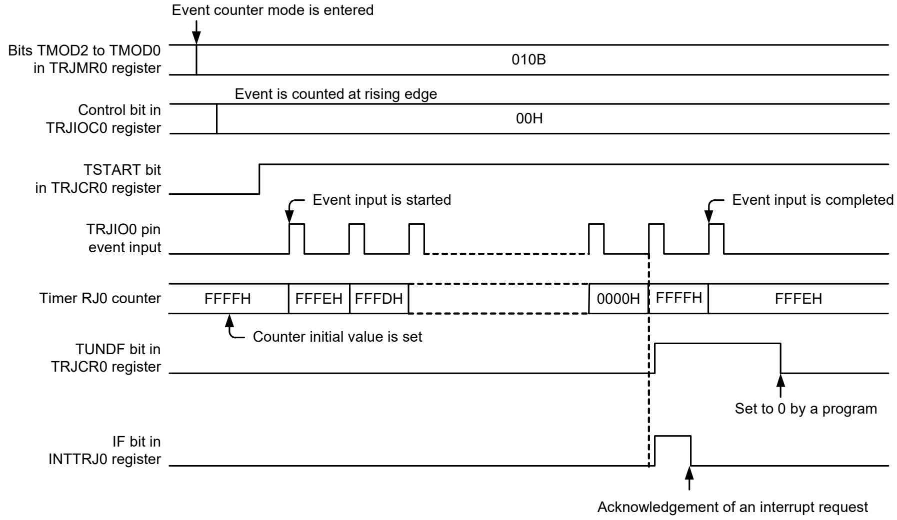
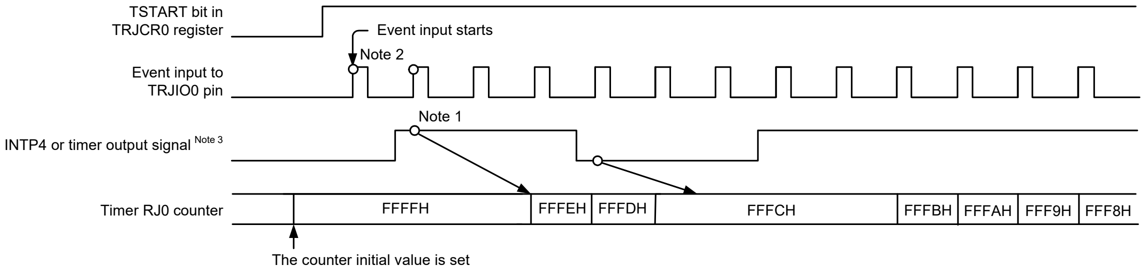


Figure 7 - 14 shows an operation example for counting during the specified period in event counter mode (bits TIOGT1 and TIOGT0 in the TRJIO0 register are set to 01B or 10B).

Figure 7 - 14 Operation Example 2 in Event Counter Mode

Timing example when the setting of operating mode is as follows:
TRJMR0 register: TMOD2, 1, 0 = 010B (event counter mode)
TRJIOC0 register: TIOGT1, 0 = 01B (event is counted during specified period for external interrupt pin)
TIPF1, 0 = 00B (no filter)
TEDGSEL = 0 (count at rising edge)
TRJISR0 register: RCCPSEL2 = 1 (high-level period is counted)



The following notes apply only when bits TIOGT1 and TIOGT0 in the TRJIOC0 register are 01B or 10B for the setting of operating mode in event count mode.

Note 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.

Note 2. Count operation may be performed for two cycles of the count source immediately after the count is started, depending on the previous state before the count is stopped.
To disable the count for two cycles immediately after the count is started, write 1 to the TSTOP bit in the TRJCR0 register to initialize the internal circuit, and then make operation settings before starting count operation.

Note 3. For the timer output signal selected by the RCCPSEL1 and RCCPSEL0 bits in the TRJISR0 register, the pin assigned to the timer output function cannot be used as the output of any multiplexed function other than the timer.

7.4.5 Pulse width measurement mode

In this mode, the pulse width of an external signal input to the TRJIO0 pin is measured.

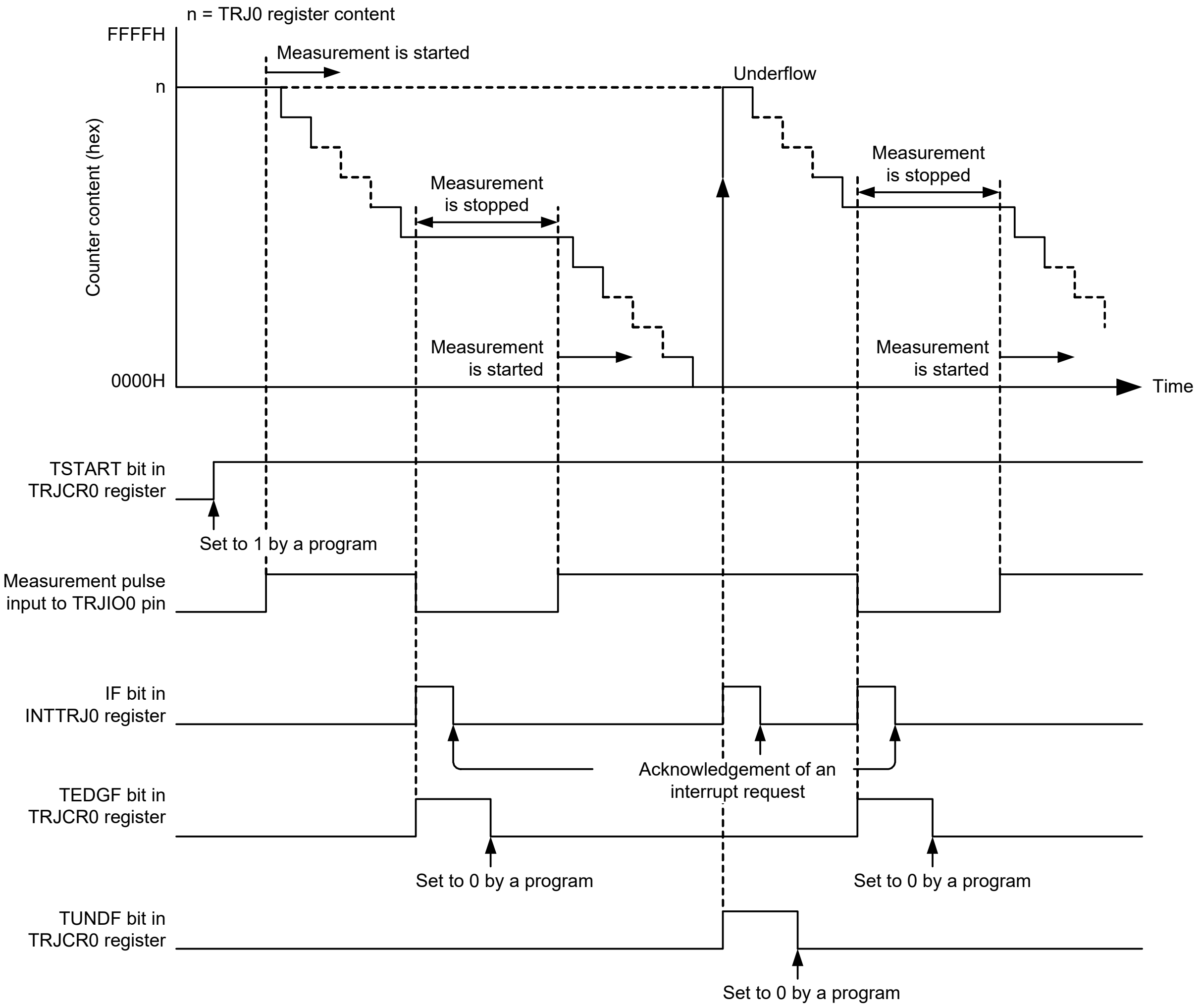
When the level specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the decrement is started with the selected count source. When the specified level on the TRJIO0 pin ends, the counter is stopped, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

Figure 7 - 15 shows the Operation Example in Pulse Width Measurement Mode.

When accessing bits TEDGF and TUNDF in the TRJCR0 register, see **7.5.2 Access to flags (bits TEDGF and TUNDF in TRJCR0 register)**.

Figure 7 - 15 Operation Example in Pulse Width Measurement Mode

This example applies when the high-level width of the measurement pulse is measured (TEDGSEL bit in TRJIOC0 register = 1)



7.4.6 Pulse period measurement mode

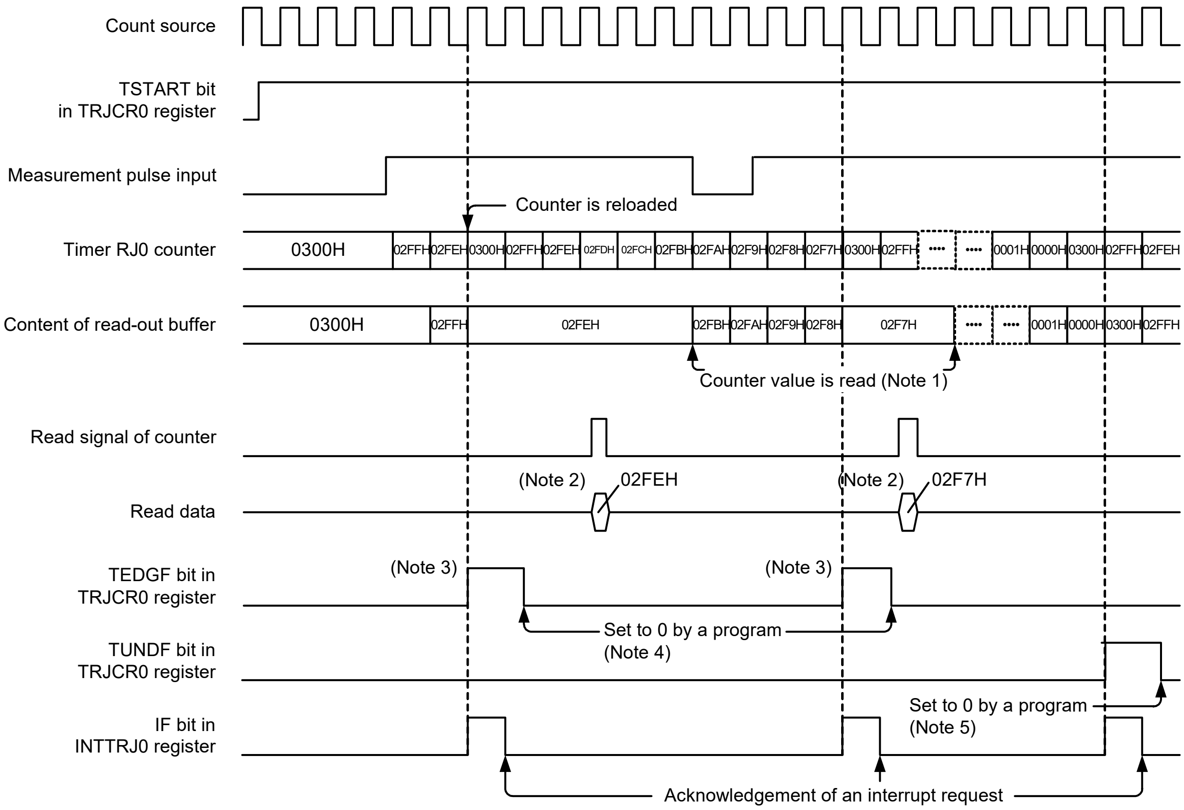
In this mode, the pulse period of an external signal input to the TRJIO0 pin is measured.

The counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register. When a pulse with the period specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the count value is transferred to the read-out buffer at the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (TRJ0 register) is read at this time and the difference from the reload value is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

Figure 7 - 16 shows the Operation Example in Pulse Period Measurement Mode.

Only input pulses with a period longer than twice the period of the count source. Also, the low-level and high-level widths must be both longer than the period of the count source. If a pulse period shorter than these conditions is input, the input may be ignored

Figure 7 - 16 Operation Example in Pulse Period Measurement Mode



This example applies when the initial value of the TRJ0 register is set to 0300H, the TEDGSEL bit in the TRJIOC0 register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the TRJ0 register must be performed during the period from when the TEDGF bit is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the TRJ0 register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.

Note 2. When the TRJ0 register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received).

Note 4. To set to 0 by a program, write 0 to the TEDGF bit in the TRJCR0 register using an 8-bit memory manipulation instruction.

Note 5. To set to 0 by a program, write 0 to the TUNDF bit in the TRJCR0 register using an 8-bit memory manipulation instruction.

7.4.7 Coordination with event link controller (ELC)

Through coordination with the ELC, event input from the ELC can be set to be the count source. Bits TCK0 to TCK2 in the TRJMR0 register count at the rising edge of event input from the ELC. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

- Procedure for starting operation

(1) Set the event output destination select register (ELSELRn) for the ELC.
(2) Set the operating mode for the event generation source.
(3) Set the mode for timer RJ.
(4) Start the count operation of timer RJ.
(5) Start the operation of the event generation source.

- Procedure for stopping operation

(1) Stop the operation of the event generation source.
(2) Stop the count operation of timer RJ.
(3) Set the event output destination select register (ELSELRn) for the ELC to 0.

7.4.8 Output settings for each mode

Table 7 - 6 and Table 7 - 7 list the states of pins TRJO0 and TRJIO0 in each mode.

Table 7 - 6 TRJO0 Pin Setting

Operating Mode	TRJIOC0 Register		TRJO0 Pin Output
	TOENA Bit	TEDGSEL Bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 7 - 7 TRJIO0 Pin Setting

Operating Mode	TRJIOC0 Register		TRJIO0 Pin I/O
	PMXX Bit Note	TEDGSEL Bit	
Timer mode	0 or 1	0 or 1	Input (Not used)
Pulse output mode	1	0 or 1	Output disabled (Hi-z output)
	0	1	Normal output
		0	Inverted output
Event counter mode	1	0 or 1	Input
Pulse width measurement mode			
Pulse period measurement mode			

Note The port mode register (PMxx) bit corresponding to port multiplexed with TRJIO0 function.

7.5 Notes on Timer RJ

7.5.1 Count operation start and stop control

- When event count mode is set or the count source is set to other than the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for three cycles of the count source. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 1 (count in progress).
After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for three cycles of the count source. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.
Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **CHAPTER 16 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

- When event count mode is set or the count source is set to the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for two cycles of the CPU clock. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 1 (count in progress).
After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for two cycles of the CPU clock. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.
Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **CHAPTER 16 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

7.5.2 Access to flags (bits TEDGF and TUNDF in TRJCR0 register)

Bits TEDGF and TUNDF in the TRJCR0 register are set to 0 by writing 0 by a program, but writing 1 to these bits has no effect. If a read-modify-write instruction is used to set the TRJCR0 register, bits TEDGF and TUNDF may be erroneously set to 0 depending on the timing, even when the TEDGF bit is set to 1 (active edge received) and the TUNDF bit is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCR0 register.

7.5.3 Access to counter register

When bits TSTART and TCSTF in the TRJCR0 register are both 1 (count starts), allow at least three cycles of the count source clock between writes when writing to the TRJ0 register successively.

7.5.4 When changing mode

The registers associated with timer RJ operating mode (TRJIOC0, TRJMR0, and TRJISR0) can be changed only when the count is stopped with both the TSTART and TCSTF bits set to 0 (count stops). Do not change these registers during count operation.
When the registers associated with timer RJ operating mode are changed, the values of bits TSTART and TCSTF are undefined. Write 0 (no active edge received) to the TEDGF bit and 0 (no underflow) to the TUNDF bit before starting the count.

7.5.5 Procedure for setting pins TRJO0 and TRJIO0

After a reset, the I/O ports multiplexed with pins TRJO0 and TRJIO0 function as input ports.

To output from pins TRJO0 and TRJIO0, use the following setting procedure:
Changing procedure

(1) Set the mode.
(2) Set the initial value/output enabled.
(3) Set the port register bits corresponding to pins TRJO0 and TRJIO0 to 0.
(4) Set the port mode register bits corresponding to pins TRJO0 and TRJIO0 to output mode.
(Output is started from pins TRJO0 and TRJIO0)
(5) Start the count (TSTART in TRJCR0 register = 1).

To input from the TRJIO0 pin, use the following setting procedure:

(1) Set the mode.
(2) Set the initial value/edge selected.
(3) Set the port mode register bit corresponding to TRJIO0 pin to input mode.
(Input is started from the TRJIO0 pin)
(4) Start the count (TSTART in TRJMR0 register = 1).
(5) Wait until the TCSTF bit in the TRJCR0 register is set to 1 (count in progress).
(In event counter mode only)
(6) Input an external event from the TRJIO0 pin.
(7) The processing on completion of the first measurement is invalid (the measured value is valid for the second and subsequent times). (In pulse width measurement mode and pulse period measurement mode only)

7.5.6 When timer RJ is not used

When timer RJ is not used, set bits TMOD2 to TMOD0 in the TRJMR0 register to 000B (timer mode) and set the TOENA bit in the TRJIOC0 register to 0 (TRJO output disabled).

7.5.7 When timer RJ operating clock is stopped

Supplying or stopping the timer RJ clock can be controlled by the TRJ0EN bit in the PER1 register. Note that the following SFRs cannot be accessed while the timer RJ clock is stopped. Make sure the timer RJ clock is supplied before accessing any of these registers.
Registers TRJ0, TRJCR0, TRJMR0, TRJIOC0, and TRJISR0.

7.5.8 Procedure for setting STOP mode (event counter mode)

To perform event counter mode operation during STOP mode, first supply the timer RJ clock and then use the following procedure to enter STOP mode.
Setting procedure

(1) Set the operating mode.
(2) Start the count (TSTART = 1, TCSTF = 1).
(3) Stop supplying the timer RJ clock.

To stop event counter mode operation during STOP mode, use the following procedure to stop operation.

(1) Supply the timer RJ clock.
(2) Stop the count (TSTART = 0, TCSTF = 0)

7.5.9 Functional restriction in STOP mode (event counter mode only)

When event counter mode operation is performed during STOP mode, the digital filter function cannot be used.

7.5.10 When count is forcibly stopped by TSTOP bit

After the counter is forcibly stopped by the TSTOP bit in the TRJCR0 register, do not access the following SFRs for one cycle of the count source.
Registers TRJ0, TRJCR0, and TRJMR0

7.5.11 Digital filter

When the digital filter is used, do not start timer operation for five cycles of the digital filter clock after setting bits TIPF1 and TIPF0.
Also, do not start timer operation for five cycles of the digital filter clock when the TEDGSEL bit in the TRJIOC register is changed while the digital filter is used.

7.5.12 When selecting f_{IL} as count source

When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the operation speed mode control register (OSMC) to 1.

CHAPTER 8 TIMER RD

8.1 Functions of Timer RD

Timer RD has four modes:

- Timer mode

- Input capture function	Transfer the counter value to a register with an external signal as the trigger
- Output compare function	Detect register value matches with a counter (Pin output can be changed at detection)
- PWM function	Output pulse of any width continuously

The following three modes use the PWM function.

• Reset synchronous PWM mode	Output three-phase waveforms (6) without sawtooth wave modulation and dead time
• Complementary PWM mode	Output three-phase waveforms (6) with triangular wave modulation and dead time
• PWM3 mode	Output PWM waveforms (2) with a fixed period

The timer mode input capture function, output compare function, and PWM function are equivalent in timer RD0 and timer RD1, and these functions can be selected individually for each pin. Also, a combination of these functions can be used in timer RD0 and timer RD1.

In reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, a waveform is output with a combination of counters and registers in timer RD0 and timer RD1. Pin functions depend on the mode.

Timer RD has eight I/O pins.

The operating clock for timer RD is f_{CLK} or f_{HOCO}.

Timer RD has a forced cutoff function.

The INTP0 interrupt signal can forcibly cut off output on the output pin of timer RD by fixing it to the high-Z, high output, or low output state.

When connected to the PWM option unit, the output signal from the comparator can be used to forcibly cut off output on the output pin of timer RD by placing it in the high-Z state.

When connected to the ELC, the output signal from the comparator can be used to forcibly cut off output on the output pin of timer RD by fixing it to the high-Z, high output, or low output state.

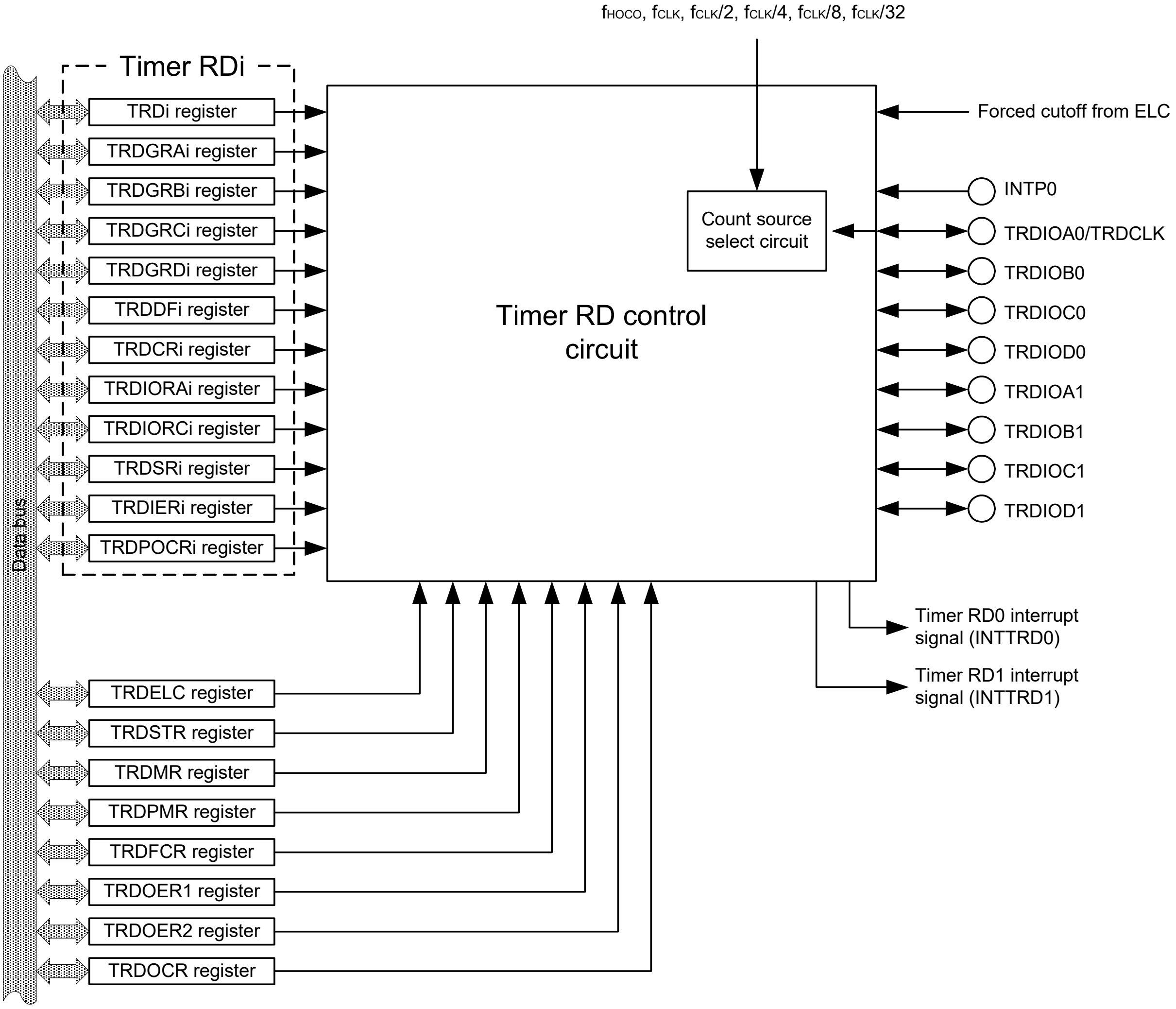
Furthermore, the forced cutoff is possible when the timer RD output pins are in use as port pins connecting to PWM option unit.

For details, see **8.4.4 Pulse Output Forced Cutoff**.

8.2 Configuration of Timer RD

Figure 8 - 1 shows the Timer RD Block Diagram and Table 8 - 1 lists the Timer RD Pin Configuration.

Figure 8 - 1 Timer RD Block Diagram



Remark
i = 0 or 1

Table 8 - 1 Timer RD Pin Configuration

Pin Name	Assigned Pin	I/O	Function
TRDIOA0/TRDCLK	P17	Input/Output	Function varies depending on the mode. Refer to descriptions of individual modes for details.
TRDIOB0	P15	Input/Output	
TRDIOC0	P16	Input/Output	
TRDIOD0	P14	Input/Output	
TRDIOA1	P13	Input/Output	
TRDIOB1	P12	Input/Output	
TRDIOC1	P11	Input/Output	
TRDIOD1	P10	Input/Output	

8.3 Registers

Table 8 - 2 lists the Timer RD Register Configuration.

Table 8 - 2 Timer RD Register Configuration

Register Name	Symbol	After Reset	Address	Access Size
Peripheral enable register 1	PER1	00H	F007AH	8
Timer RD ELC register	TRDELC	00H Note	F0260H	8
Timer RD start register	TRDSTR	0CH Note	F0263H	8
Timer RD mode register	TRDMR	00H Note	F0264H	8
Timer RD PWM function select register	TRDPMR	00H Note	F0265H	8
Timer RD function control register	TRDFCR	80H Note	F0266H	8
Timer RD output master enable register 1	TRDOER1	FFH Note	F0267H	8
Timer RD output master enable register 2	TRDOER2	00H Note	F0268H	8
Timer RD output control register	TRDOCR	00H Note	F0269H	8
Timer RD digital filter function select register 0	TRDDF0	00H Note	F026AH	8
Timer RD digital filter function select register 1	TRDDF1	00H Note	F026BH	8
Timer RD control register 0	TRDCR0	00H Note	F0270H	8
Timer RD I/O control register A0	TRDIORA0	00H Note	F0271H	8
Timer RD I/O control register C0	TRDIORC0	88H Note	F0272H	8
Timer RD status register 0	TRDSR0	00H Note	F0273H	8
Timer RD interrupt enable register 0	TRDIER0	00H Note	F0274H	8
Timer RD PWM function output level control register 0	TRDPOCR0	00H Note	F0275H	8
Timer RD counter 0	TRD0	0000H Note	F0276H	16
Timer RD general register A0	TRDGRA0	FFFFH Note	F0278H	16
Timer RD general register B0	TRDGRB0	FFFFH Note	F027AH	16
Timer RD general register C0	TRDGRC0	FFFFH Note	FFF58H	16
Timer RD general register D0	TRDGRD0	FFFFH Note	FFF5AH	16
Timer RD control register 1	TRDCR1	00H Note	F0280H	8
Timer RD I/O control register A1	TRDIORA1	00H Note	F0281H	8
Timer RD I/O control register C1	TRDIORC1	88H Note	F0282H	8
Timer RD status register 1	TRDSR1	00H Note	F0283H	8
Timer RD interrupt enable register 1	TRDIER1	00H Note	F0284H	8
Timer RD PWM function output level control register 1	TRDPOCR1	00H Note	F0285H	8
Timer RD counter 1	TRD1	0000H Note	F0286H	16
Timer RD general register A1	TRDGRA1	FFFFH Note	F0288H	16
Timer RD general register B1	TRDGRB1	FFFFH Note	F028AH	16
Timer RD general register C1	TRDGRC1	FFFFH Note	FFF5CH	16
Timer RD general register D1	TRDGRD1	FFFFH Note	FFF5EH	16
Port register 1	P1	00H	FFF01H	8
Port mode register 1	PM1	FFH	FFF21H	8

Note The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use timer RD, be sure to set bit 4 (TRD0EN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After Reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
PER1	TMKAEN	PWMOPEN	OACMPEN	TRD0EN	0	0	0	TRJ0EN

TRD0EN	Control of timer RD input clock supply
0	Stops input clock supply. • SFR used by timer RD cannot be written. • Timer RD is in the reset status.
1	Enables input clock supply. • SFR used by timer RD can be read and written.

Caution 1. When setting timer RD, be sure to set the TRD0EN bit to 1 first. If TRD0EN = 0, writing to a control register of timer RD is ignored, and all read values are default values (except for port mode register 1 (PM1), and port register 1 (P1)).

Caution 2. Be sure to set the following bits to 0:
Bits 1 to 3

Caution 3. When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

8.3.2 Timer RD ELC register (TRDELC)

Figure 8 - 3 Format of Timer RD ELC register (TRDELC)

Address: F0260H After Reset: 00H Note R/W

Symbol	7	6	5	4	3	2	1	0
TRDELC	0	0	ELCOBE1	ELCICE1	0	0	ELCOBE0	ELCICE0

ELCOBE1	ELC event input 1 enable for timer RD pulse output forced cutoff
0	Forced cutoff is disabled
1	Forced cutoff is enabled

ELCICE1	ELC event input 1 select for timer RD input capture D1
0	TRDIOD1 input capture is selected
1	Event input 1 from the event link controller (ELC) is selected

ELCOBE0	ELC event input 0 enable for timer RD pulse output forced cutoff
0	Forced cutoff is disabled
1	Forced cutoff is enabled

ELCICE0	ELC event input 0 select for timer RD input capture D0
0	TRDIOD0 input capture is selected
1	Event input 0 from the event link controller (ELC) is selected

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.3 Timer RD start register (TRDSTR)

The TRDSTR register can be set by an 8-bit memory manipulation instruction. See **8.7.1 (1) TRDSTR Register in the usage notes on timer RD**.

Figure 8 - 4 Format of Timer RD start register (TRDSTR)

Address: F0263H After Reset: 0CH Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDSTR	0	0	0	0	CSEL1	CSEL0	TSTART1	TSTART0

CSEL1	TRD1 count operation select Note 2
0	Count stops at compare match with TRDGRA1 register
1	Count continues after compare match with TRDGRA1 register Note 3

CSEL0	TRD0 count operation select
0	Count stops at compare match with TRDGRA0 register
1	Count continues after compare match with TRDGRA0 register Note 3

TSTART1	TRD1 count start flag Notes 4, 5
0	Count stops
1	Count starts

TSTART0	TRD0 count start flag Notes 6, 7
0	Count stops
1	Count starts

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Do not use in PWM3 mode.

Note 3. Set to 1 for the input capture function.

Note 4. Write 0 to the TSTART1 bit while the CSEL1 bit is set to 1.

Note 5. When the CSEL1 bit is 0 and a compare match signal (TRDIOA1) is generated, this flag is set to 0 (count stops).

Note 6. Write 0 to the TSTART0 bit while the CSEL0 bit is set to 1.

Note 7. When the CSEL0 bit is 0 and a compare match signal (TRDIOA0) is generated, this flag is set to 0 (count stops).

8.3.4 Timer RD mode register (TRDMR)

Figure 8 - 5 Format of Timer RD mode register (TRDMR)

Address: F0264H After Reset: 00H Note 1 R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
TRDMR	TRDBFD1	TRDBFC1	TRDBFD0	TRDBFC0	0	0	0	TRDSYNC

TRDBFD1	TRDGRD1 register function select Note 2
0	General register
1	Buffer register for TRDGRB1 register

TRDBFC1	TRDGRC1 register function select Note 2
0	General register
1	Buffer register for TRDGRA1 register

TRDBFD0	TRDGRD0 register function select Note 2
0	General register
1	Buffer register for TRDGRB0 register

TRDBFC0	TRDGRC0 register function select Notes 2, 3
0	General register
1	Buffer register for TRDGRA0 register

TRDSYNC	Timer RD synchronous Note 4
0	TRD0 and TRD1 operate independently
1	TRD0 and TRD1 operate synchronously

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. In the output compare function, if 0 (TRDGRji register output pin is changed) is selected for the IOj3 (j = C or D) bit in the TRDIORCi (i = 0 or 1) register, set the TRDBFji bit in the TRDMR register to 0.

Note 3. Set to 0 (general register) in complementary PWM mode.

Note 4. Set to 0 (TRD0 and TRD1 operate independently) in reset synchronous PWM mode, complementary PWM mode, and PWM3 mode.

8.3.5 Timer RD PWM function select register (TRDPMR)

Figure 8 - 6 Format of Timer RD PWM function select register (TRDPMR) [Timer Mode]

Address: F0265H After Reset: 00H Note R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
TRDPMR	0	TRDPWMD1	TRDPWMC1	TRDPWMB1	0	TRDPWMD0	TRDPWMC0	TRDPWMB0

TRDPWMD1	PWM function of TRDIOD1 select
0	Input capture function or output compare function
1	PWM function

TRDPWMC1	PWM function of TRDIOC1 select
0	Input capture function or output compare function
1	PWM function

TRDPWMB1	PWM function of TRDIOB1 select
0	Input capture function or output compare function
1	PWM function

TRDPWMD0	PWM function of TRDIOD0 select
0	Input capture function or output compare function
1	PWM function

TRDPWMC0	PWM function of TRDIOC0 select
0	Input capture function or output compare function
1	PWM function

TRDPWMB0	PWM function of TRDIOB0 select
0	Input capture function or output compare function
1	PWM function

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.6 Timer RD function control register (TRDFCR)

Figure 8 - 7 Format of Timer RD function control register (TRDFCR)

Address: F0266H After Reset: 80H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDFCR	PWM3	STCLK	0	0	OLS1	OLS0	CMD1	CMD0

PWM3	PWM3 mode select Note 2
• In the timer mode, set to 1 (other than PWM3 mode). • In PWM3 mode, set to 0 (PWM3 mode). • Disabled in reset synchronous and complementary PWM modes.	

STCLK	External clock input select
• In the timer mode, the reset synchronous PWM mode, and the complementary PWM mode, 0: External clock input disabled 1: External clock input enabled • In PWM3 mode, set to 0 (external clock input disabled).	

OLS1	Counter-phase output level select (in reset synchronous PWM mode or complementary PWM mode)
• In reset synchronous and complementary PWM modes, 0: High initial output and low active level 1: Low initial output and high active level • Disabled in timer and PWM3 modes.	

OLS0	Phase output level select (in reset synchronous PWM mode or complementary PWM mode)
• In reset synchronous and complementary PWM modes, 0: High initial output and low active level 1: Low initial output and high active level • Disabled in timer and PWM3 modes.	

CMD1	CMD0	Combination mode select Notes 3, 4
• In timer and PWM3 modes, set to 00B (timer mode or PWM3 mode). • In reset synchronous PWM mode, set to 01B (reset synchronous PWM mode). • In complementary PWM mode,		
CMD1	CMD0	
1	0	Complementary PWM mode (transfer from the buffer register to the general register when TRD1 underflows)
1	1	Complementary PWM mode (transfer from the buffer register to the general register at compare match between registers TRD0 and TRDGRA0)
Other than the above: Do not set.		

(**Notes** are listed on the next page.)

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. When bits CMD1 and CMD0 are set to 00B (timer mode or PWM3 mode), the setting of the PWM3 bit is enabled.

Note 3. Set bits CMD0 and CMD1 when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 0 (count stops).

Note 4. When bits CMD1 and CMD0 are set to 01B, 10B, or 11B, the MCU enters reset synchronous PWM mode or complementary PWM mode regardless of the settings of the TRDPMR register.

8.3.7 Timer RD output master enable register 1 (TRDOER1)

Figure 8 - 8 Format of Timer RD output master enable register 1 (TRDOER1) [Output Compare Function, PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]

Address: F0267H After Reset: FFH Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDOER1	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0

ED1	TRDIOD1 output disable Note 2
0	Output enabled
1	Output disabled (TRDIOD1 pin functions as an I/O port.)

EC1	TRDIOC1 output disable Note 2
0	Output enabled
1	Output disabled (TRDIOC1 pin functions as an I/O port.)

EB1	TRDIOB1 output disable Note 2
0	Output enabled
1	Output disabled (TRDIOB1 pin functions as an I/O port.)

EA1	TRDIOA1 output disable Notes 2, 3
0	Output enabled
1	Output disabled (TRDIOA1 pin functions as an I/O port)

ED0	TRDIOD0 output disable Note 2
0	Output enabled
1	Output disabled (TRDIOD0 pin functions as an I/O port.)

EC0	TRDIOC0 output disable Note 2
0	Output enabled
1	Output disabled (TRDIOC0 pin functions as an I/O port.)

EB0	TRDIOB0 output disable
0	Output enabled
1	Output disabled (TRDIOB0 pin functions as an I/O port.)

EA0	TRDIOA0 output disable Notes 3, 4
0	Output enabled
1	Output disabled (TRDIOA0 pin functions as an I/O port)

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set to 1 in PWM3 mode.

Note 3. Set to 1 in PWM function.

Note 4. Set to 1 in reset synchronous PWM mode and complementary PWM mode.

8.3.8 Timer RD output master enable register 2 (TRDOER2)

Figure 8 - 9 Format of Timer RD output master enable register 2 (TRDOER2)
[PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]

Address: F0268H After Reset: 00H Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
TRDOER2	TRDPTO	0	0	0	0	0	0	TRDSHUTS

TRDPTO	INTP0 of pulse output forced cutoff signal input enabled Note 2
0	Pulse output forced cutoff input disabled
1	Pulse output forced cutoff input enabled (The TRDSHUTS bit is set to 1 when a low level is applied to the INTP0 pin.)

TRDSHUTS	Forced cutoff flag
0	Not forcibly cut off
1	Forcibly cut off
This bit is set to 1 when the pulse is forcibly cut off by an INTP0 or ELC input event. This bit is not automatically cleared. To stop the forced cutoff of the pulse, write 0 to this bit while the count is stopped (TSTARTi = 0). The pulse is also forcibly cut off when 1 is written to the TRDSHUTS bit in an enabled mode.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. See **8.4.4 Pulse Output Forced Cutoff**.

8.3.9 Timer RD output control register (TRDOCR)

Write to the TRDOCR register when bits TSTART0 and TSTART1 in the TRDSTR register are both 0 (count stops).

Figure 8 - 10 Format of Timer RD output control register (TRDOCR) [Output Compare Function]

Address: F0269H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TRDIOD1 initial output level select Note 2
0	Low initial output
1	High initial output

TOC1	TRDIOC1 initial output level select Note 2
0	Low initial output
1	High initial output

TOB1	TRDIOB1 initial output level select Note 2
0	Low initial output
1	High initial output

TOA1	TRDIOA1 initial output level select
0	Low initial output
1	High initial output

TOD0	TRDIOD0 initial output level select Note 2
0	Low initial output
1	High initial output

TOC0	TRDIOC0 initial output level select Note 2
0	Low initial output
1	High initial output

TOB0	TRDIOB0 initial output level select Note 2
0	Low initial output
1	High initial output

TOA0	TRDIOA0 initial output level select
0	Low initial output
1	High initial output

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

Figure 8 - 11 Format of Timer RD output control register (TRDOCR) [PWM Function]

Address: F0269H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TRDIOD1 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level

TOC1	TRDIOC1 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level

TOB1	TRDIOB1 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level

TOA1	TRDIOA1 initial output level select
Set to 0.	

TOD0	TRDIOD0 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level

TOC0	TRDIOC0 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level
Enabled in reset synchronous and complementary PWM modes.	

TOB0	TRDIOB0 initial output level select Note 2
0	Initial output is not active level
1	Initial output is active level

TOA0	TRDIOA0 initial output level select
Set to 0.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

Figure 8 - 12 Format of Timer RD output control register (TRDOCR) [Reset Synchronous PWM Mode, Complementary PWM Mode]

Address: F0269H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1, TOC1, TOB1, TOA1, TOD0, TOB0, TOA0	Setting these bits to 1 is invalid in the reset synchronous PWM mode and complementary PWM mode. Be sure to set these bits to 0. In the reset synchronous PWM mode and complementary PWM mode, the setting of the OLS1 and OLS0 bits in TRDFCR determine the initial level independently of the setting in these bits.

TOC0	TRDIOC0 initial output level select Note 2
0	Initial output L
1	Initial output H
In the reset synchronous PWM mode, the output is inverted every PWM period. In complementary PWM mode, the output is inverted every 1/2 PWM period.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

Figure 8 - 13 Format of Timer RD output control register (TRDOCR) [PWM3 Mode]

Address: F0269H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TRDIOD1 initial output level select
Disabled in PWM3 mode.	

TOC1	TRDIOC1 initial output level select
Disabled in PWM3 mode.	

TOB1	TRDIOB1 initial output level select
Disabled in PWM3 mode.	

TOA1	TRDIOA1 initial output level select
Disabled in PWM3 mode.	

TOD0	TRDIOD0 initial output level select
Disabled in PWM3 mode.	

TOC0	TRDIOC0 initial output level select
Disabled in PWM3 mode.	

TOB0	TRDIOB0 initial output level select Note 2
0	Low initial output, high active level, high output at TRDGRB1 compare match, and low output at TRDGRB0 compare match
1	High initial output, low active level, low output at TRDGRB1 compare match, and high output at TRDGRB0 compare match

TOA0	TRDIOA0 initial output level select
0	Low initial output, high active level, high output at TRDGRA1 compare match, and low output at TRDGRA0 compare match
1	High initial output, low active level, low output at TRDGRA1 compare match, and high output at TRDGRA0 compare match

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

8.3.10 Timer RD digital filter function select register i (TRDDFi) (i = 0 or 1)

Figure 8 - 14 Format of Timer RD digital filter function select register i (TRDDFi) (i = 0 or 1) [Input Capture Function]

Address: F026AH (TRDDF0), F026BH (TRDDF1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	Clock select for digital filter function Note 2
0	0	$f_{CLK}/32$ Note 3
0	1	$f_{CLK}/8$ Note 3
1	0	f_{CLK} Note 3
1	1	Count source (clock selected by bits TCK0 to TCK2 in the TRDCRi register)

PENB1	PENB0	TRDIOB pin pulse forced cutoff control
0	0	Set to 00B.

DFD	TRDIODi pin digital filter function select
0	Digital filter function disabled
1	Digital filter function enabled
When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock.	

DFC	TRDIOCi pin digital filter function select
0	Digital filter function disabled
1	Digital filter function enabled
When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock.	

DFB	TRDIOBi pin digital filter function select
0	Digital filter function disabled
1	Digital filter function enabled
When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock.	

DFA	TRDIOAi pin digital filter function select
0	Digital filter function disabled
1	Digital filter function enabled
When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set bits DFCK0 and DFCK1 before starting count operation.

Note 3. When FRQSEL4 = 1 in the user option byte (000C2H), $f_{CLK}/32$, $f_{CLK}/8$, and f_{CLK} are set to $f_{HOCO}/32$, $f_{HOCO}/8$, and f_{HOCO}, respectively.

Figure 8 - 15 Format of Timer RD digital filter function select register i (TRDDFi) (i = 0 or 1) [PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]

Address: F026AH (TRDDF0), F026BH (TRDDF1) After Reset: 00H Note R/W

Symbol	7	6	5	4	3	2	1	0
TRDDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	TRDIOA pin pulse forced cutoff control
0	0	Forced cutoff disabled
0	1	High-impedance output
1	0	Low output
1	1	High output
Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped.		

PENB1	PENB0	TRDIOB pin pulse forced cutoff control
0	0	Forced cutoff disabled
0	1	High-impedance output
1	0	Low output
1	1	High output
Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped.		

DFD	DFC	TRDIOC pin pulse forced cutoff control
0	0	Forced cutoff disabled
0	1	High-impedance output
1	0	Low output
1	1	High output
Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped.		

DFB	DFA	TRDIOD pin pulse forced cutoff control
0	0	Forced cutoff disabled
0	1	High-impedance output
1	0	Low output
1	1	High output
Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped.		

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.11 Timer RD control register i (TRDCRi) (i = 0 or 1)

The TRDCR1 register is not used in reset synchronous PWM mode or PWM3 mode.

Figure 8 - 16 Format of Timer RD control register i (TRDCRi) (i = 0 or 1) [Input Capture Function and Output Compare Function]

Address: F0270H (TRDCR0), F0280H (TRDCR1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRDi counter clear select
0	0	0	Clear disabled (free-running operation)
0	0	1	Clear by input capture/compare match with TRDGRAi
0	1	0	Clear by input capture/compare match with TRDGRBi
0	1	1	Synchronous clear (clear simultaneously with other timer RDi counter) Note 2
1	0	1	Clear by input capture/compare match with TRDGRCi
1	1	0	Clear by input capture/compare match with TRDGRDi
Other than above			Setting prohibited

CKEG1	CKEG0	External clock edge select Note 3
0	0	Count at the rising edge
0	1	Count at the falling edge
1	0	Count at both edges
Other than above		Setting prohibited

TCK2	TCK1	TCK0	Count source select
0	0	0	f_{CLK}, f_{HOCO} Note 4
0	0	1	$f_{CLK}/2$ Note 5
0	1	0	$f_{CLK}/4$ Note 5
0	1	1	$f_{CLK}/8$ Note 5
1	0	0	$f_{CLK}/32$ Note 5
1	0	1	TRDCLK input Note 6
Other than above			Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Enabled when the TRDSYNC bit in the TRDMR register is 1 (TRD0 and TRD1 operate synchronously).

Note 3. Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

Note 4. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Note 5. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

Note 6. Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Figure 8 - 17 Format of Timer RD control register i (TRDCRi) (i = 0 or 1) [PWM Mode]

Address: F0270H (TRDCR0), F0280H (TRDCR1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRDi counter clear select
Set to 001B (TRDi register is cleared at compare match with TRDGRAi register).			

CKEG1	CKEG0	External clock edge select Note 2
0	0	Count at the rising edge
0	1	Count at the falling edge
1	0	Count at both edges
Other than above		Setting prohibited

TCK2	TCK1	TCK0	Count source select
0	0	0	f_{CLK}, f_{HOCO} Note 3
0	0	1	$f_{CLK}/2$ Note 4
0	1	0	$f_{CLK}/4$ Note 4
0	1	1	$f_{CLK}/8$ Note 4
1	0	0	$f_{CLK}/32$ Note 4
1	0	1	TRDCLK input Note 5
Other than above			Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

Note 3. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Note 4. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

Note 5. Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Figure 8 - 18 Format of Timer RD control register 0 (TRDCR0) [Reset Synchronous PWM Mode]

Address: F0270H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 counter clear select
Set to 001B (TRD0 register is cleared at compare match with TRDGRA0 register).			

CKEG1	CKEG0	External clock edge select Note 2
0	0	Count at the rising edge
0	1	Count at the falling edge
1	0	Count at both edges
Other than above		Setting prohibited

TCK2	TCK1	TCK0	Count source select
0	0	0	f_{CLK}, f_{HOCO} Note 3
0	0	1	$f_{CLK}/2$ Note 4
0	1	0	$f_{CLK}/4$ Note 4
0	1	1	$f_{CLK}/8$ Note 4
1	0	0	$f_{CLK}/32$ Note 4
1	0	1	TRDCLK input Note 5
Other than above			Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

Note 3. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Note 4. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

Note 5. Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Figure 8 - 19 Format of Timer RD control register i (TRDCRi) (i = 0 or 1) [Complementary PWM Mode]

Address: F0270H (TRDCR0), F0280H (TRDCR1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 counter clear select
Set to 000B (clear disabled (free-running operation)).			

CKEG1	CKEG0	External clock edge select Notes 2, 3
0	0	Count at the rising edge
0	1	Count at the falling edge
1	0	Count at both edges
Other than above		Setting prohibited

TCK2	TCK1	TCK0	Count source select
0	0	0	f_{CLK}, f_{HOCO} Note 4
0	0	1	$f_{CLK}/2$ Note 5
0	1	0	$f_{CLK}/4$ Note 5
0	1	1	$f_{CLK}/8$ Note 5
1	0	0	$f_{CLK}/32$ Note 5
1	0	1	TRDCLK input Note 6
Other than above			Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

Note 3. Set the same value to bits TCK0 to TCK2, CKEG0, and CKEG1 in registers TRDCR0 and TRDCR1.

Note 4. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Note 5. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

Note 6. Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Figure 8 - 20 Format of Timer RD control register 0 (TRDCR0) [PWM3 Mode]

Address: F0270H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TRD0 counter clear select
Set to 001B (TRD0 register is cleared at compare match with TRDGRA0 register).			

CKEG1	CKEG0	External clock edge select
Disabled in PWM3 mode.		

TCK2	TCK1	TCK0	Count source select
0	0	0	f_{CLK}, f_{HOCO} Note 2
0	0	1	$f_{CLK}/2$ Note 3
0	1	0	$f_{CLK}/4$ Note 3
0	1	1	$f_{CLK}/8$ Note 3
1	0	0	$f_{CLK}/32$ Note 3
Other than above			Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source, select f_{IH} as f_{CLK} before starting timer count operation.

Note 3. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

8.3.12 Timer RD I/O control register Ai (TRDIORAi) (i = 0 or 1)

Figure 8 - 21 Format of Timer RD I/O control register Ai (TRDIORAi) (i = 0 or 1) [Input Capture Function]

Address: F0271H (TRDIORA0), F0281H (TRDIORA1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TRDGRB mode select Note 2
Set to 1 (input capture) in the input capture function.	

IOB1	IOB0	TRDGRB control
0	0	Input capture to TRDGRBi at the rising edge
0	1	Input capture to TRDGRBi at the falling edge
1	0	Input capture to TRDGRBi at both edges
Other than above		Setting prohibited

IOA2	TRDGRA mode select Note 3
Set to 1 (input capture) in the input capture function.	

IOA1	IOA0	TRDGRA control
0	0	Input capture to TRDGRAi at the rising edge
0	1	Input capture to TRDGRAi at the falling edge
1	0	Input capture to TRDGRAi at both edges
Other than above		Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

Note 3. If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

Figure 8 - 22 Format of Timer RD I/O control register Ai (TRDIORAi) (i = 0 or 1) [Output Compare Function]

Address: F0271H (TRDIORA0), F0281H (TRDIORA1) After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TRDGRB mode select Note 2
Set to 0 (output compare) in the output compare function.	

IOB1	IOB0	TRDGRB control
0	0	Pin output by compare match is disabled (TRDIOBi pin functions as an I/O port)
0	1	Low output by compare match with TRDGRBi
1	0	High output by compare match with TRDGRBi
1	1	Toggle output by compare match with TRDGRBi

IOA2	TRDGRA mode select Note 3
Set to 0 (output compare) in the output compare function.	

IOA1	IOA0	TRDGRA control
0	0	Pin output by compare match is disabled (TRDIOAi pin functions as an I/O port)
0	1	Low output by compare match with TRDGRAi
1	0	High output by compare match with TRDGRAi
1	1	Toggle output by compare match with TRDGRAi

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

Note 3. If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

8.3.13 Timer RD I/O control register Ci (TRDIORCi) (i = 0 or 1)

Figure 8 - 23 Format of Timer RD I/O control register Ci (TRDIORCi) [Input Capture Function]

Address: F0272H (TRDIORC0), F0282H (TRDIORC1) After Reset: 88H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TRDGRD register function select
Set to 1 (general register or buffer register) in the input capture function.	

IOD2	TRDGRD mode select Note 2
Set to 1 (input capture) in the input capture function.	

IOD1	IOD0	TRDGRD control
0	0	Input capture to TRDGRDi at the rising edge
0	1	Input capture to TRDGRDi at the falling edge
1	0	Input capture to TRDGRDi at both edges
Other than above		Setting prohibited

IOC3	TRDGRC register function select
Set to 1 (general register or buffer register) in the input capture function.	

IOC2	TRDGRC mode select Note 3
Set to 1 (input capture) in the input capture function.	

IOC1	IOC0	TRDGRC control
0	0	Input capture to TRDGRCi at the rising edge
0	1	Input capture to TRDGRCi at the falling edge
1	0	Input capture to TRDGRCi at both edges
Other than above		Setting prohibited

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

Note 3. If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

Figure 8 - 24 Format of Timer RD I/O control register Ci (TRDIORCi) (i = 0 or 1) [Output Compare Function]

Address: F0272H (TRDIORC0), F0282H (TRDIORC1) After Reset: 88H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TRDGRD register function select
0	TRDIOB output register (see **8.5.2 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi**)
1	General register or buffer register

IOD2	TRDGRD mode select Note 2
Set to 0 (output compare) in the output compare function.	

IOD1	IOD0	TRDGRD control
0	0	Pin output by compare match is disabled
0	1	Low output by compare match with TRDGRDi
1	0	High output by compare match with TRDGRDi
1	1	Toggle output by compare match with TRDGRDi

IOC3	TRDGRC register function select
0	TRDIOA output register (see **8.5.2 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi**)
1	General register or buffer register

IOC2	TRDGRC mode select Note 3
Set to 0 (output compare) in the output compare function.	

IOC1	IOC0	TRDGRC control
0	0	Pin output by compare match is disabled
0	1	Low output by compare match with TRDGRCi
1	0	High output by compare match with TRDGRCi
1	1	Toggle output by compare match with TRDGRCi

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

Note 3. If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

8.3.14 Timer RD status register 0 (TRDSR0)

Figure 8 - 25 Format of Timer RD status register 0 (TRDSR0) [Input Capture Function]

Address: F0273H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	Overflow flag Note 2
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the TRD0 register overflows	

IMFD	Input capture/compare match flag D
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOD0 pin Note 4	

IMFC	Input capture/compare match flag C
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOC0 pin Note 4	

IMFB	Input capture/compare match flag B
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOB0 pin Note 5	

IMFA	Input capture/compare match flag A
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOA0 pin Note 5	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. When the counter value of timer RD0 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD0 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCR0 register, the overflow flag is set to 1.

Note 3. The writing results are as follows:

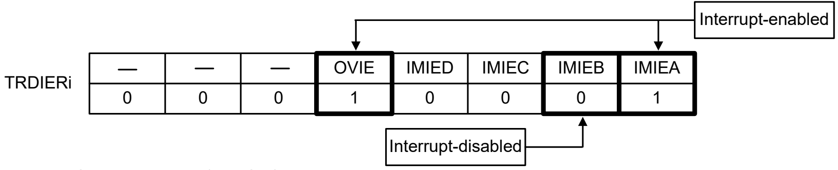
- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  When status flags of interrupt sources (applicable status flags) of timer RD are set to 0 and their interrupts are disabled in timer RD interrupt enable register i (TRDIERi), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

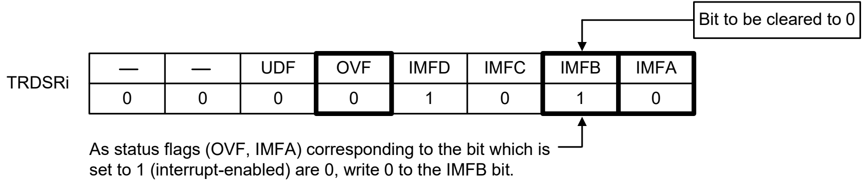
(b) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

Example: To clear the IMFB bit to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).

• Timer RD Interrupt Enable Register i (TRDIERi)



• Timer RD Status Register i (TRDSRi)

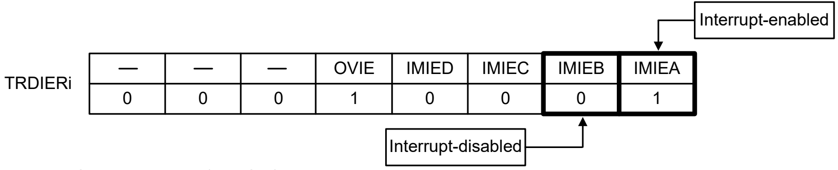


As status flags (OVF, IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) are 0, write 0 to the IMFB bit.

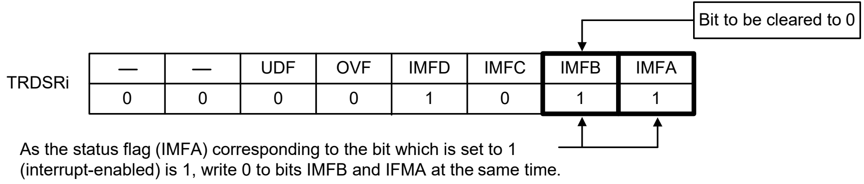
(c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the IMFB bit to 0 when the IMIEA bit is set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).

• Timer RD Interrupt Enable Register i (TRDIERi)



• Timer RD Status Register i (TRDSRi)



As the status flag (IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) is 1, write 0 to bits IMFB and IFMA at the same time.

Note 4. Edge selected by bits IOk1 and IOk0 (k = C or D) in the TRDIORC0 register.
Including when the TRDBFk0 bit in the TRDMR register is 1 (TRDGRk0 is buffer register).

Note 5. Edge selected by bits IOj1 and IOj0 (j = A or B) in the TRDIORA0 register.

Figure 8 - 26 Format of Timer RD status register 0 (TRDSR0) [Functions Other Than Input Capture Function]

Address: F0273H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	Overflow flag Note 3
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the TRD0 register overflows	

IMFD	Input capture/compare match flag D
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD0 and TRDGRD0 match. Note 4	

IMFC	Input capture/compare match flag C
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD0 and TRDGRC0 match. Note 4	

IMFB	Input capture/compare match flag B
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD0 and TRDGRB0 match.	

IMFA	Input capture/compare match flag A
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD0 and TRDGRA0 match.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

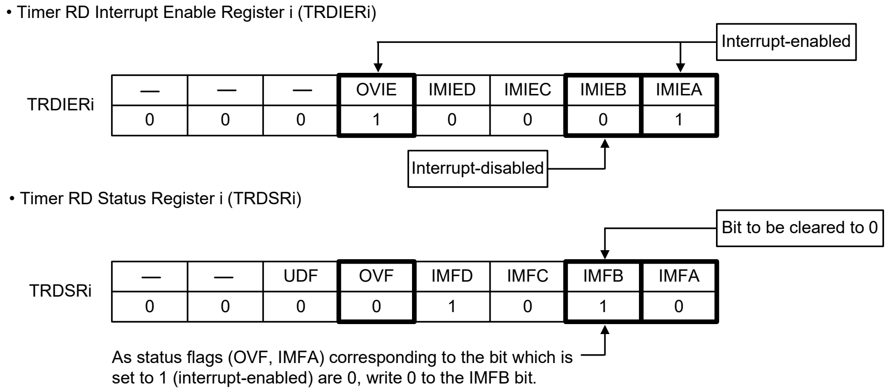
Note 2. The writing results are as follows:

- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  When status flags of interrupt sources (applicable status flags) of timer RD are set to 0 and their interrupts are disabled in timer RD interrupt enable register i (TRDIERi), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

(b) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

Example: To clear the IMFB bit to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



• Timer RD Interrupt Enable Register i (TRDIERi)

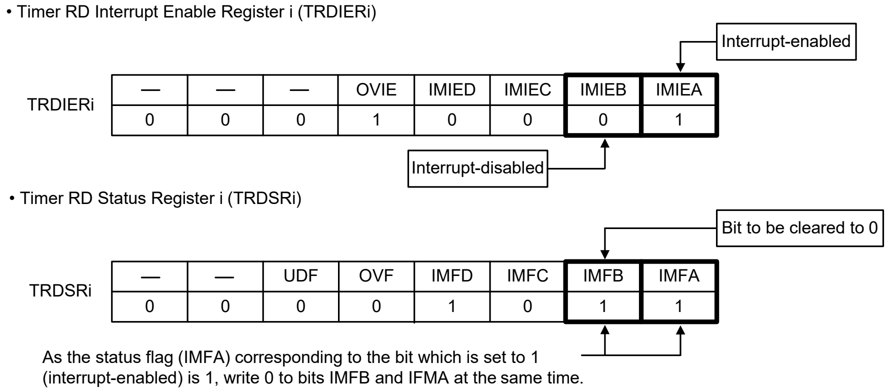
TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

• Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

(c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the IMFB bit to 0 when the IMIEA bit is set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



• Timer RD Interrupt Enable Register i (TRDIERi)

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

• Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

Note 3. When the counter value of timer RD0 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD0 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCR0 register, the overflow flag is set to 1.

Note 4. Including when the TRDBFk0 bit (k = C or D) in the TRDMR register is set to 1 (TRDGRK0 is buffer register).

8.3.15 Timer RD status register 1 (TRDSR1)

Figure 8 - 27 Format of Timer RD status register 1 (TRDSR1) [Input Capture Function]

Address: F0283H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	Underflow flag
Disabled in the input capture function.	

OVF	Overflow flag Note 2
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the TRD1 register overflows	

IMFD	Input capture/compare match flag D
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOD1 pin Note 4	

IMFC	Input capture/compare match flag C
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOC1 pin Note 4	

IMFB	Input capture/compare match flag B
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOB1 pin Note 5	

IMFA	Input capture/compare match flag A
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOA1 pin Note 5	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

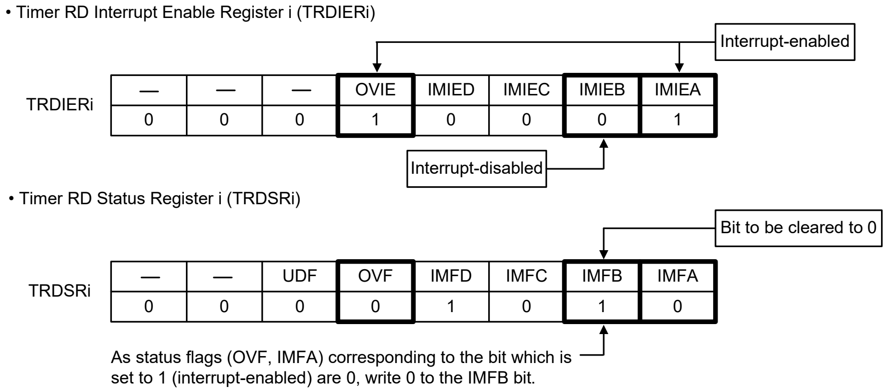
Note 2. When the counter value of timer RD1 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD1 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCR1 register, the overflow flag is set to 1.

Note 3. The writing results are as follows:

- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  When status flags of interrupt sources (applicable status flags) of timer RD are set to 0 and their interrupts are disabled in timer RD interrupt enable register i (TRDIERi), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

(b) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.
Example: To clear the IMFB bit to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



- Timer RD Interrupt Enable Register i (TRDIERi)

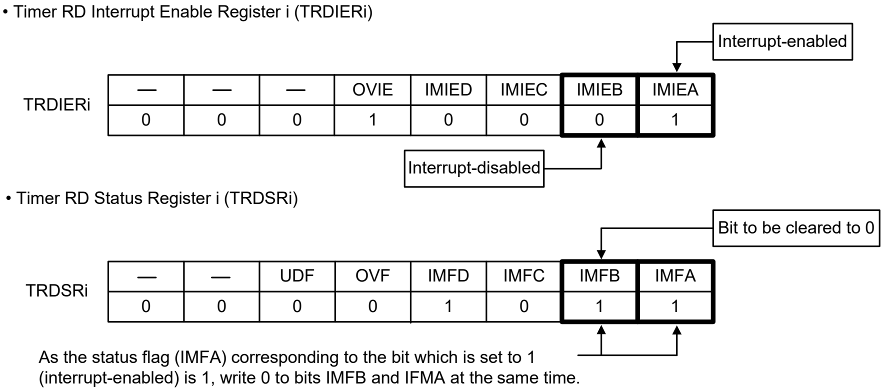
TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

- Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

As status flags (OVF, IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) are 0, write 0 to the IMFB bit.

(c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.
Example: To clear the IMFB bit to 0 when the IMIEA bit is set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



- Timer RD Interrupt Enable Register i (TRDIERi)

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

- Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

As the status flag (IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) is 1, write 0 to bits IMFB and IFMA at the same time.

Note 4. Edge selected by bits IOk1 and IOk0 (k = C or D) in the TRDIORC1 register.
Including when the TRDBFk1 bit in the TRDMR register is 1 (TRDGRk1 is buffer register).

Note 5. Edge selected by bits IOj1 and IOj0 (j = A or B) in the TRDIORA1 register.

Figure 8 - 28 Format of Timer RD status register 1 (TRDSR1) [Functions Other Than Input Capture Function]

Address: F0283H After Reset: 00H Note 1 R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	Underflow flag
In complementary PWM mode [Source for setting to 0] Write 0 after reading. Note 2 [Sources for setting to 1] When TRD1 underflows. Enabled only in complementary PWM mode.	

OVF	Overflow flag Note 3
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the TRD1 register overflows	

IMFD	Input capture/compare match flag D
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD1 and TRDGRD1 match. Note 4	

IMFC	Input capture/compare match flag C
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD1 and TRDGRC1 match. Note 4	

IMFB	Input capture/compare match flag B
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD1 and TRDGRB1 match.	

IMFA	Input capture/compare match flag A
[Source for setting to 0] Write 0 after reading. Note 2 [Source for setting to 1] When the values of TRD1 and TRDGRA1 match.	

Note 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. The writing results are as follows:

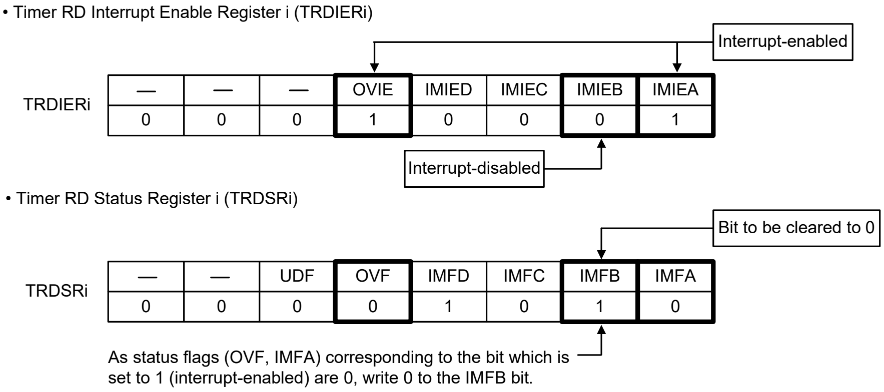
- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.

When status flags of interrupt sources (applicable status flags) of timer RD are set to 0 and their interrupts are disabled in timer RD interrupt enable register i (TRDIERi), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

(b) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

Example: To clear the IMFB bit to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



- Timer RD Interrupt Enable Register i (TRDIERi)

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

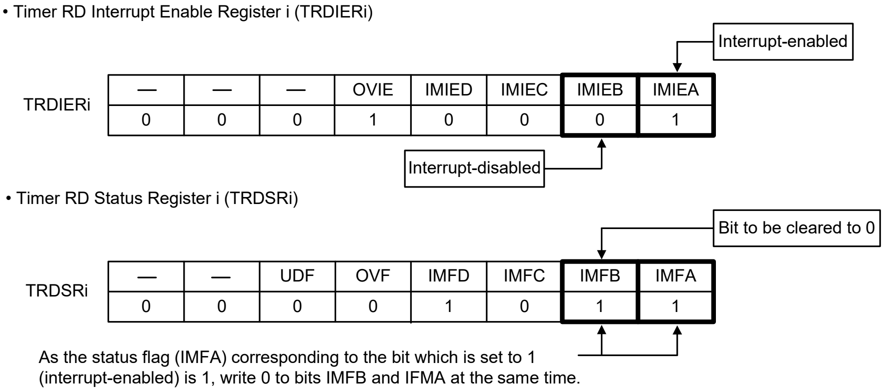
- Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

As status flags (OVF, IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) are 0, write 0 to the IMFB bit.

(c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the IMFB bit to 0 when the IMIEA bit is set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



- Timer RD Interrupt Enable Register i (TRDIERi)

TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

- Timer RD Status Register i (TRDSRi)

TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

As the status flag (IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) is 1, write 0 to bits IMFB and IFMA at the same time.

Note 3. When the counter value of timer RD1 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD1 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCR1 register, the overflow flag is set to 1.

Note 4. Including when the TRDBFk1 bit (k = C or D) in the TRDMR register is set to 1 (TRDGRK1 is buffer register).

8.3.16 Timer RD interrupt enable register i (TRDIERi) (i = 0 or 1)

Figure 8 - 29 Format of Timer RD interrupt enable register i (TRDIERi) (i = 0 or 1)

Address: F0274H (TRDIER0), F0284H (TRDIER1) After Reset: 00H Note R/W

Symbol	7	6	5	4	3	2	1	0
TRDIERi	0	0	0	OVIE	IMIED	IMIEC	IMIEB	IMIEA

OVIE	Overflow/underflow interrupt enable
0	Interrupt (OVI) by bits OVF and UDF disabled
1	Interrupt (OVI) by bits OVF and UDF enabled

IMIED	Input capture/compare match interrupt enable D
0	Interrupt (IMID) by the IMFD bit is disabled
1	Interrupt (IMID) by the IMFD bit is enabled

IMIEC	Input capture/compare match interrupt enable C
0	Interrupt (IMIC) by the IMFC bit is disabled
1	Interrupt (IMIC) by the IMFC bit is enabled

IMIEB	Input capture/compare match interrupt enable B
0	Interrupt (IMIB) by the IMFB bit is disabled
1	Interrupt (IMIB) by the IMFB bit is enabled

IMIEA	Input capture/compare match interrupt enable A
0	Interrupt (IMIA) by the IMFA bit is disabled
1	Interrupt (IMIA) by the IMFA bit is enabled

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.17 Timer RD PWM function output level control register i (TRDPOCRi) (i = 0 or 1)

Settings to the TRDPOCRi register are enabled only in PWM function. When not in PWM function, they are disabled.

Figure 8 - 30 Format of Timer RD PWM function output level control register i (TRDPOCRi) (i= 0 or 1) [PWM Function]

Address: F0275H (TRDPOCR0), F0285H (TRDPOCR1) After Reset: 00H Note R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOCRi	0	0	0	0	0	POLD	POLC	POLB

POLD	PWM function output level control D
0	TRDIODi output level is low active
1	TRDIODi output level is high active

POLC	PWM function output level control C
0	TRDIOCi output level is low active
1	TRDIOCi output level is high active

POLB	PWM function output level control B
0	TRDIOBi output level is low active
1	TRDIOBi output level is high active

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.18 Timer RD counter i (TRDi) (i = 0 or 1)

[Timer Mode]

Access the TRDi register in 16-bit units. Do not access it in 8-bit units.

[Reset Synchronous PWM Mode and PWM3 Mode]

Access the TRD0 register in 16-bit units. Do not access it in 8-bit units. The TRD1 register is not used in reset synchronous PWM mode and PWM3 mode.

[Complementary PWM Mode (TRD0)]

Access the TRD0 register in 16-bit units. Do not access it in 8-bit units.

[Complementary PWM Mode (TRD1)]

Access the TRD1 register in 16-bit units. Do not access it in 8-bit units.

Figure 8 - 31 Format of Timer RD counter i (TRDi) (i = 0 or 1) [Timer Mode]

Address: F0276H (TRD0), F0286H (TRD1) After Reset: 0000H [Note] R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDi																

—	Function	Setting Range
Bits 15 to 0	Count the count source. Count operation is incremented. When an overflow occurs, the OVF bit in the TRDSRi register is set to 1.	0000H to FFFFH

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 8 - 32 Format of Timer RD counter 0 (TRD0) [Reset Synchronous PWM Mode and PWM3 Mode]

Address: F0276H (TRD0) After Reset: 0000H [Note] R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0																

—	Function	Setting Range
Bits 15 to 0	Count the count source. Count operation is incremented. When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1.	0000H to FFFFH

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 8 - 33 Format of Timer RD counter 0 (TRD0) [Complementary PWM Mode (TRD0)]

Address: F0276H (TRD0) After Reset: 0000H [Note] R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0																

—	Function	Setting Range
Bits 15 to 0	Dead time must be set. Count the count source. Count operation is incremented or decremented. When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1.	0001H to FFFFH

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 8 - 34 Format of Timer RD counter 1 (TRD1) [Complementary PWM Mode (TRD1)]

Address: F0286H (TRD1) After Reset: 0000H Note R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD1																

—	Function	Setting Range
Bits 15 to 0	Set to 0000H. Count the count source. Count operation is incremented or decremented. When an underflow occurs, the UDF bit in the TRDSR1 register is set to 1.	0000H to FFFFH

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

8.3.19 Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1)

[input capture function]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in the input capture function:
TRDOER1, TRDOER2, TRDOCR, TRDPOCR0, and TRDPOCR1

Set the pulse width of the input capture signal applied to the TRDIOji pin to three or more cycles of the timer RD operating clock (f_{CLK}) when no digital filter is used (the DFj bit in the TRDDFi register is 0).

[Output Compare Function]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in the output compare function:
TRDDF0, TRDDF1, TRDPOCR0, and TRDPOCR1

[PWM Function]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in PWM function:
TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDIORA1, and TRDIORC1

[Reset Synchronous PWM Mode]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in reset synchronous PWM mode:
TRDPMR, TRDOCR Note, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

Note The TOC0 bit in the TRDOCR register is enabled as an initial output setting of TRDIOC0 in reset synchronous PWM mode and complementary PWM mode.

[Complementary PWM Mode]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
The TRDGRC0 register is not used in complementary PWM mode.

The following registers are disabled in complementary PWM mode.
TRDPMR, TRDOCR Note, TRDDF0 TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

Note The TOC0 bit in the TRDOCR register is enabled as an initial output setting of TRDIOC0 in reset synchronous PWM mode and complementary PWM mode.

Since values cannot be written to the TRDGRB0, TRDGRA1, or TRDGRB1 register directly after count operation starts (prohibited item), use the TRDGRD0, TRDGRC1, or TRDGRD1 register as a buffer register.
However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits TRDBFD0, TRDBFC1, and TRDBFD1 to 0 (general register). After this, bits TRDBFD0, TRDBFC1, and TRDBFD1 may be set to 1 (buffer register).

[PWM3 Mode]
Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in PWM3 mode:
TRDPMR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

Registers TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1 are not used in PWM3 mode. To use them as buffer registers, set bits TRDBFC0, TRDBFC1, TRDBFD0, and TRDBFD1 to 0 (general register) and write a value to the TRDGRC0, TRDGRC1, TRDGRD0, or TRDGRD1 register. After this, bits TRDBFC0, TRDBFC1, TRDBFD0, and TRDBFD1 may be set to 1 (buffer register).

Figure 8 - 35 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [Input Capture Function]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0),
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH Note R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 3 TRDGRji Register Functions in Input Capture Function**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 3 TRDGRji Register Functions in Input Capture Function

Register	Setting	Register Function	Input-Capture Input Pin
TRDGRAi	—	General register. The value of the TRDi register can be read at input capture.	TRDIOAi
TRDGRBi			TRDIOBi
TRDGRCi	TRDBFCi = 0	General register. The value of the TRDi register can be read at input capture.	TRDIOCi
TRDGRDi	TRDBFDi = 0		TRDIODi
TRDGRCi	TRDBFCi = 1	Buffer register. The value of the TRDi register can be read at input capture (see **8.4.2 Buffer Operation**).	TRDIOAi
TRDGRDi	TRDBFDi = 1		TRDIOBi

Remark i = 0 or 1, j = A, B, C, or D
TRDBFCi, TRDBFDi: Bits in TRDMR register

Figure 8 - 36 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [Output Compare Function]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0),
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH Note R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 4 TRDGRji Register Functions in Output Compare Function**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 4 TRDGRji Register Functions in Output Compare Function

<table>
<tr><th rowspan="2">Register</th><th colspan="2">Setting</th><th rowspan="2" colspan="2">Register Function</th><th rowspan="2">Output-Compare Output Pin</th></tr>
<tr><th>TRDBFki</th><th>IOj3</th></tr>
<tr><td>TRDGRAi</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2" colspan="2">General register. Write the compare value.</td><td>TRDIOAi</td></tr>
<tr><td>TRDGRBi</td><td>TRDIOBi</td></tr>
<tr><td>TRDGRCi</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2" colspan="2">General register. Write the compare value.</td><td>TRDIOCi</td></tr>
<tr><td>TRDGRDi</td><td>TRDIODi</td></tr>
<tr><td>TRDGRCi</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2" colspan="2">Buffer register. Write the next compare value (see 8.4.2 Buffer Operation).</td><td>TRDIOAi</td></tr>
<tr><td>TRDGRDi</td><td>TRDIOBi</td></tr>
<tr><td>TRDGRCi</td><td rowspan="2">0</td><td rowspan="2">0</td><td>TRDIOAi output control</td><td rowspan="2">(See 8.5.2 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi.)</td><td>TRDIOAi</td></tr>
<tr><td>TRDGRDi</td><td>TRDIOBi output control</td><td>TRDIOBi</td></tr>
</table>

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}) and the compare value is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

Remark i = 0 or 1, j = A, B, C, or D, k = C or D
TRDBFji: Bit in TRDMR register, IOj3: Bit in TRDIORCi register

Figure 8 - 37 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [PWM Mode]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), After Reset: FFFFH [Note] R/W
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 5 TRDGRji Register Functions in PWM Function**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 5 TRDGRji Register Functions in PWM Function

Register	Setting	Register Function	PWM Output Pin
TRDGRAi	—	General register. Set the PWM period.	—
TRDGRBi	—	General register. Set the changing point of PWM output.	TRDIOBi
TRDGRCi	TRDBFCi = 0	General register. Set the changing point of PWM output.	TRDIOCi
TRDGRDi	TRDBFDi = 0		TRDIODi
TRDGRCi	TRDBFCi = 1	Buffer register. Set the next PWM period (see **8.4.2 Buffer Operation**).	—
TRDGRDi	TRDBFDi = 1	Buffer register. Set the changing point of the next PWM output (see **8.4.2 Buffer Operation**).	TRDIOBi

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}) and the compare value is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

Remark i = 0 or 1, j = A, B, C, or D
TRDBFCi, TRDBFDi: Bits in TRDMR register

Figure 8 - 38 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [Reset Synchronous PWM Mode]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0),
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH Note R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 6 TRDGRji Register Functions in Reset Synchronous PWM Mode**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 6 TRDGRji Register Functions in Reset Synchronous PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	—	General register. Set the PWM period.	(TRDIOC0, output inverted every PWM period)
TRDGRB0	—	General register. Set the changing point of PWM1 output.	TRDIOB0 TRDIOD0
TRDGRC0	TRDBFC0 = 0	(Not used in reset synchronous PWM mode.)	—
TRDGRD0	TRDBFD0 = 0		
TRDGRA1	—	General register. Set the changing point of PWM2 output.	TRDIOA1 TRDIOC1
TRDGRB1	—	General register. Set the changing point of PWM3 output.	TRDIOB1 TRDIOD1
TRDGRC1	TRDBFC1 = 0	(Not used in reset synchronous PWM mode.)	—
TRDGRD1	TRDBFD1 = 0		
TRDGRC0	TRDBFC0 = 1	Buffer register. Set the next PWM period (see **8.4.2 Buffer Operation**).	(TRDIOC0, output inverted every PWM period)
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the changing point of the next PWM1 (see **8.4.2 Buffer Operation**).	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of the next PWM2 (see **8.4.2 Buffer Operation**).	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the changing point of the next PWM3 (see **8.4.2 Buffer Operation**).	TRDIOB1 TRDIOD1

Caution **When the setting of bits TCK2 to TCK0 in the TRDCR0 register is 000B (f_{CLK}, f_{HOCO}) and the compare value is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

Remark i = 0 or 1, j = A, B, C, or D
TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

Figure 8 - 39 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [Complementary PWM Mode]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0),
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH Note R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 7 TRDGRji Register Functions in Complementary PWM Mode**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 7 TRDGRji Register Functions in Complementary PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	—	General register. Set the PWM period at initialization. Setting range: ≥ Value set in TRD0 register (initial count value) ≤ FFFFh - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	(TRDIOC0, output inverted every half period)
TRDGRB0	—	General register. Set the changing point of PWM1 output at initialization. Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB0 TRDIOD0
TRDGRA1	—	General register. Set the changing point of PWM2 output at initialization. Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOA1 TRDIOC1
TRDGRB1	—	General register. Set the changing point of PWM3 output at initialization. Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB1 TRDIOD1
TRDGRC0	—	(Not used in complementary PWM mode.)	—
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the changing point of next PWM1 output (see **8.4.2 Buffer Operation**). Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRB0 register for initialization.	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of next PWM2 output (see **8.4.2 Buffer Operation**). Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRA1 register for initialization.	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the changing point of next PWM3 output (see **8.4.2 Buffer Operation**). Setting range: ≥ Value set in TRD0 register (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRB1 register for initialization.	TRDIOB1 TRDIOD1

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}) and the compare value is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

Remark i = 0 or 1, j = A, B, C, or D

TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

Figure 8 - 40 Format of Timer RD general registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [PWM3 Mode]

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), After Reset: FFFFH Note R/W
FFF58H (TRDGRC0), FFF5AH (TRDGRD0),
F0288H (TRDGRA1), F028AH (TRDGRB1),
FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDGRAi TRDGRBi TRDGRCi TRDGRDi																

—	Function
Bits 15 to 0	See **Table 8 - 8 TRDGRji Register Functions in PWM3 Mode**.

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 8 - 8 TRDGRji Register Functions in PWM3 Mode

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	—	General register. Set the PWM period. Setting range: ≥ Value set in TRDGRA1 register	TRDIOA0
TRDGRA1		General register. Set the changing point (active level timing) of PWM output Setting range: ≤ Value set in TRDGRA0 register	
TRDGRB0		General register. Set the changing point (the timing for returning to initial output level) of PWM output. Setting range: ≥ Value set in TRDGRB1 register and ≤ Value set in TRDGRA0 register	TRDIOB0
TRDGRB1		General register. Set the changing point (active level timing) of PWM output Setting range: ≤ Value set in TRDGRB0 register	
TRDGRC0	TRDBFC0 = 0	(Not used in PWM3 mode.)	—
TRDGRC1	TRDBFC1 = 0		
TRDGRD0	TRDBFD0 = 0		
TRDGRD1	TRDBFD1 = 0		
TRDGRC0	TRDBFC0 = 1	Buffer register. Set the next PWM period (see **8.4.2 Buffer Operation**). Setting range: ≤ Value set in TRDGRC1 register	TRDIOA0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of next PWM output (see **8.4.2 Buffer Operation**). Setting range: ≤ Value set in TRDGRC0 register	
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the changing point of next PWM output (see **8.4.2 Buffer Operation**). Setting range: ≥ Value set in TRDGRD1 register and ≤ Value set in TRDGRC0 register	TRDIOB0
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the changing point of next PWM output (see **8.4.2 Buffer Operation**). Setting range: ≤ Value set in TRDGRD0 register	

Caution **When the setting of bits TCK2 to TCK0 in the TRDCR0 register is 000B (f_{CLK}, f_{HOCO}) and the compare value is set to 0000H, a request signal to the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

Remark i = 0 or 1, j = A, B, C, or D

TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

8.3.20 Port mode register 1 (PM1)

This register sets input/output of port 1 in 1-bit units.
When using the ports (P10/TRDIOD1, P11/TRDIOC1, etc.) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

(Example) When using P10/TRDIOD1 for timer output
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 0.

When using the ports (P10/TRDIOD1, P11/TRDIOC1, etc.) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

(Example) When using P10/TRDIOD1 for timer input
Set the PM10 bit of port mode register 1 to 1.
Set the P10 bit of port register 1 to 0 or 1.

The PM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 8 - 41 Format of Port mode register 1 (PM1)

Address: FFF21H After Reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PMmn	Pmn pin I/O mode selection (m = 1; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Items Common to Multiple Modes

8.4.1 Count Sources

The count source selection method is the same in all modes. However, the external clock cannot be selected in PWM3 mode.

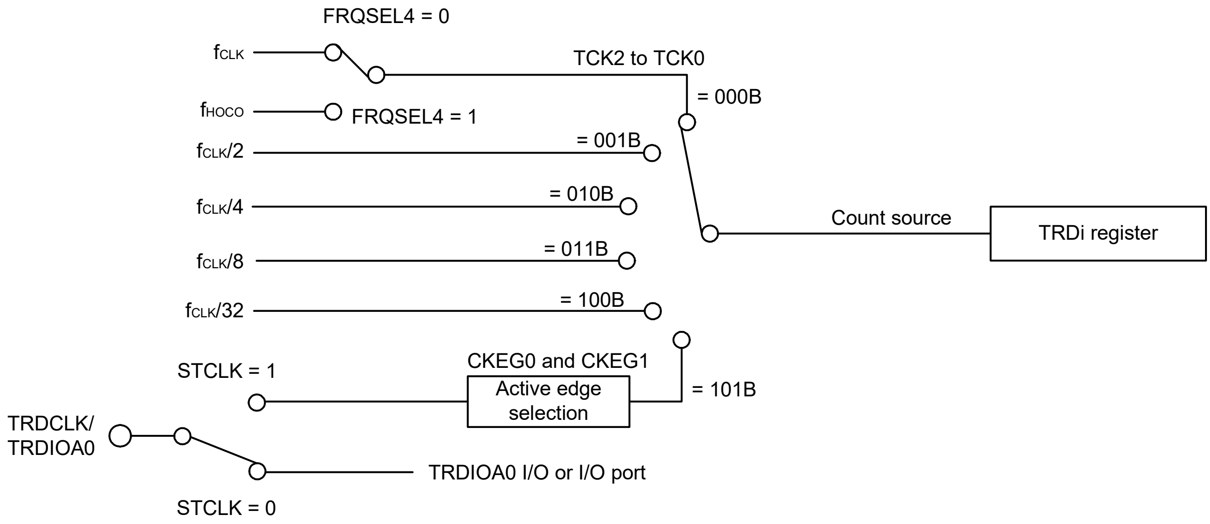
Table 8 - 9 Count Source Selection

Count Source	Selection
f_{CLK}, f_{HOCO} Note, f_{CLK}/2, f_{CLK}/4, f_{CLK}/8, f_{CLK}/32	The count source is selected by bits TCK2 to TCK0 in the TRDCRi register.
External signal input to TRDCLK pin	The STCLK bit in the TRDFCR register is set to 1 (external clock input enabled). Bits TCK2 to TCK0 in the TRDCRi register are set to 101B (count source: external clock). The active edge is selected by bits CKEG1 and CKEG0 in the TRDCRi register. The port mode register bit for the I/O port multiplexed with the TRDCLK pin is set to 1 (input mode).

Remark i = 0 or 1

Note f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Figure 8 - 42 Count Source Block Diagram



Remark
i = 0 or 1

TCK0 to TCK2, CKEG0, CKEG1: Bits in TRDCRi register
STCLK: Bit in TRDFCR register
FRQSEL4: Bit in user option byte (000C2H)

Set the pulse width of the external clock applied to the TRDCLK pin to three or more cycles of the timer RD operating clock (f_{CLK}).

8.4.2 Buffer Operation

The TRDGRCi register (i = 0 or 1) can be used as the buffer register for the TRDGRAi register, and the TRDGRDi register can be used as the buffer register for the TRDGRBi register by means of bits TRDBFCi and TRDBFDi in the TRDMR register.

- TRDGRAi buffer register:TRDGRCi register
- TRDGRBi buffer register:TRDGRDi register

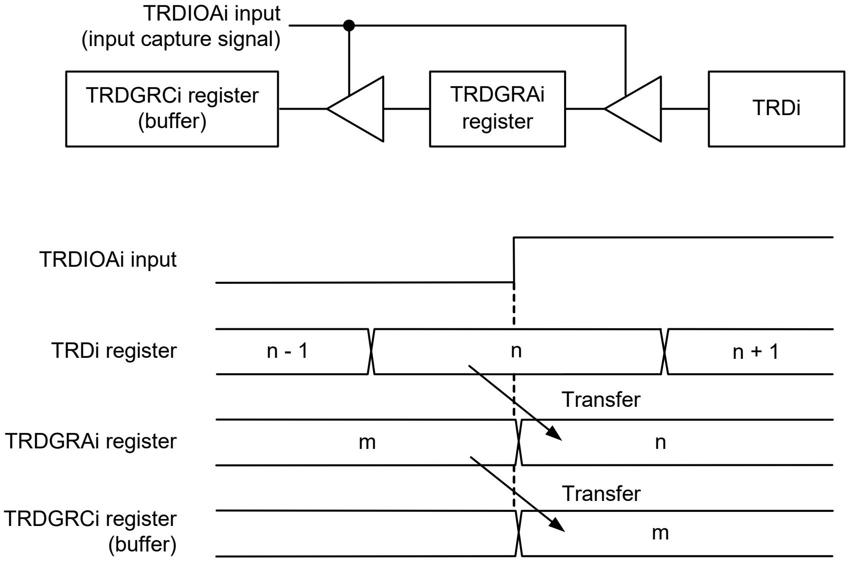
Buffer operation depends on the mode. Table 8 - 10 lists the Buffer Operation in Each Mode.

Table 8 - 10 Buffer Operation in Each Mode

Function and Mode		Transfer Timing	Transfer Register
Timer mode	Input capture function	TRDIOAi input signal (Input capture signal input)	Transfer content of TRDGRAi register to TRDGRCi register (buffer register)
		TRDIOBi input signal (Input capture signal input)	Transfer content of TRDGRBi register to TRDGRDi register (buffer register)
	Output compare function	Compare match with TRDi register and TRDGRAi register	Transfer content of TRDGRCi register (buffer register) to TRDGRAi register
		Compare match with TRDi register and TRDGRBi register	Transfer content of TRDGRDi register (buffer register) to TRDGRBi register
	PWM function	Compare match with TRDi register and TRDGRAi register	Transfer content of TRDGRCi register (buffer register) to TRDGRAi register
		Compare match with TRDi register and TRDGRBi register	Transfer content of TRDGRDi register (buffer register) to TRDGRBi register
Reset synchronous PWM mode		Compare match with TRD0 register and TRDGRA0 register	Transfer content of TRDGRCi register (buffer register) to TRDGRAi register Transfer content of TRDGRDi register (buffer register) to TRDGRBi register
Complementary PWM mode		• Underflow of TRD1 register when CMD1 and CMD0 bits in TRDFCR register are 11B • Compare match with TRD0 register and TRDGRA0 register when CMD1 and CMD0 bits in TRDFCR register are 10B	Transfer content of TRDGRC1 register (buffer register) to TRDGRA1 register Transfer content of TRDGRDi register (buffer register) to TRDGRBi register
PWM3 mode		Compare match with TRD0 register and TRDGRA0 register	Transfer content of TRDGRCi register (buffer register) to TRDGRAi register Transfer content of TRDGRDi register (buffer register) to TRDGRBi register

Remark i = 0 or 1

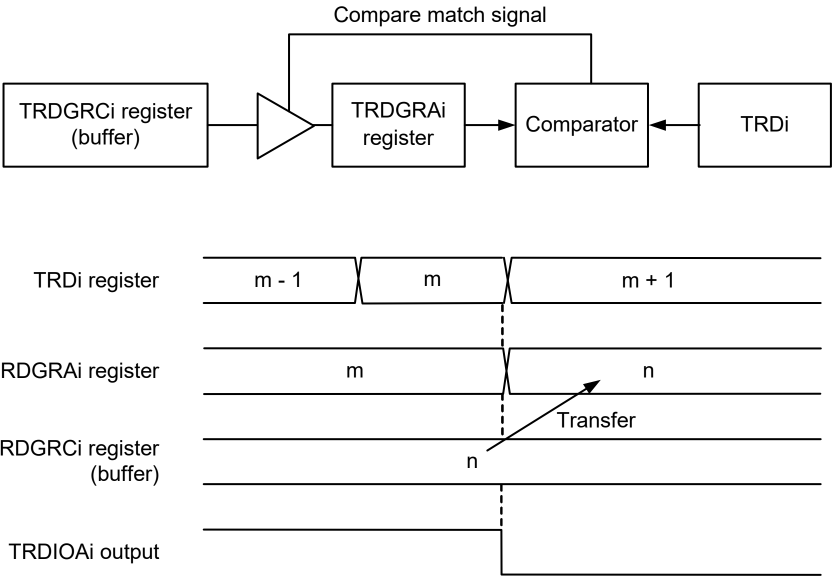
Figure 8 - 43 Buffer Operation in Input Capture Function



Remark
i = 0 or 1

The above diagram applies under the following conditions:
- The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
- Bits IOA2 to IOA0 in the TRDIORAi register are set to 100B (input capture at the rising edge).

Figure 8 - 44 Buffer Operation in Output Compare Function



Remark
i = 0 or 1

The above diagram applies under the following conditions:
- The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
- Bits IOA2 to IOA0 in the TRDIORAi register are set to 001B (low output by compare match).

Perform the following for the timer mode (input capture and output compare functions).
When using the TRDGRCi (i = 0 or 1) register as the buffer register for the TRDGRAi register
- Set the IOC3 bit in the TRDIORCi register to 1 (general register or buffer register).
- Set the IOC2 bit in the TRDIORCi register to the same value as the IOA2 bit in the TRDIORAi register.

When using the TRDGRDi register as the buffer register for the TRDGRBi register
- Set the IOD3 bit in the TRDIORCi register to 1 (general register or buffer register).
- Set the IOD2 bit in the TRDIORCi register to the same value as the IOB2 bit in the TRDIORAi register.

In the input capture function, when the TRDGRCi register or TRDGRDi register is used as a buffer register, the IMFC bit or IMFD bit in the TRDSRi register is set to 1 at the input edge of the TRDIOCi pin or TRDIODi pin.

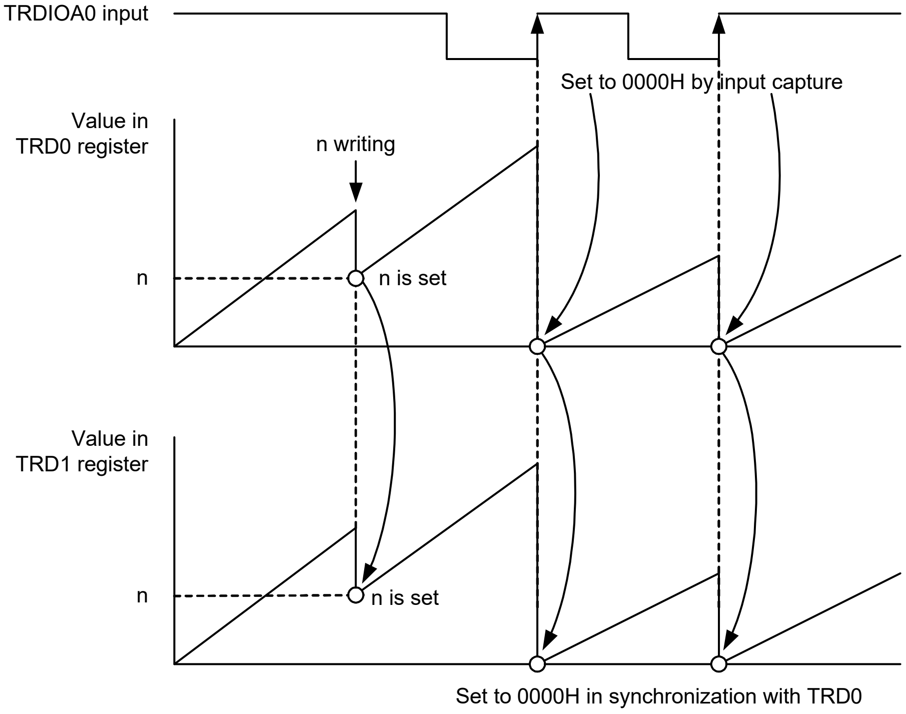
When also using registers TRDGRCi and TRDGRDi as buffer registers for the output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, bits IMFC and IMFD in the TRDSRi register are set to 1 by a compare match with the TRDi register.

8.4.3 Synchronous Operation

The TRD1 register is synchronized with the TRD0 register

- Synchronous preset
  When the TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation), the data is written to both the TRD0 and TRD1 registers after writing to the TRDi register.
- Synchronous clear
  When the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 in the TRDCR0 register are 011B (synchronous clear), the TRD0 register is set to 0000H at the same time as the TRD1 register is set to 0000H.
  Also, when the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 are 011B (synchronous clear), the TRD1 register is set to 0000H at the same time as the TRD0 register is set to 0000H.

Figure 8 - 45 Synchronous Operation



The above diagram applies under the following conditions:

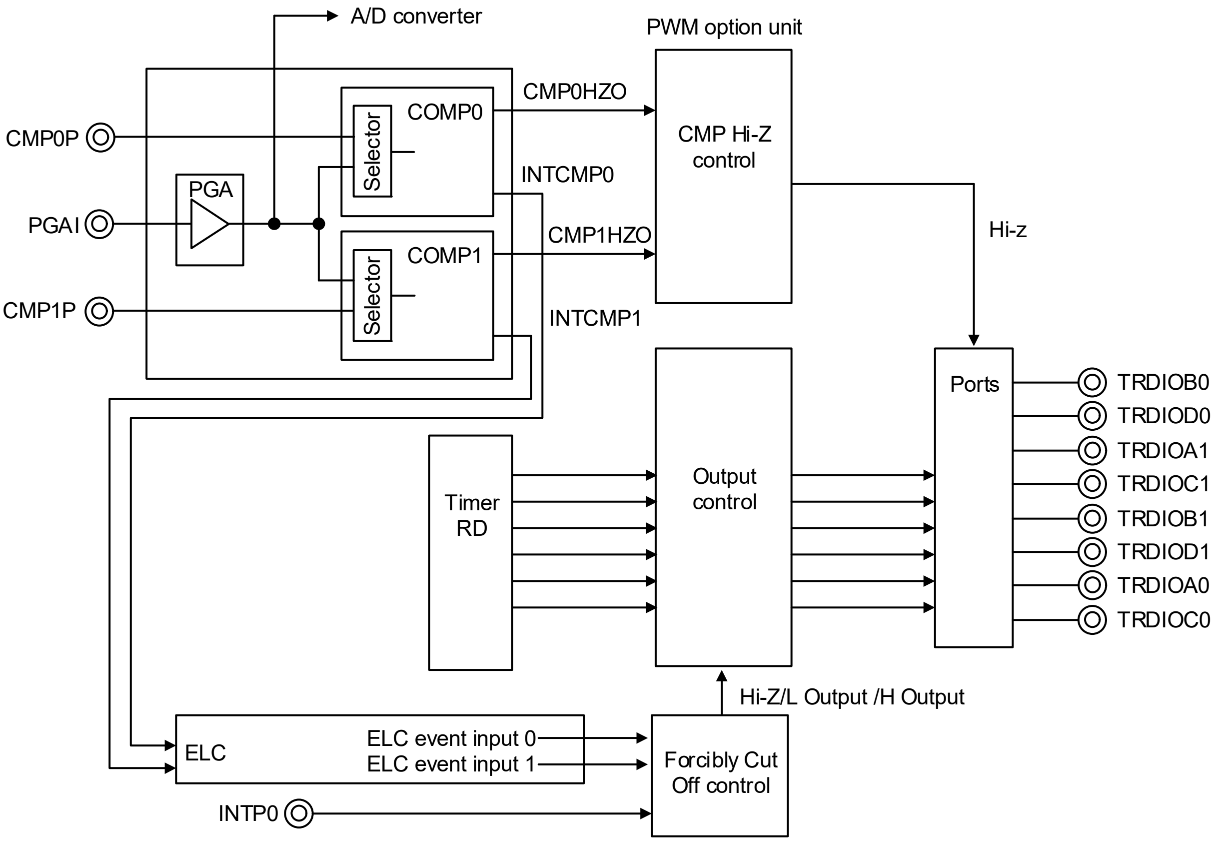
- The TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation).
- Bits CCLR2 to CCLR0 in the TRDCR0 register are set to 001B (TRD0 is set to 0000H by input capture).
  Bits CCLR2 to CCLR0 in the TRDCR1 register are set to 011B (TRD1 is set to 0000H in synchronization with TRD0).
- Bits IOA2 to IOA0 in the TRDIORA0 register are set to 100B.
- Bits CMD1 to CMD0 in the TRDFCR register are set to 00B.
  The PWM 3 bit in the TRDFCR register is set to 1.

} (Input capture at the rising edge of TRDIOA0 input)

8.4.4 Pulse Output Forced Cutoff

In the PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, the pulse output can be cut off by the INTP0 pin input.

Figure 8 - 46 Block Diagram of Pulse Output Forced Cutoff



Timer RD has three kinds of pulse output forced cutoff functions.

(1) Forced cutoff of timer RD by the INTP0 interrupt signal

Output on the output pin of timer RD can be fixed to the high-Z, high output, or low output state.

Restart after forced cutoff must be handled by software.

The pins used for output in these functions or modes can function as the output pin of timer RD when the corresponding bit in the TRDOER1 register is set to 0 (timer RD output enabled). When the TRDPTO bit in the TRDOER2 register is 1 (pulse output forced cutoff signal input INTP0 enabled), the output pin used as a timer RD output port outputs the output value set by the DFCK1, DFCK0, PENB1, PENB0, DFD, DFC, DFB, or DFA bit in the TRDDF0 or TRDDF1 register.

Make the following settings to use this function:

- Set the pin state when the pulse output is forcibly cut off (high impedance, low output, or high output) using TRDDFi.
- When pulse output is forcibly cut out, the TRDSHUTS bit in the TRDOER2 register is set to 1. To suspend the forced cutoff of the pulse output, set the TRDSHUTS bit to 0 while the count is stopped (TSTARTi = 0).
- Set the TRDPTO bit in the TRDOER2 register to 1 (pulse output forced cutoff signal input INTP0 enabled).

(2) Forced cutoff of timer RD by connecting the output signal of the comparator to the PWM option unit
Output on the output pin of timer RD can be placed in the high-Z state.
Restart after forced cutoff must be handled by the software.
The pins used for output in these functions or modes can function as the output pin of timer RD when the corresponding bit in the TRDOER1 register is set to 0 (timer RD output enabled).
When the setting of the CMD[1:0] bits in the TRDFCR register is for reset synchronous PWM mode or complementary PWM mode while the overcurrent detection mode is selected by the HDM bit in the OPMR register and the OPHS0 bit in the OPHS register is set to 1 (selecting the hi-Z state for the pins), the output pins used as the output port from timer RD are placed in the hi-Z state.

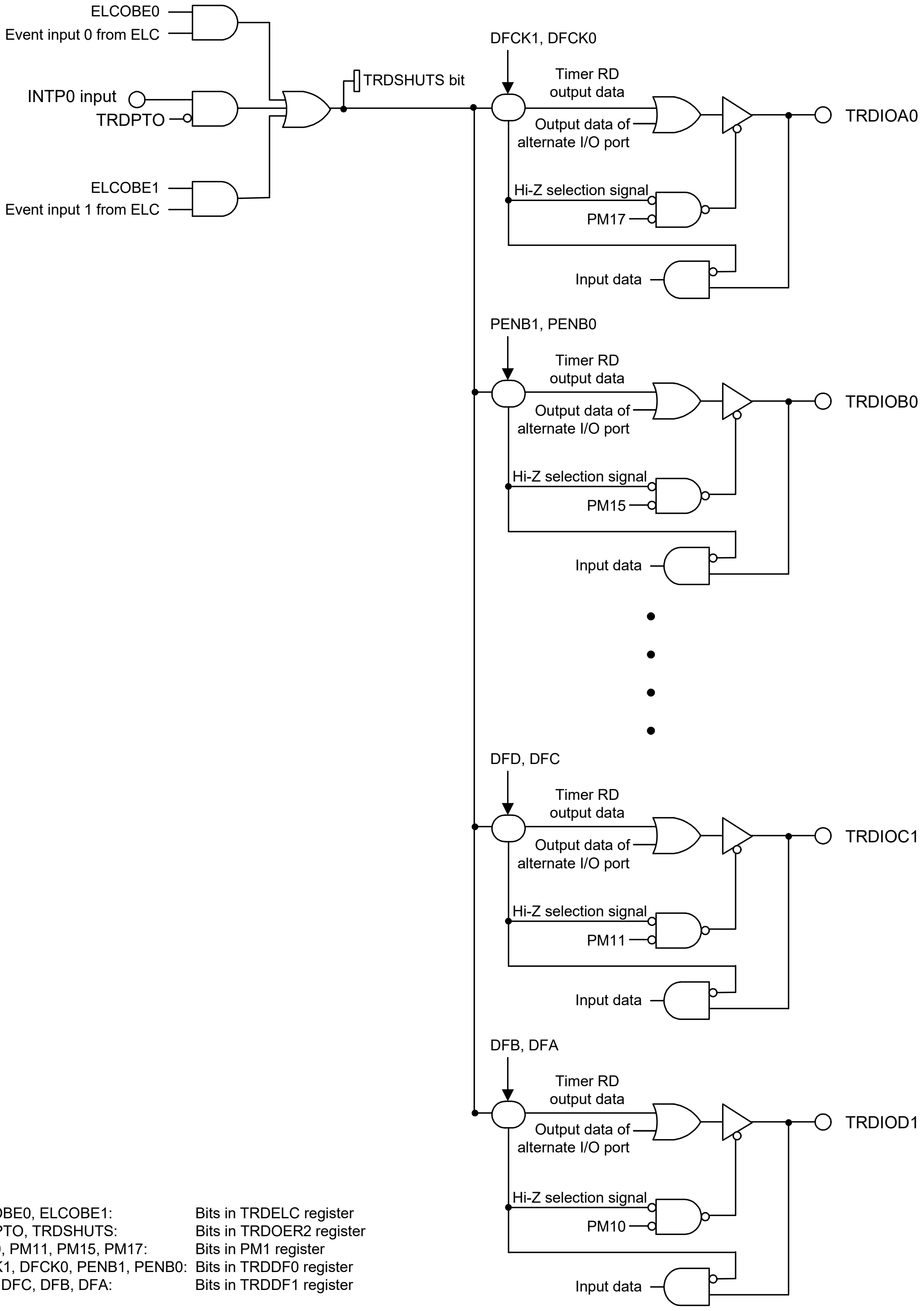
Make the following settings to use this function:
• For forced cutoff of the timer RD by connecting the output signal of the comparator to the PWM option unit, see **13.4.4 Setting procedure**.

(3) Forced cutoff by event input from the event link controller (ELC)
Output on the output pin of timer RD can be placed in the high-Z, high output, or low output state.
The pins used for output in these functions or modes can function as the output pin of timer RD when the corresponding bit in the TRDOER1 register is set to 0 (timer RD output enabled).
When the ELCICE0 or ELCICE1 bit in the TRDELC register is 1 (respectively selecting event input 0 or 1 from the event link controller, ELC) while the pulse output mode (PWM function, reset synchronous PWM mode, complementary PWM mode, or PWM3 mode) is selected and the ELCOBE0 or ELCOBE1 bit in the TRDELC register is 1 (enabling forced cutoff), the output pins used as the output port of timer RD output signals in accord with the settings of the DFCK1, DFCK0, PENB1, PENB0, DFD, DFC, DFB, and DFA bits in the TRDDF0 or TRDDF1 register.

Make the following settings to use this function:
• Set the pin state when the pulse output is forcibly cut off (high impedance, low output, or high output) using TRDDFi.
• Refer to **8.4.5 Event Input from Event Link Controller (ELC)** for details on pulse forced cutoff by ELC event input.
• When pulse output is forcibly cut out, the TRDSHUTS bit in the TRDOER2 register is set to 1. To suspend the forced cutoff of the pulse output, set the TRDSHUTS bit to 0 while the count is stopped (TSTARTi = 0).

The comparator has two kinds of overcurrent detection functions.
(1) Two-stage overcurrent detection function
(2) Overcurrent/induced current detection function
For the operation of the comparator after forced-cutoff, see **13.4.1 Two-stage overcurrent detection function**, **13.4.2 Overcurrent/induced current detection function**, and **13.4.3 Operation example of overcurrent detection function**.

Figure 8 - 47 Pulse Output Forced Cutoff



ELCOBE0, ELCOBE1:	Bits in TRDELC register
TRDPTO, TRDSHUTS:	Bits in TRDOER2 register
PM10, PM11, PM15, PM17:	Bits in PM1 register
DFCK1, DFCK0, PENB1, PENB0:	Bits in TRDDF0 register
DFD, DFC, DFB, DFA:	Bits in TRDDF1 register

8.4.5 Event Input from Event Link Controller (ELC)

Timer RD performs two operations by event input from the ELC.

(a) TRDIOD0/TRDIOD1 input capture

Timer RD captures the TRDIOD0/TRDIOD1 input when an event is input from the ELC. The IMFD bit in the TRDSRi register is set to 1 at this time. To use this function, select the input capture function in timer mode and set the ELCICE0 or ELCICE1 bit in the TRDELC register to 1. This function is disabled in any other modes (for the output compare function in timer mode, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode).

(b) Pulse output forced cutoff operation Note

The pulse output is forcibly cutoff by event input from the ELC. To use this function, select pulse output mode (PWM function, reset synchronous PWM mode, complementary PWM mode, or PWM3 mode) and set the ELCOBE0 or ELCOBE1 bit to 1. This function is disabled for the input capture function in timer mode.

Note The pulse output is cutoff during the low input period for forced cutoff from the INTP0 pin, but the pulse output is cutoff once by a single event input from the ELC for forced cutoff by the ELC event.

[Setting Procedure]

(1) Set timer RD as the ELC event link destination.

(2) Set bits ELCICEi (i = 0 or 1) and ELCOBEi (i = 0 or 1) to 1 in the TRDELC register.

8.4.6 Event Output to Event Link Controller (ELC)

Table 8 - 11 lists the Timer RD Modes and Event Output to ELC.

Table 8 - 11 Timer RD Modes and Event Output to ELC

Used Mode	Output Source	ELC
Input capture function	TRDIOA0 edge detection set by bits IOA1 and IOA0 in the TRDIORA0 register	Available
	TRDIOB0 edge detection set by bits IOB1 and IOB0 in the TRDIORA0 register	Available
	TRDIOC0 edge detection set by bits IOC1 and IOC0 in the TRDIORC0 register	—
	TRDIOD0 edge detection set by bits IOD1 and IOD0 in the TRDIORC0 register	—
	TRDIOA1 edge detection set by bits IOA1 and IOA0 in the TRDIORA1 register	Available
	TRDIOB1 edge detection set by bits IOB1 and IOB0 in the TRDIORA1 register	Available
	TRDIOC1 edge detection set by bits IOC1 and IOC0 in the TRDIORC1 register	—
	TRDIOD1 edge detection set by bits IOD1 and IOD0 in the TRDIORC1 register	—
Output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode	Compare match between registers TRD0 and TRDGRA0	Available
	Compare match between registers TRD0 and TRDGRB0	Available
	Compare match between registers TRD0 and TRDGRC0	—
	Compare match between registers TRD0 and TRDGRD0	—
	Compare match between registers TRD1 and TRDGRA1	Available
	Compare match between registers TRD1 and TRDGRB1	Available
	Compare match between registers TRD1 and TRDGRC1	—
	Compare match between registers TRD1 and TRDGRD1	—
Complementary PWM mode	TRD1 register underflow	Available

8.5 Timer RD Operation

8.5.1 Input capture function

The input capture function measures the external signal width and period. The content of the TRDi register (counter) is transferred to the TRDGRji register as a trigger of the TRDIOji pin (i = 0 or 1, j = A, B, C, or D) external signal (input capture). Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, the input capture function, or any other mode or function, can be selected for each individual pin.

Figure 8 - 48 shows the Block Diagram of Input Capture Function, Table 8 - 12 lists the Input Capture Function Specifications, and Figure 8 - 49 shows an Operation Example of Input Capture Function.

Figure 8 - 48 Block Diagram of Input Capture Function

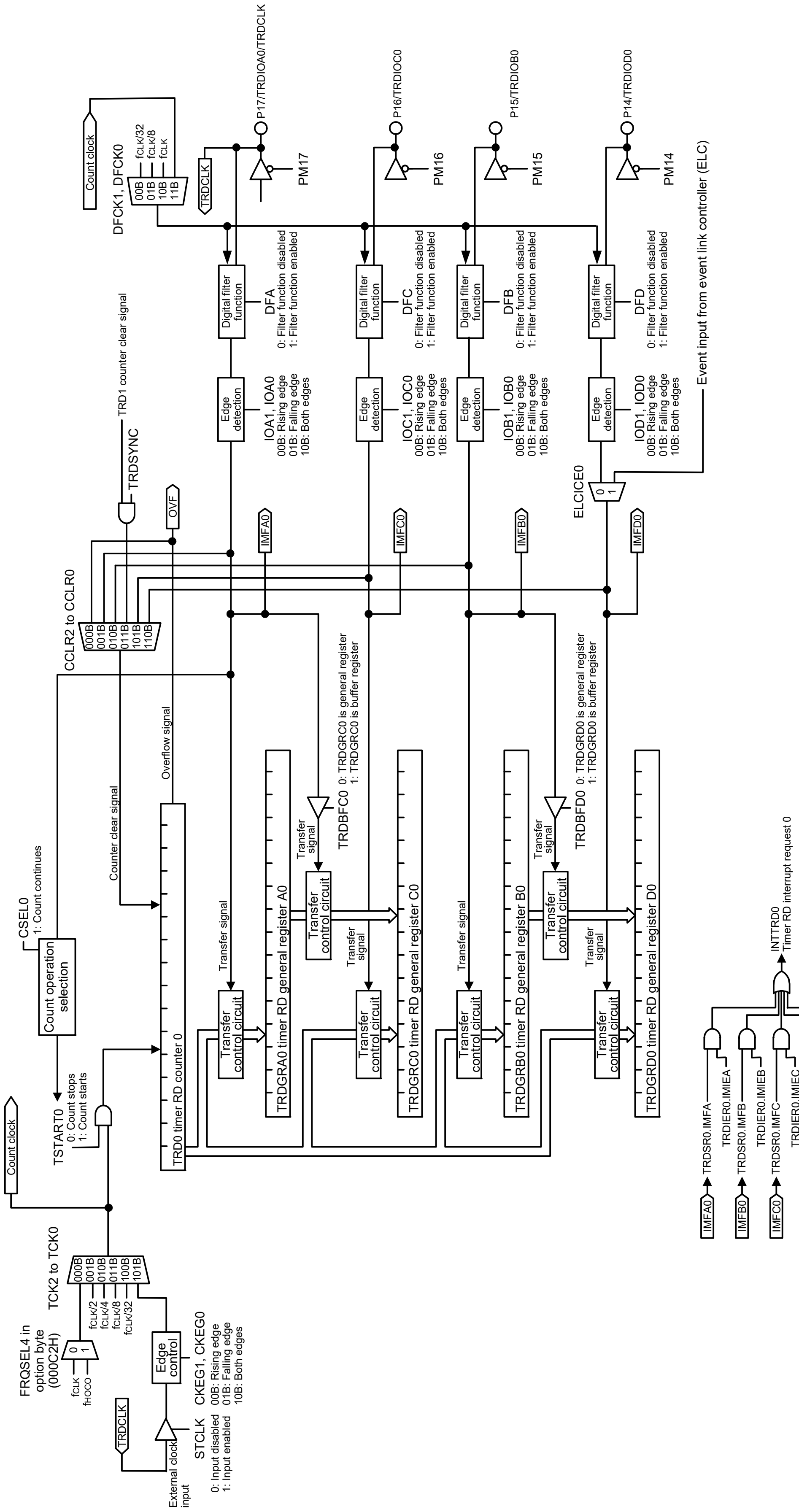


Table 8 - 12 Input Capture Function Specifications

Item	Specification
Count sources	f_{HOCO} Note, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRDCLK pin (active edge selected by a program)
Count operations	Increment
Count period	When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000B (free-running operation). 1/fk × 65536 fk: Frequency of count source
Count start condition	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop condition	0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1.
Interrupt request generation timing	• Input capture (active edge of TRDIOji input) • TRDi register overflow
TRDIOA0 pin function	I/O port, input-capture input, or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin function	I/O port or input-capture input (selectable for each pin)
INTP0 pin function	Not used (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi register.
Write to timer	• When the TRDSYNC bit in the TRDMR register is 0 (timer RD0 and timer RD1 operate independently). Data can be written to the TRDi register. • When the TRDSYNC bit in the TRDMR register is 1 (timer RD0 and timer RD1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register.
Selectable functions	• Input-capture input pin selection Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi. • Input-capture input active edge selection Rising edge, falling edge, or both rising and falling edges • Timing for setting the TRDi register to 0000H. At overflow or input capture • Buffer operation (see **8.4.2 Buffer Operation**) • Synchronous operation (see **8.4.3 Synchronous Operation**) • Digital filter. The TRDIOji input is sampled, and when the sampled input level match three times, that level is determined. • Input capture operation by event input from event link controller (ELC).

Note f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

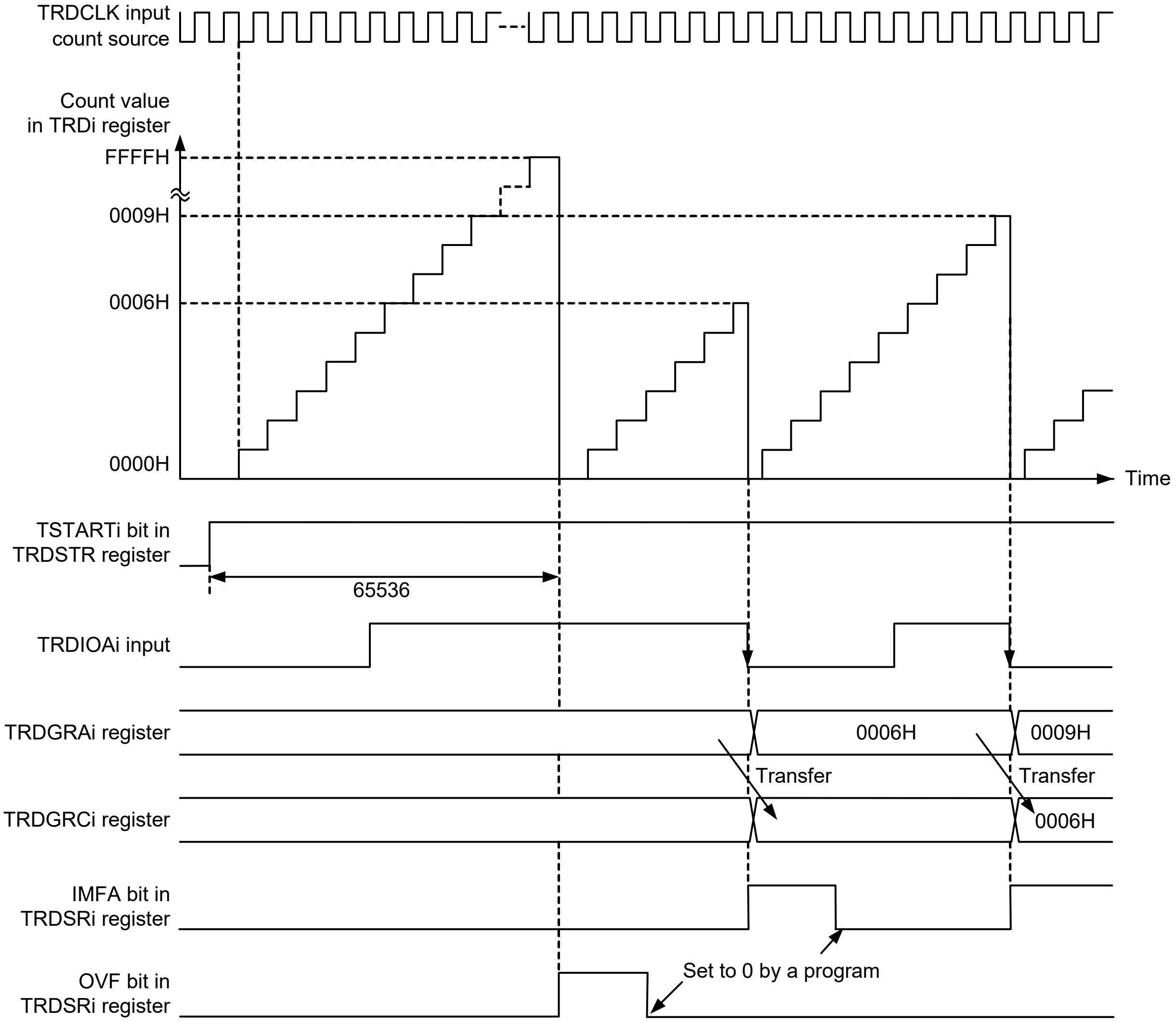
Remark i = 0 or 1, j = A, B, C, or D

(1) Operation Example

By setting bits CCLR0 to CCLR2 in the TRDCRi register (i = 0 or 1), the timer RDi counter value is reset by an input capture/compare match. Figure 8 - 49 shows an operation example with bits CCLR2 to CCLR0 set to 001B.

If the input capture operation has been set to clear the count during operation and is performed when the timer count value is FFFFH, depending on the timing between the count source and input capture operation interrupt flags bits IMFA to IMFD and OVF in the TRDSRi register may be set to 1 simultaneously.

Figure 8 - 49 Operation Example of Input Capture Function



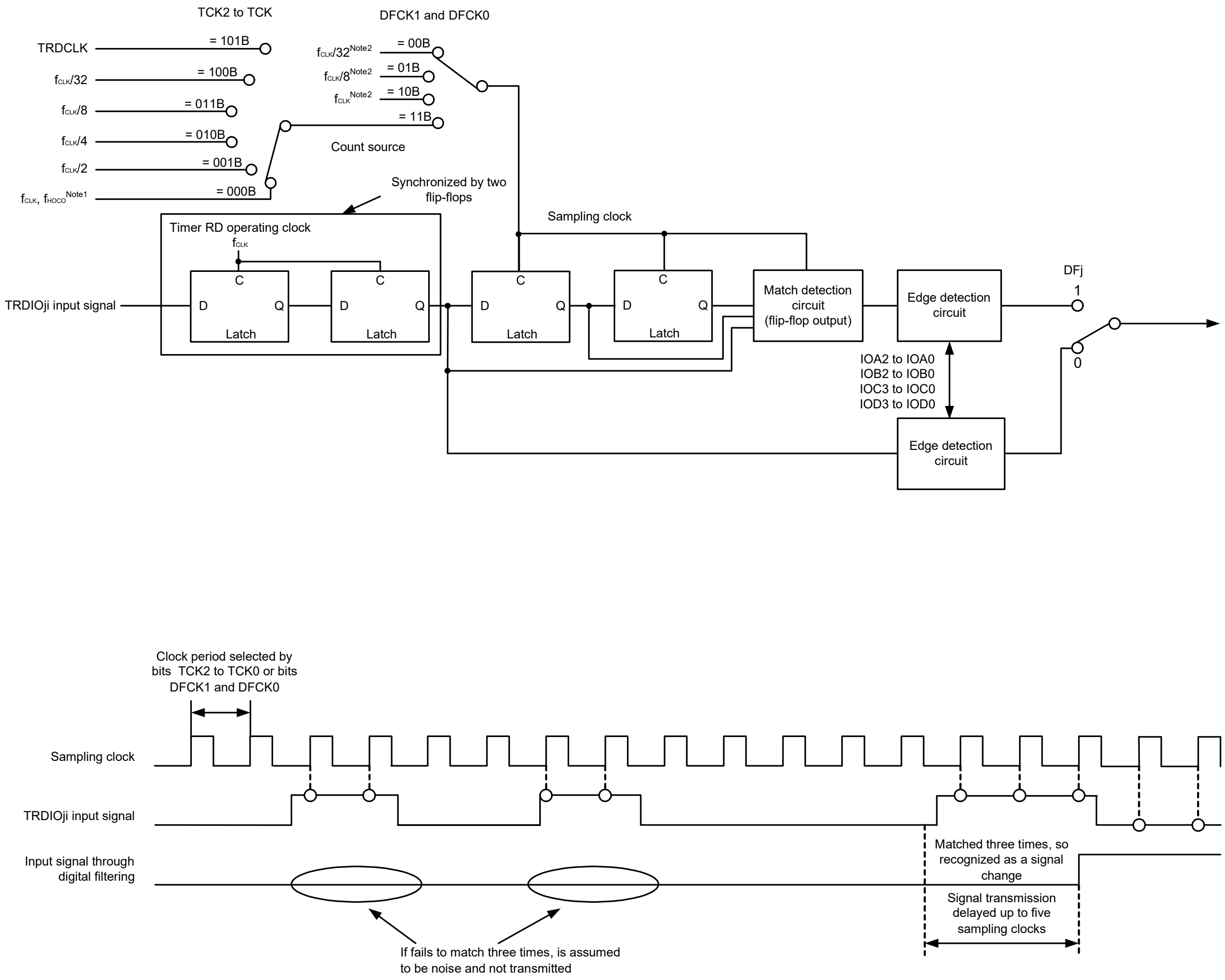
Remark
i = 0 or 1

The above diagram applies under the following conditions:
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi register is set to 0000H by TRDGRAi register input capture).
Bits TCK2 to TCK0 in the TRDCRi register are set to 101B (TRDCLK input for the count source).
Bits CKEG1 and CKEG0 in the TRDCRi register are set to 01B (count at the falling edge for the count source).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 101B (input capture at the falling edge of TRDIOAi input).
The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).

(2) Digital Filter

The TRDIOji input (i = 0 or 1, j = A, B, C, or D) is sampled, and when the sampled input level matches three times, its level is determined. Select the digital filter function and sampling clock using the TRDDFi register. Figure 8 - 50 shows the Block Diagram of Digital Filter.

Figure 8 - 50 Block Diagram of Digital Filter



Remark
i = 0 or 1, j = A, B, C, or D

TCK0 to TCK2: Bits in TRDCRi register
DFCK0, DFCK1, DFj: Bits in TRDDF register
IOA0 to IOA2, IOB0 to IOB2: Bits in TRDIORAi register
IOC0 to IOC3, IOD0 to IOD3: Bits in TRDIORCi register

Note 1. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H).

Note 2. When FRQSEL4 = 1 in the user option byte (000C2H), $f_{CLK}/32$, $f_{CLK}/8$, and f_{CLK} are set to $f_{HOCO}/32$, $f_{HOCO}/8$, and f_{HOCO}, respectively.

8.5.2 Output compare function

This function detects matches (compare match) between the content of the TRDGRji register (j = A, B, C, or D) and the content of the TRDi register (counter) (i = 0 or 1). When the contents match, an arbitrary level is output from the TRDIOji pin. Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, the output compare function, or any other mode or function, can be selected for each individual pin.
Figure 8 - 51 shows the Block Diagram of Output Compare Function, Table 8 - 13 lists the Output Compare Function Specifications, and Figure 8 - 52 shows an Operation Example of Output Compare Function.

Figure 8 - 51 Block Diagram of Output Compare Function

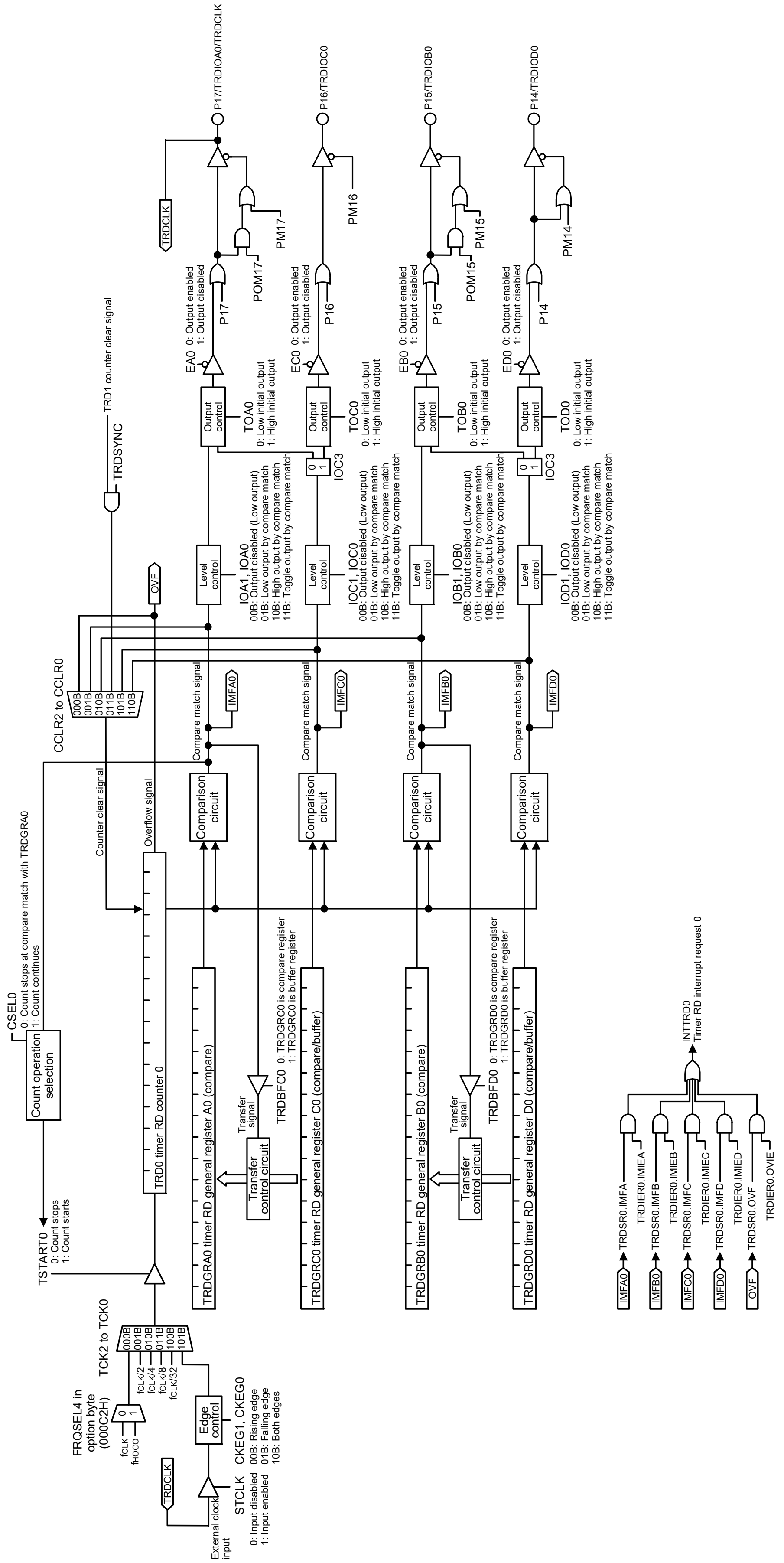


Table 8 - 13 Output Compare Function Specifications

Item	Specification
Count sources	f_{HOCO} Note, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRDCLK pin (active edge selected by a program)
Count operations	Increment
Count period	• When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000B (free-running operation). 1/fk × 65536 fk: Frequency of count source • When bits CCLR1 and CCLR0 in the TRDCRi register are set to 01B or 10B (TRDi register is set to 0000H at compare match with TRDGRji register). 1/fk × (n + 1) n: Value set in the TRDGRji register
Waveform output timing	Compare match (contents of registers TRDi and TRDGRji match)
Count start condition	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop conditions	• 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. The output compare output pin holds the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register. The output compare output pin holds the level after output change by compare match.
Interrupt request generation timing	• Compare match (contents of registers TRDi and TRDGRji match) • TRDi register overflow
TRDIOA0 pin function	I/O port, output-compare output, or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin function	I/O port or output-compare output (selectable for each pin)
INTP0 pin function	Not used (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi register.
Write to timer	• When the TRDSYNC bit in the TRDMR register is set to 0 (timer RD0 and timer RD1 operate independently). Data can be written to the TRDi register. • When the TRDSYNC bit in the TRDMR register is set to 1 (timer RD0 and timer RD1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register.
Selectable functions	• Output-compare output pin selection Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi. • Output level selection at compare match Low output, high output, or inverted output level • Initial output level selection The level can be set for the period from the count start to the compare match. • Timing for setting the TRDi register to 0000H Overflow or compare match in the TRDGRAi register • Buffer operation (see **8.4.2 Buffer Operation**) • Synchronous operation (see **8.4.3 Synchronous Operation**) • Changing output pins for registers TRDGRCi and TRDGRDi The TRDGRCi register can be used as output control of the TRDIOAi pin and the TRDGRDi register can be used as output control of the TRDIOBi pin. • Timer RD can be used as the internal timer without output.

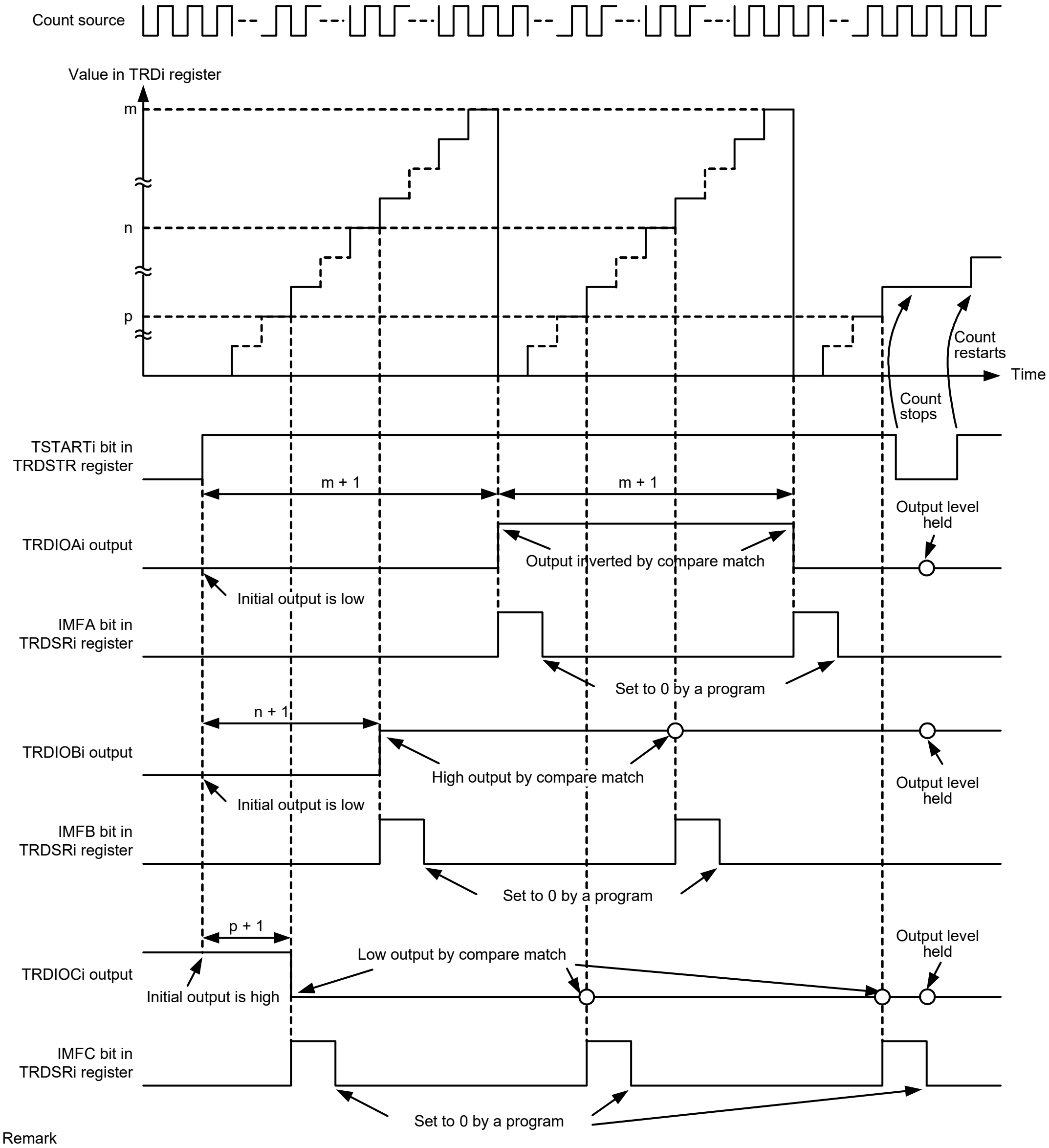
Note f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Remark i = 0 or 1, j = A, B, C, or D

(1) Operation Example

By setting bits CCLR0 to CCLR2 in the TRDCRi register (i = 0 or 1), the timer RDi counter value is reset by an input capture/compare match. If the expected compare value is FFFFH at this time, FFFFH changes to 0000H, same as the overflow operation, and the overflow flag is set to 1.

Figure 8 - 52 Operation Example of Output Compare Function



Remark
i = 0 or 1
M: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register

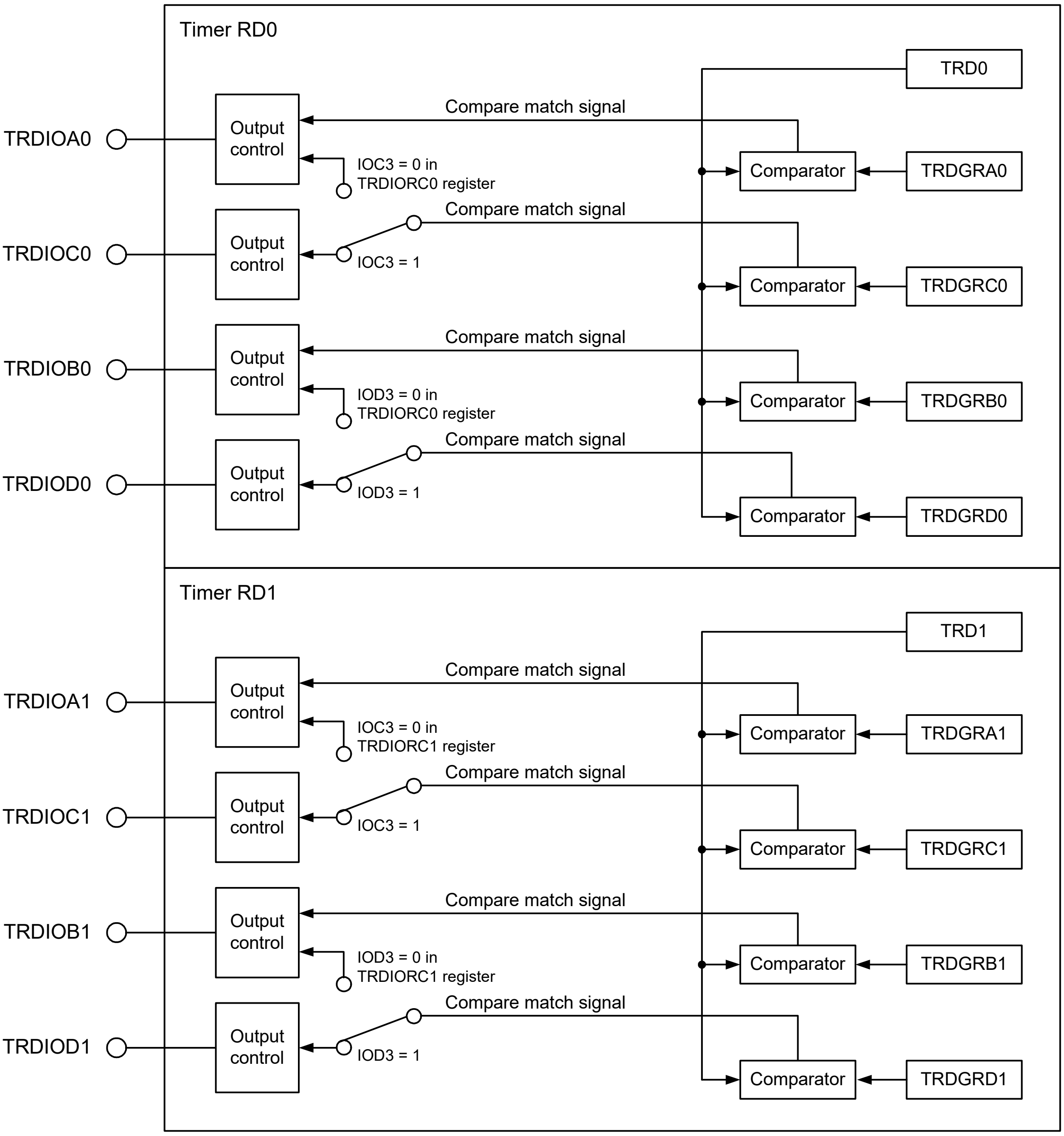
The above diagram applies under the following conditions:
The CSELi bit in the TRDSTR register is set to 1 (TRDi is not stopped by compare match).
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EAi, EBi, and ECi in the TRDOER1 register are set to 0 (TRDIOAi, TRDIOBi and TRDIOCi output enabled).
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi is set to 0000H by compare match with TRDGRAi).
Bits TOAi and TOBi in the TRDOCR register is set to 0 (initial output is low until compare match), the TOCi bit is set to 1 (initial output is high until compare match).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB2 to IOB0 in the TRDIORAi register are set to 010B (TRDIOBi high output at TRDGRBi compare match).
Bits IOC3 to IOC0 in the TRDIORCi register are set to 1001B (TRDIOCi low output at TRDGRCi register compare match).
Bits IOD3 to IOD0 in the TRDIORCi register are set to 1000B (TRDGRDi register does not control TRDIOBi pin output. Pin output by compare match is disabled).

(2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi

The TRDGRCi register can be used for output control of the TRDIOAi pin, and the TRDGRDi register can be used for output control of the TRDIOBi pin. Therefore, each pin output can be controlled as follows:

- TRDIOAi output is controlled by the values in registers TRDGRAi and TRDGRCi.
- TRDIOBi output is controlled by the values in registers TRDGRBi and TRDGRDi.

Figure 8 - 53 Changing Output Pins in Registers TRDGRCi and TRDGRDi

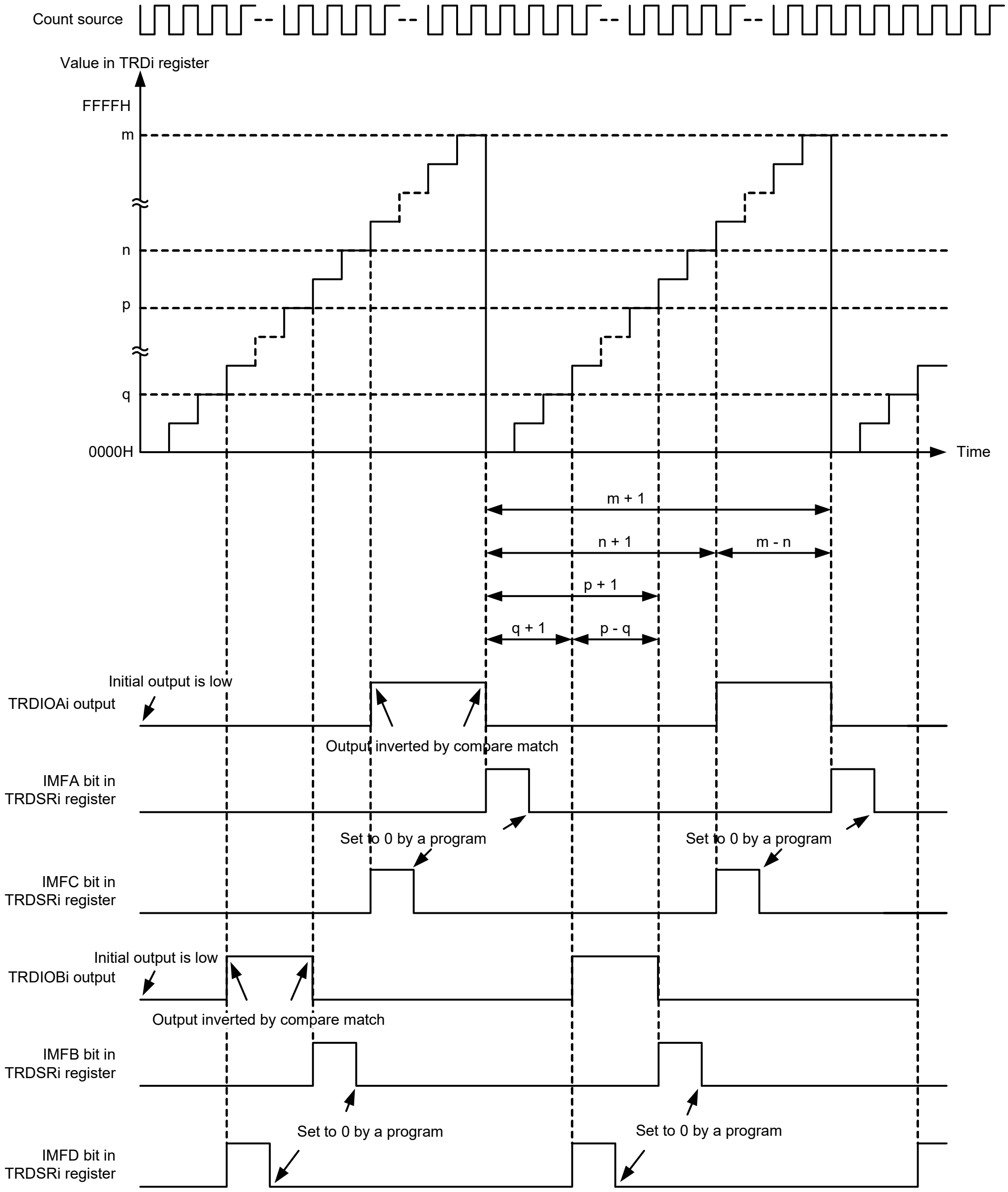


Change output pins in registers TRDGRCi and TRDGRDi as follows:

- Select 0 (TRDGRji register output pin is changed) using the IOj3 (j = C or D) bit in the TRDIORCi register.
- Set the TRDBFji bit in the TRDMR register to 0 (general register).
- Set different values in registers TRDGRCi and TRDGRAi. Also, set different values in registers TRDGRDi and TRDGRBi.

Figure 8 - 54 shows an Operation Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin.

Figure 8 - 54 Operation Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin



Remark
i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRCi register
p: Value set in TRDGRBi register
q: Value set in TRDGRDi register

The above diagram applies under the following conditions:
The CSELi bit in the TRDSTR register is set to 1 (TRDi register is not stopped by compare match).
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EAi and EBi in the TRDOER1 register are set to 0 (TRDIOAi and TRDIOBi output enabled).
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi is set to 0000H by compare match with TRDGRAi).
Bits TOAi and TOBi in the TRDOCR register are set to 0 (initial output is low until compare match).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB2 to IOB0 in the TRDIORAi register are set to 011B (TRDIOBi output inverted at TRDGRBi compare match).
Bits IOC3 to IOC0 in the TRDIORCi register are set to 0011B (TRDIOAi output inverted at TRDGRCi compare match).
Bits IOD3 to IOD0 in the TRDIORCi register are set to 0011B (TRDIOBi output inverted at TRDGRDi compare match).

8.5.3 PWM function

In PWM function, a PWM waveform is output. Up to three PWM waveforms with the same period can be output by timer RDi (i = 0 or 1). Also, up to six PWM waveforms with the same period can be output by synchronizing timer RD0 and timer RD1.

Since this mode functions by a combination of the TRDIOji pin (i = 0 or 1, j = B, C, or D) and TRDGRji register, PWM function, or any other mode or function, can be selected for each individual pin. (However, since the TRDGRAi register is used when using any pin for PWM function, the TRDGRAi register cannot be used for other modes.)

Figure 8 - 55 shows the Block Diagram of PWM Function, Table 8 - 14 lists the PWM Mode Specifications, and Figure 8 - 56 and Figure 8 - 57 show Operation Examples in PWM Function.

Figure 8 - 55 Block Diagram of PWM Function

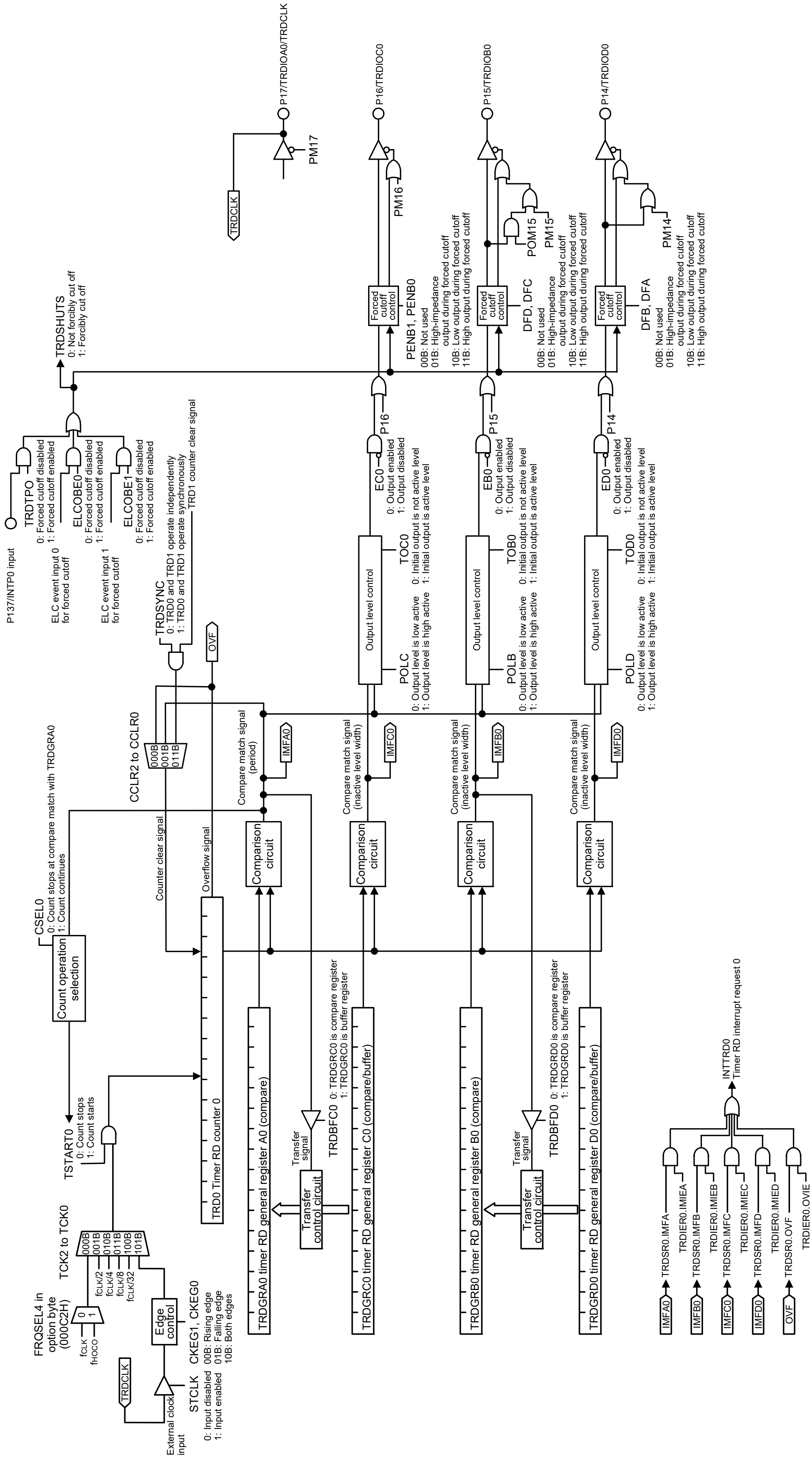
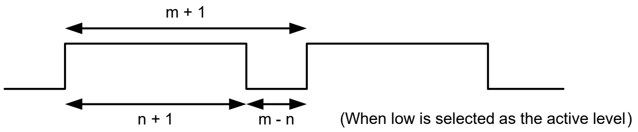


Table 8 - 14 PWM Mode Specifications

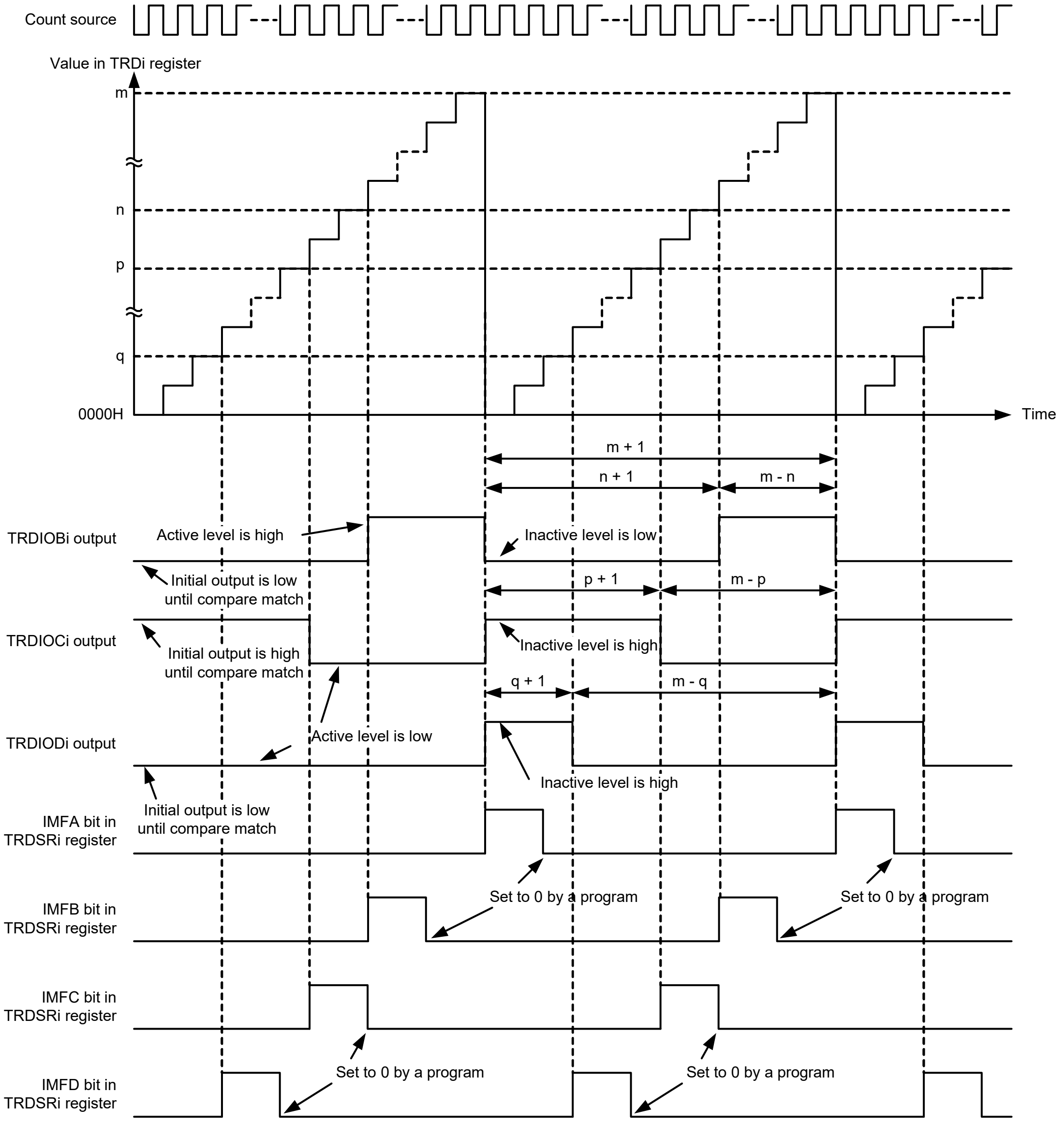
Item	Specification
Count sources	f_{HOCO} Note, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRDCLK pin (active edge selected by a program)
Count operations	Increment
PWM waveform	PWM period: 1/fk × (m + 1) Active level width: 1/fk × (m - n) Inactive level width: 1/fk × (n + 1) fk: Frequency of count source m: Value set in the TRDGRAi register n: Value set in the TRDGRji register 
Count start condition	1 (count starts) is written to the TSTARTi bit in the TRDSTR register.
Count stop conditions	• 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. The PWM output pin holds the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register. The PWM output pin holds the level after output change by compare match.
Interrupt request generation timing	• Compare match (content of the TRDi register matches content of the TRDGRhi register) • TRDi register overflow
TRDIOA0 pin function	I/O port or TRDCLK (external clock) input
TRDIOA1 pin function	I/O port
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOB1, TRDIOC1, TRDIOD1 pin function	I/O port or pulse output (selectable for each pin)
INTP0 pin function	Pulse output forced cutoff signal input (I/O port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi register.
Write to timer	The value can be written to the TRDi register.
Selectable functions	• One to three PWM output pins selectable with timer RDi Either one pin or multiple pins of TRDIOBi, TRDIOCi, and TRDIODi. • Active level selectable for each pin. • Initial output level selectable for each pin. • Synchronous operation (see **8.4.3 Synchronous Operation**) • Buffer operation (see **8.4.2 Buffer Operation**) • Pulse output forced cutoff signal input (see **8.4.4 Pulse Output Forced Cutoff**)

Note f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Remark i = 0 or 1, j = B, C, or D, h = A, B, C, or D

(1) Operation Example

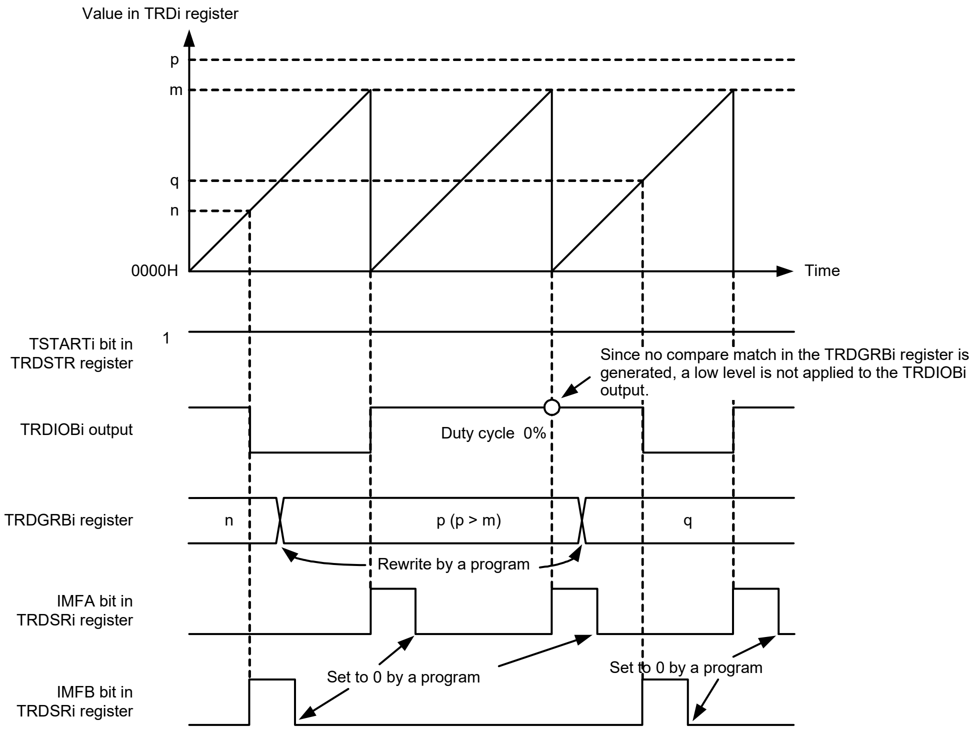
Figure 8 - 56 Operation Example in PWM Function

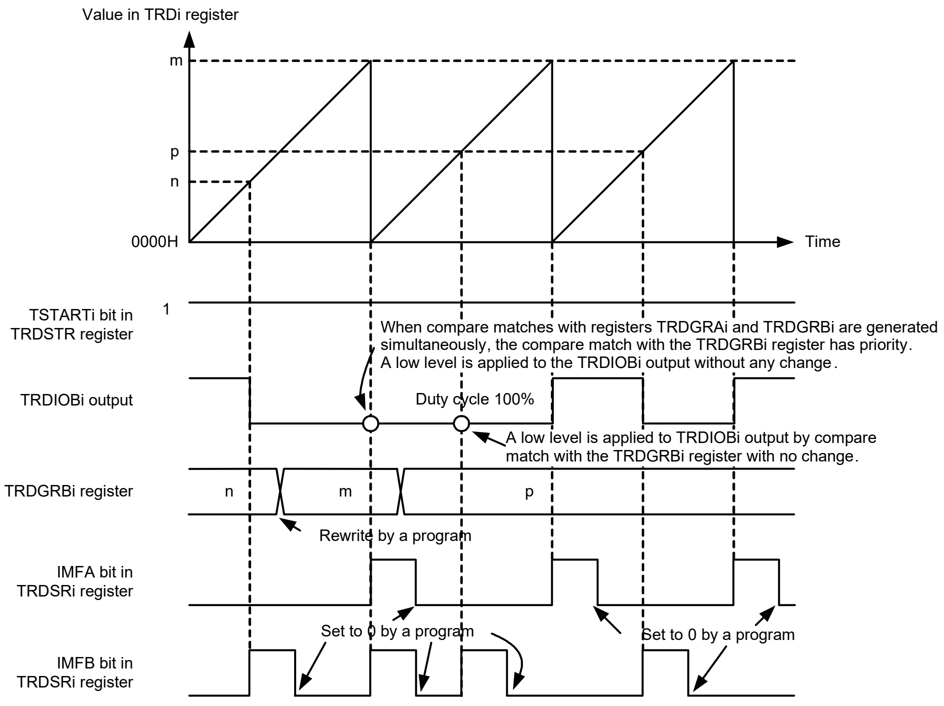


Remark
i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register
q: Value set in TRDGRDi register

The above diagram applies under the following conditions:
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EBi, ECi, and EDi in the TRDOER1 register are set to 0 (TRDIOBi, TRDIOCi and TRDIODi output enabled).
Bits TOBi and TOCi in the TRDOCR register are set to 0 (inactive level), the TODi bit is set to 1 (active level).
The POLB bit in the TRDPOCRi register is set to 1 (active level is high), bits POLC and POLD are set to 0 (active level is low).

Figure 8 - 57 Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)





Remark
i = 0 or 1
m: Value set in TRDGRAi register

The above diagram applies under the following conditions:
The EBi bit in the TRDOER1 register is set to 0 (TRDIOBi output enabled).
The POLB bit in the TRDPOCRi register is set to 0 (active level is low).

8.5.4 Reset synchronous PWM mode

In this mode, three normal-phases and three counter-phases of the PWM waveform are output with the same period (three-phase, sawtooth wave modulation, and no dead time).

Figure 8 - 58 shows the Block Diagram of Reset Synchronous PWM Mode, Table 8 - 15 lists the Reset Synchronous PWM Mode Specifications, Figure 8 - 59 shows an Operation Example in Reset Synchronous PWM Mode.

See **Figure 8 - 57 Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)** for an operation example in PWM Mode with duty cycle 0% and duty cycle 100%.

Figure 8 - 58 Block Diagram of Reset Synchronous PWM Mode

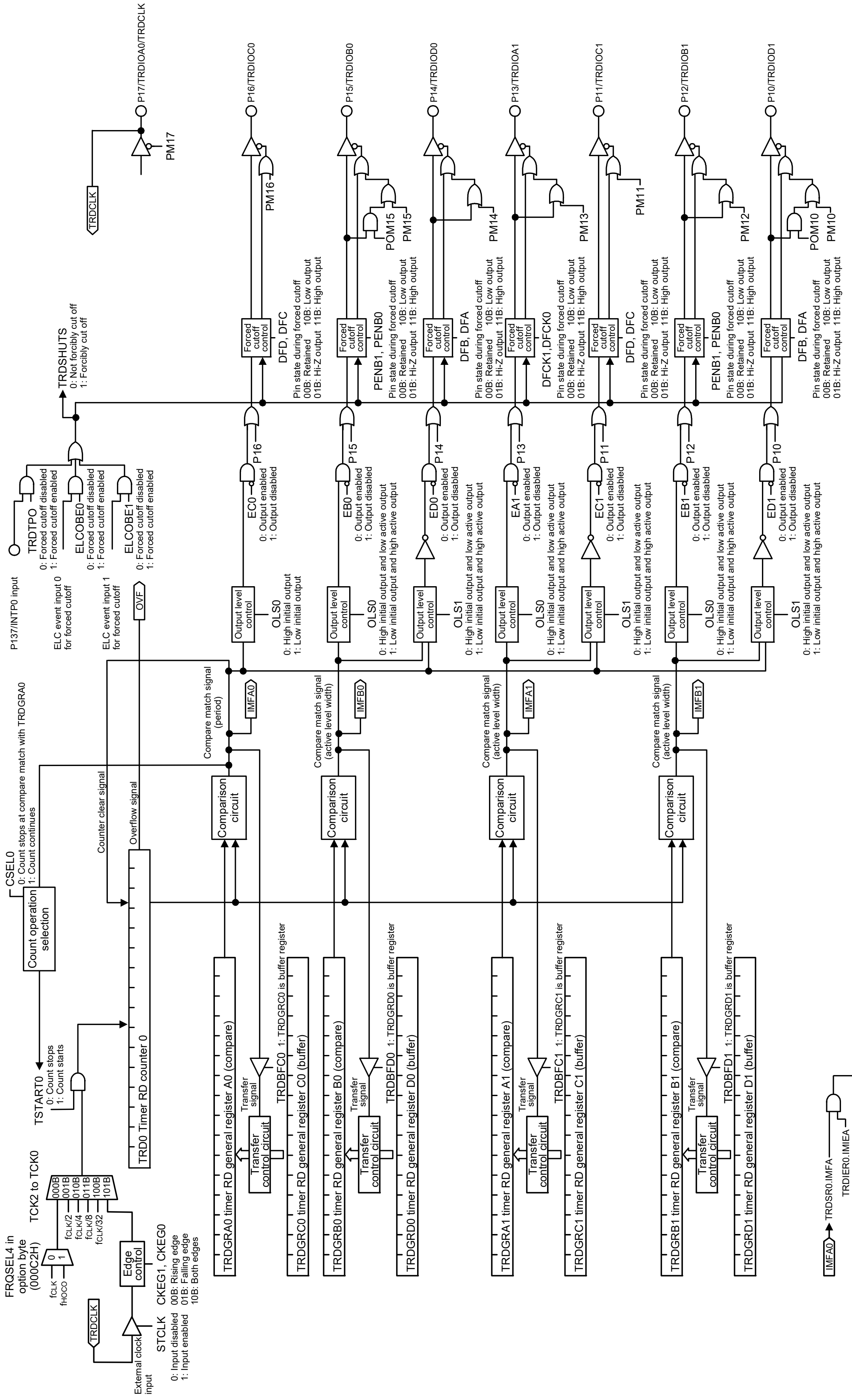
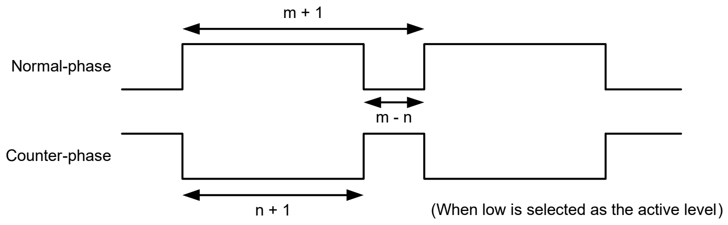


Table 8 - 15 Reset Synchronous PWM Mode Specifications

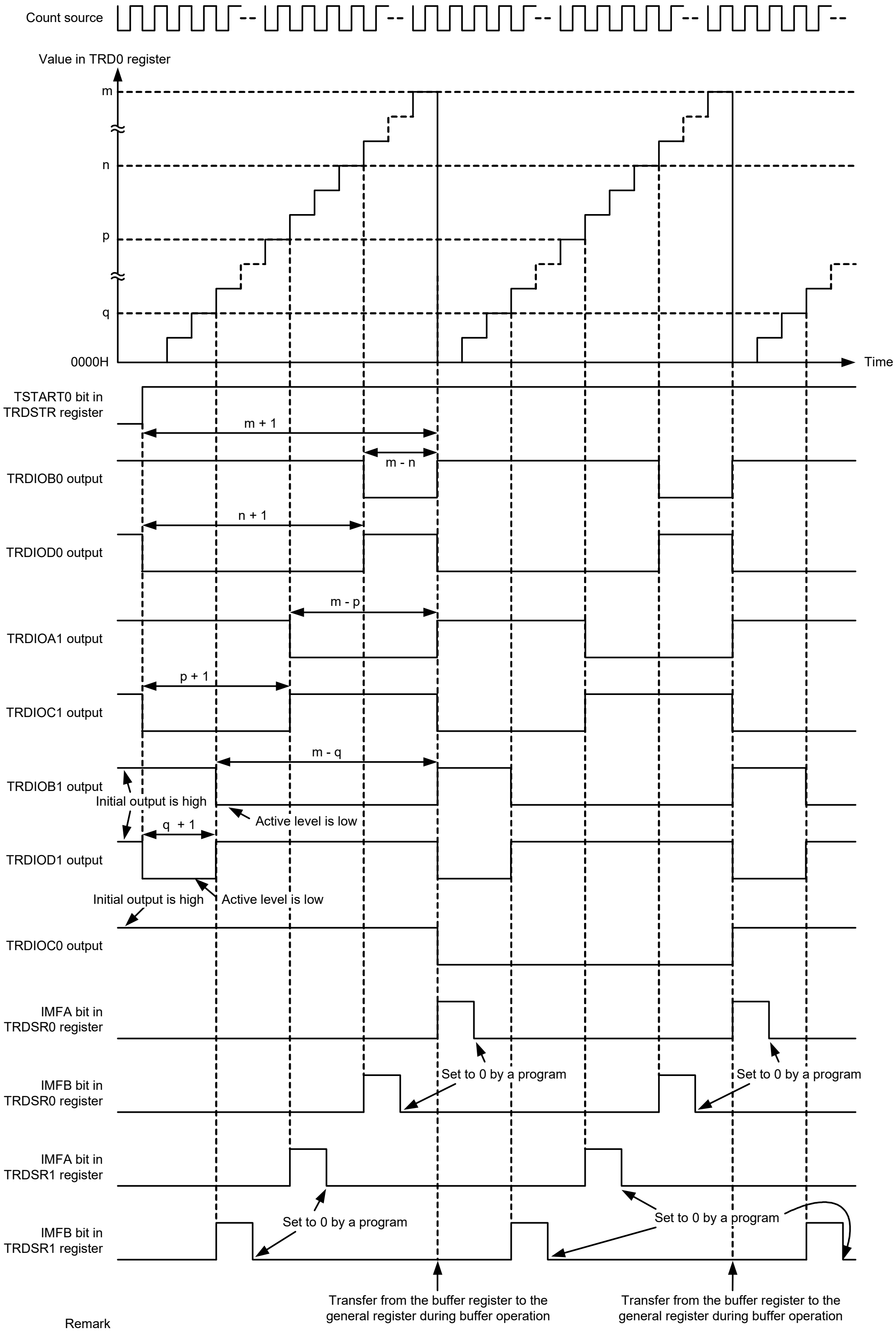
Item	Specification
Count sources	f_{HOCO} [Note], f_{CLK}, $f_{\text{CLK}}/2$, $f_{\text{CLK}}/4$, $f_{\text{CLK}}/8$, $f_{\text{CLK}}/32$ External signal input to the TRDCLK pin (active edge selected by a program)
Count operations	The TRD0 register is incremented (the TRD1 register is not used).
PWM waveform	PWM period: 1/fk × (m + 1) Active level of normal-phase: 1/fk × (m - n) Inactive level of counter-phase: 1/fk × (n + 1) fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) 
Count start condition	1 (count starts) is written to the TSTART0 bit in the TRDSTR register.
Count stop conditions	• 0 (count stops) is written to the TSTART0 bit when the CSEL0 bit in the TRDSTR register is set to 1. The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register. • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRA0 register. The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.
Interrupt request generation timing	• Compare match (content of the TRD0 register matches content of registers TRDGRj0, TRDGRA1, and TRDGRB1) • TRD0 register overflow
TRDIOA0 pin function	I/O port or TRDCLK (external clock) input
TRDIOB0 pin function	PWM1 output normal-phase output
TRDIOD0 pin function	PWM1 output counter-phase output
TRDIOA1 pin function	PWM2 output normal-phase output
TRDIOC1 pin function	PWM2 output counter-phase output
TRDIOB1 pin function	PWM3 output normal-phase output
TRDIOD1 pin function	PWM3 output counter-phase output
TRDIOC0 pin function	Output inverted every PWM period
INTP0 pin function	Pulse output forced cutoff signal input (I/O port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRD0 register.
Write to timer	The value can be written to the TRD0 register.
Selectable functions	• The normal-phase and counter-phase active level and initial output level are selected individually. • Buffer operation (see **8.4.2 Buffer Operation**) • Pulse output forced cutoff signal input (see **8.4.4 Pulse Output Forced Cutoff**)

Note f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Remark j = A, B, C, or D

(1) Operation Example

Figure 8 - 59 Operation Example in Reset Synchronous PWM Mode



Remark
i = 0 or 1
m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
p: Value set in TRDGRA1 register
q: Value set in TRDGRB1 register

The above diagram applies under the following condition:
Bits OLS1 and OLS0 in the TRDFCR register are set to 0 (initial output level is high, active level is low).

8.5.5 Complementary PWM mode

In this mode, three normal-phases and three counter-phases of the PWM waveform are output with the same period (three-phase, triangular wave modulation, and with dead time).

Figure 8 - 60 shows the Block Diagram of Complementary PWM Mode, Table 8 - 16 lists the Complementary PWM Mode Specifications, and Figure 8 - 61 shows the Output Model of Complementary PWM Mode, and Figure 8 - 62 shows an Operation Example in Complementary PWM Mode.

Figure 8 - 60 Block Diagram of Complementary PWM Mode

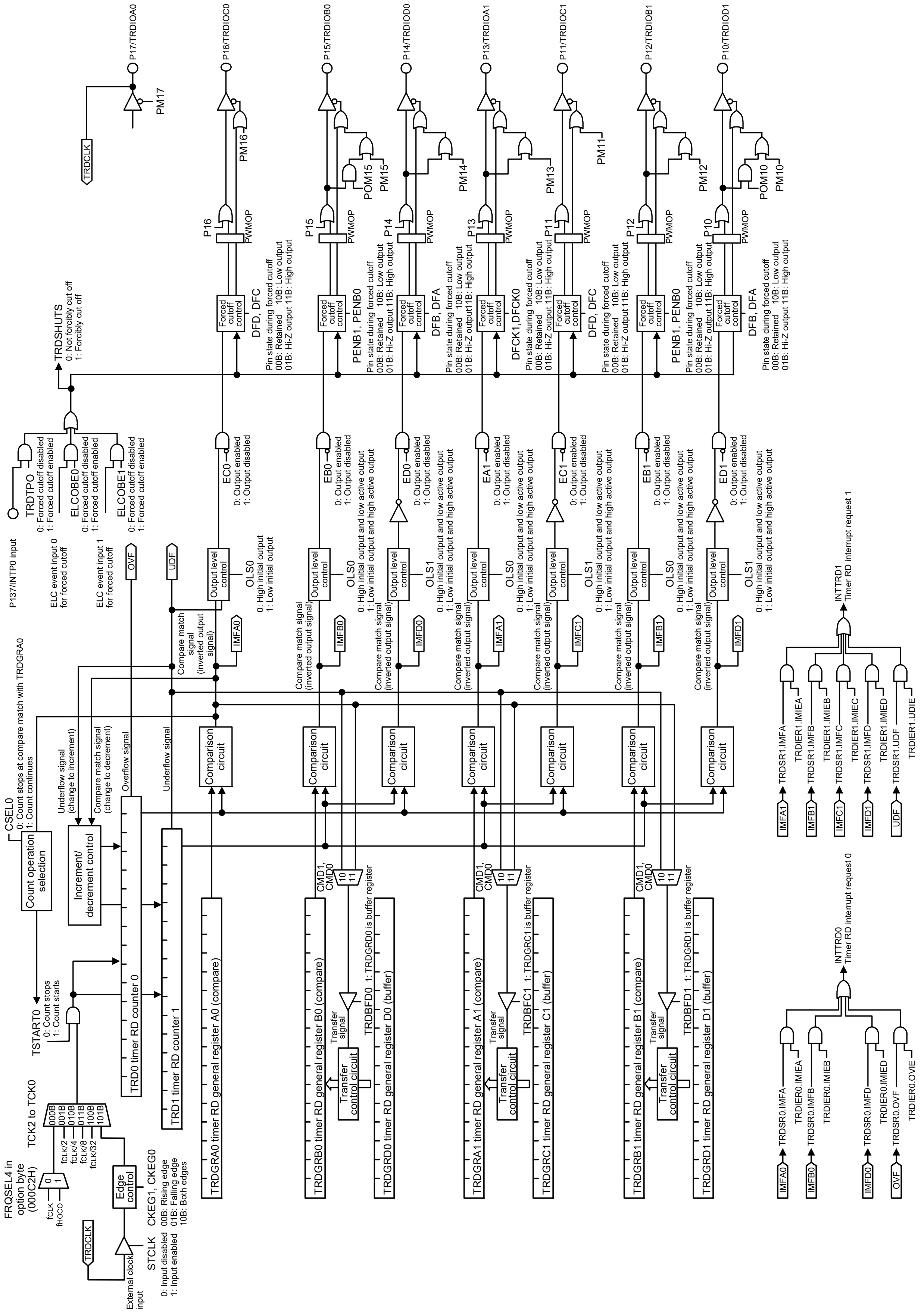
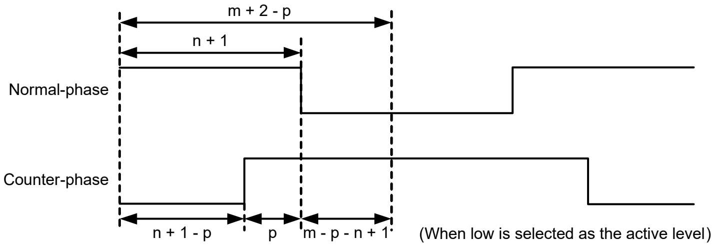


Table 8 - 16 Complementary PWM Mode Specifications

Item	Specification
Count sources	f_{HOCO} Note 1, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRDCLK pin (active edge selected by a program) Set bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as bits TCK2 to TCK0 in the TRDCR0 register.
Count operations	Increment or decrement. Registers TRD0 and TRD1 are decremented with the compare match with registers TRD0 and TRDGRA0 during increment operation. When the TRD1 register changes from 0000H to FFFFH during decrement operation, and registers TRD0 and TRD1 are incremented.
PWM operations	PWM period: 1/fk × (m + 2 - p) × 2 Note 2 Dead time: p Active level width of normal-phase: 1/fk × (m - n - p + 1) × 2 Active level width of counter-phase: 1/fk × (n + 1 - p) × 2 fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) p: Value set in the TRD0 register 
Count start condition	1 (count starts) is written to bits TSTART0 and TSTART1 in the TRDSTR register.
Count stop condition	0 (count stops) is written to bits TSTART0 and TSTART1 in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. (The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.)
Interrupt request generation timing	• Compare match (content of the TRDi register matches content of the TRDGRji register) • TRD1 register underflow
TRDIOA0 pin function	I/O port or TRDCLK (external clock) input
TRDIOB0 pin function	PWM1 output normal-phase output
TRDIOD0 pin function	PWM1 output counter-phase output
TRDIOA1 pin function	PWM2 output normal-phase output
TRDIOC1 pin function	PWM2 output counter-phase output
TRDIOB1 pin function	PWM3 output normal-phase output
TRDIOD1 pin function	PWM3 output counter-phase output
TRDIOC0 pin function	Output inverted every 1/2 period of PWM
INTP0 pin function	Pulse output forced cutoff signal input (I/O port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi register.
Write to timer	The value can be written to the TRDi register.
Selectable functions	• Pulse output forced cutoff signal input (see **8.4.4 Pulse Output Forced Cutoff**) • The normal-phase and counter-phase active level and initial output level are selected individually. • Transfer timing from the buffer register selection

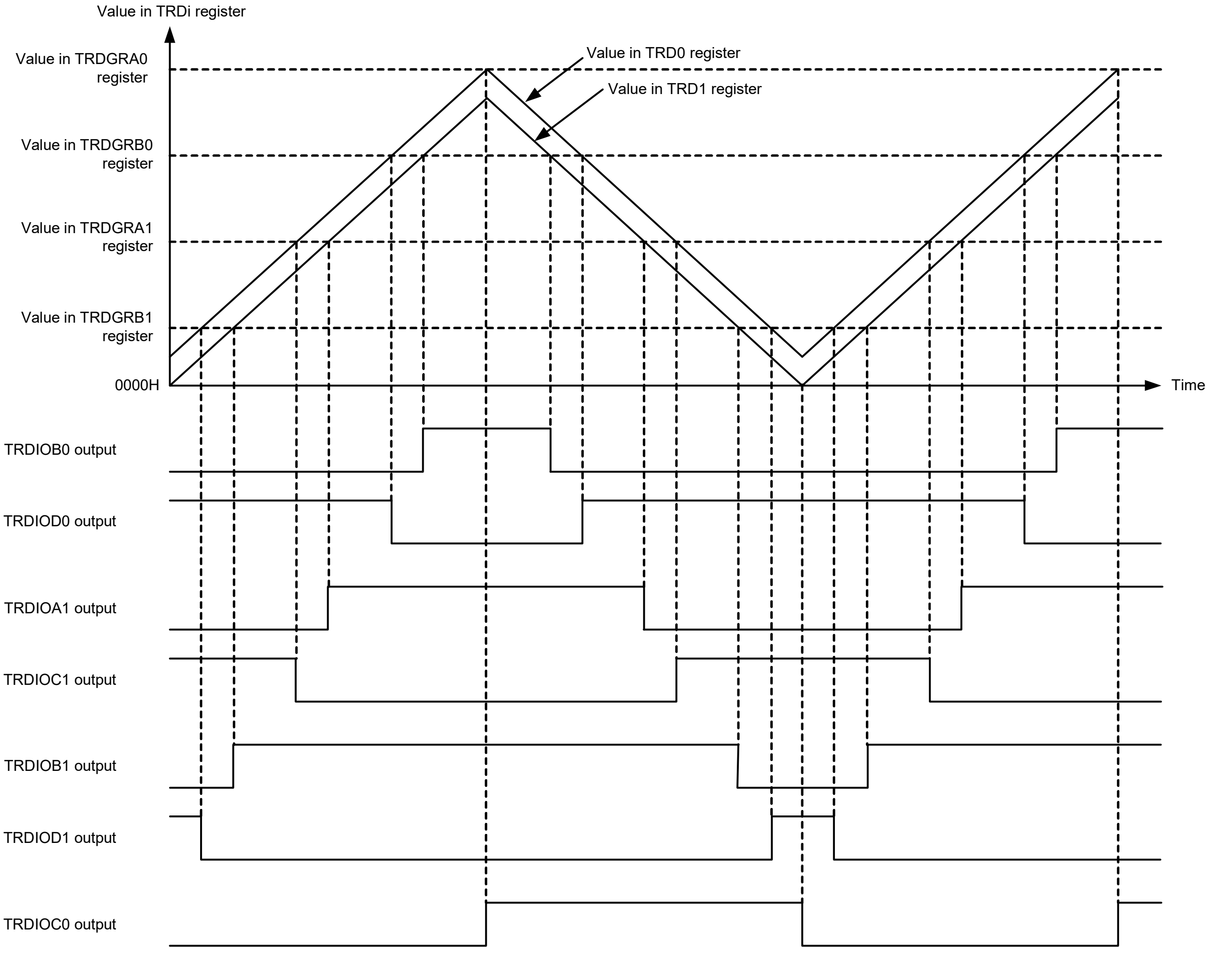
Note 1. f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Note 2. After a count starts, the PWM period is fixed.

Remark i = 0 or 1, j = A, B, C, or D

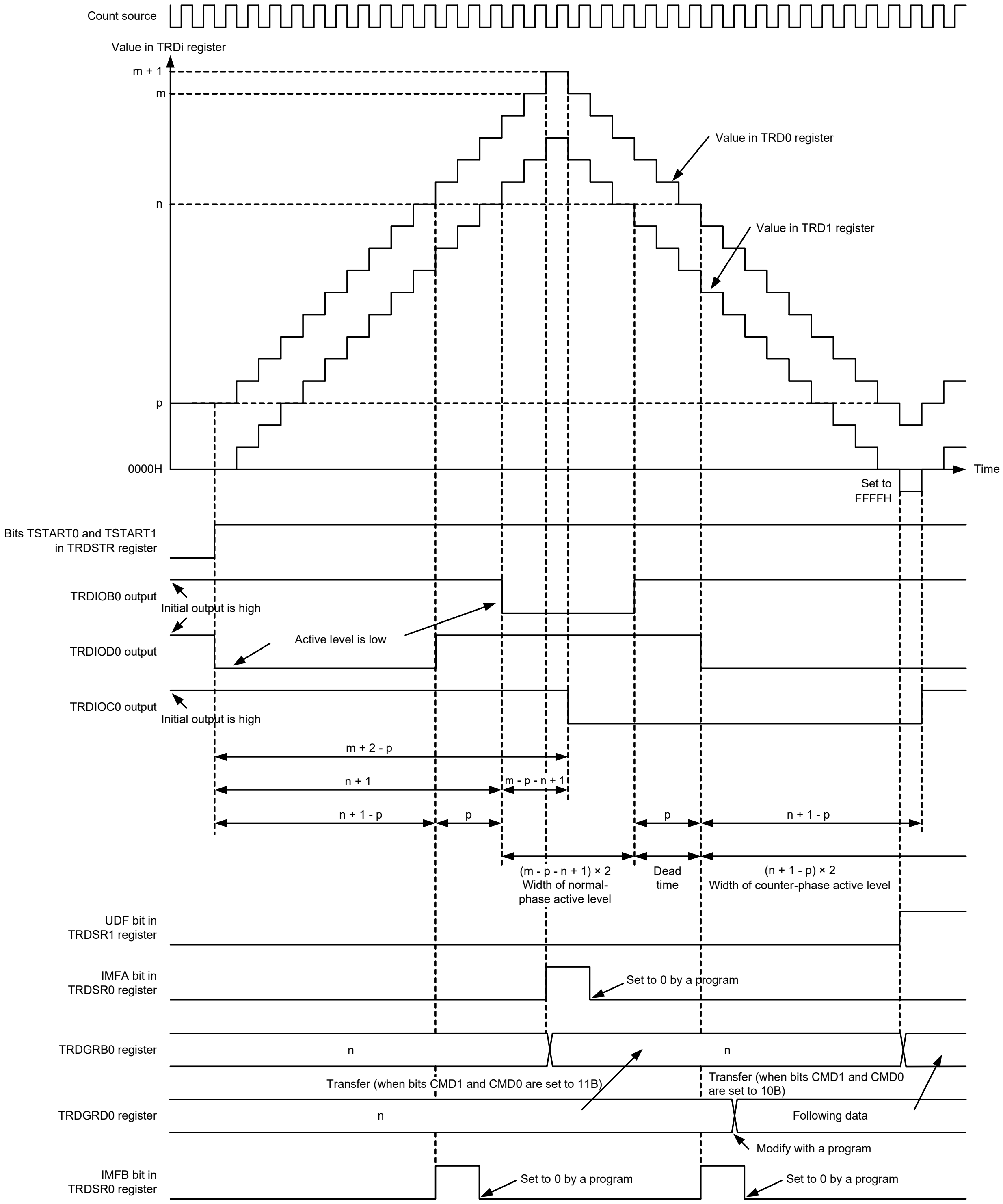
(1) Operation Example

Figure 8 - 61 Output Model of Complementary PWM Mode



Remark i = 0 or 1

Figure 8 - 62 Operation Example in Complementary PWM Mode



Remark
CMD0, CMD1: Bits in TRDFCR register
i = 0 or 1
m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
p: Value set in TRD0 register

The above diagram applies under the following condition:
Bits OLS1 and OLS0 in TRDFCR are set to 0 (initial output level is high, active level is low for normal-phase and counter-phase).

(2) Transfer Timing from Buffer Register

- Transfer from the TRDGRD0, TRDGRC1, or TRDGRD1 register to the TRDGRB0, TRDGRA1, or TRDGRB1 register.

  When bits CMD1 and CMD0 in the TRDFCR register are set to 10B, the content is transferred when the TRD1 register underflows.

  When bits CMD1 and CMD0 are set to 11B, the content is transferred at compare match between registers TRD0 and TRDGRA0.

8.5.6 PWM3 mode

In this mode, two PWM waveforms are output with the same period.

Figure 8 - 63 shows the Block Diagram of PWM3 Mode, Table 8 - 17 lists the PWM3 Mode Specifications, and Figure 8 - 64 shows an Operation Example in PWM3 Mode.

Figure 8 - 63 Block Diagram of PWM3 Mode

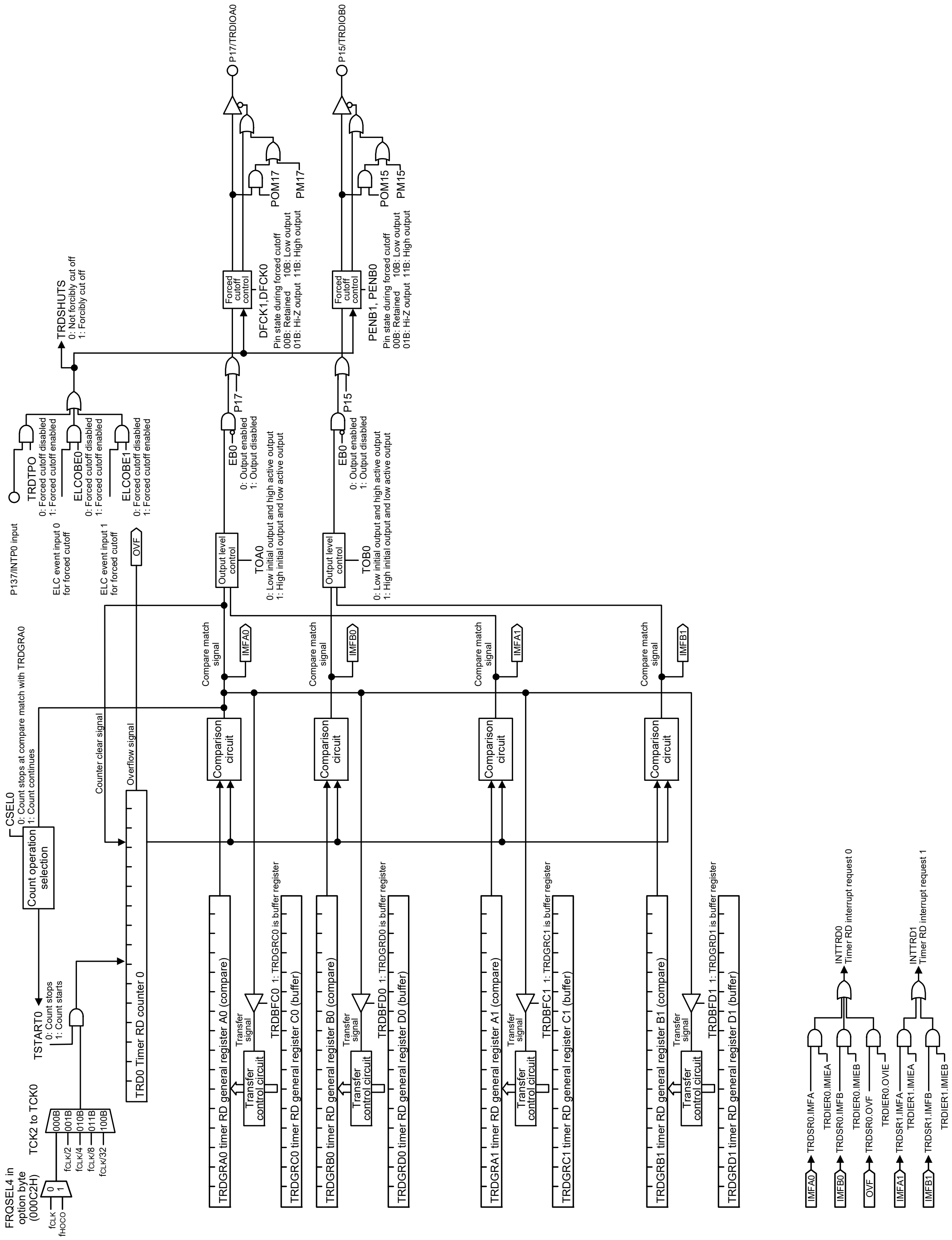
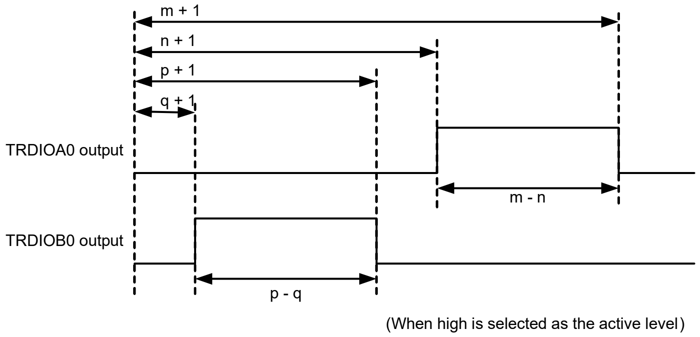


Table 8 - 17 PWM3 Mode Specifications

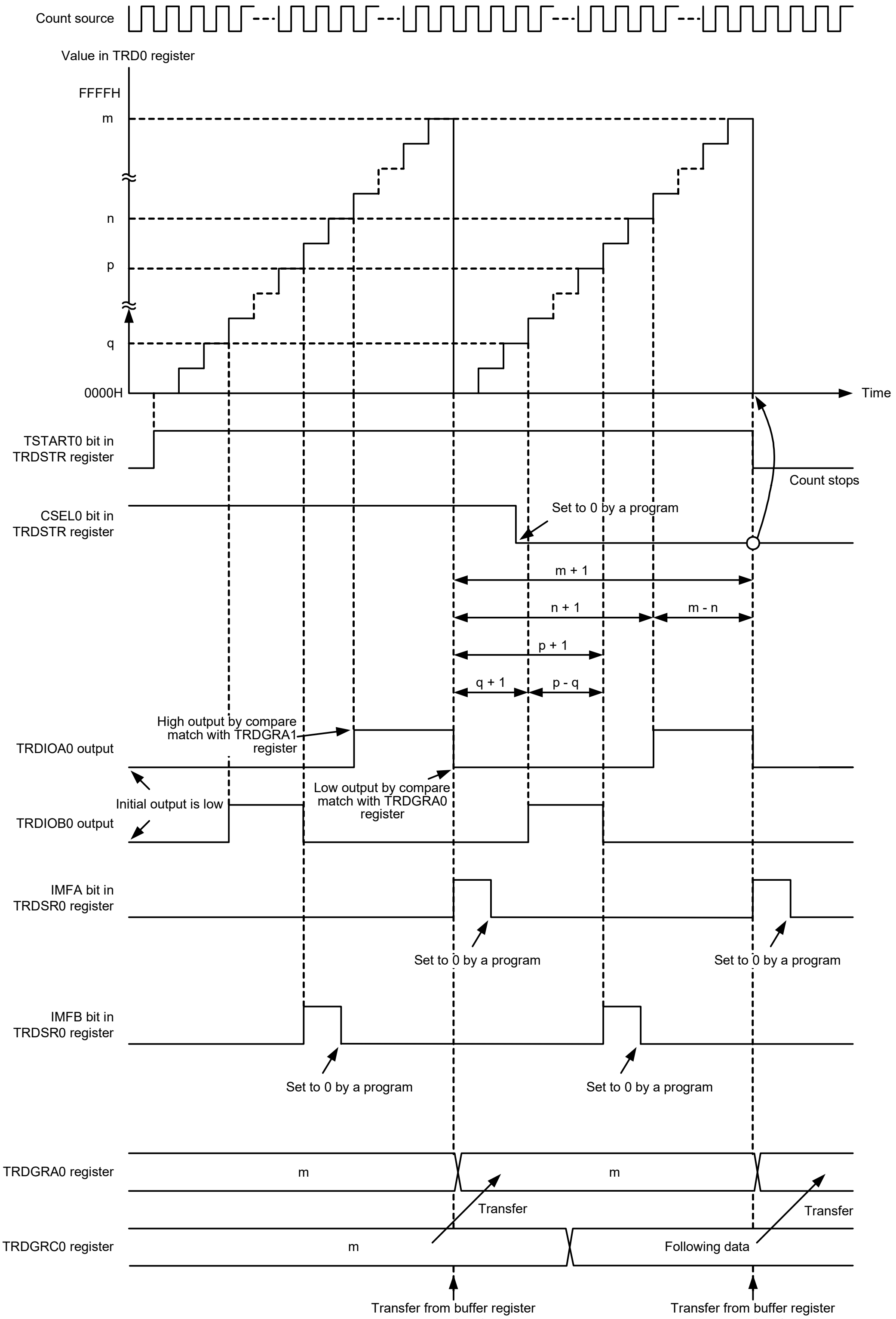
Item	Specification
Count sources	f_{HOCO} Note, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$
Count operations	The TRD0 register is incremented (the TRD1 register is not used).
PWM waveform	PWM period: 1/fk × (m + 1) Active level width of TRDIOA0 output: 1/fk × (m - n) Active level width of TRDIOB0 output: 1/fk × (p - q) fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRA1 register p: Value set in the TRDGRB0 register q: Value set in the TRDGRB1 register 
Count start condition	1 (count starts) is written to the TSTART0 bit in the TRDSTR register.
Count stop conditions	• 0 (count stops) is written to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. The PWM output pin holds the output level before the count stops. • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRA0 register. The PWM output pin holds the level after output change by compare match.
Interrupt request generation timing	• Compare match (content of the TRDi register matches content of the TRDGRji register) • TRD0 register overflow
TRDIOA0, TRDIOB0 pin function	PWM output
TRDIOA0, TRDIOD0, and TRDIOA1 to TRDIOD1 pin function	I/O port
INTP0 pin function	Pulse output forced cutoff signal input (I/O port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRD0 register.
Write to timer	The value can be written to the TRD0 register.
Selectable functions	• Pulse output forced cutoff signal input (see **8.4.4 Pulse Output Forced Cutoff**) • Active level selectable for each pin • Buffer operation (see **8.4.2 Buffer Operation**)

Note f_{HOCO} is selected only when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f_{HOCO} as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

Remark i = 0 or 1, j = A, B, C, or D

(1) Operation Example

Figure 8 - 64 Operation Example in PWM3 Mode



Remark
j = A or B
m: Value set in TRDGRA0 register
n: Value set in TRDGRA1 register
p: Value set in TRDGRB0 register
q: Value set in TRDGRB1 register

The above diagram applies under the following conditions:

- Both the TOA0 and TOB0 bits in the TRDOCR register are set to 0 (initial output is low, high output by compare match with TRDGRj1 register, low output by compare match with TRDGRj0 register).
- The TRDBFC0 bit in the TRDMR register is set to 1 (TRDGRC0 register is buffer register for TRDGRA0 register).

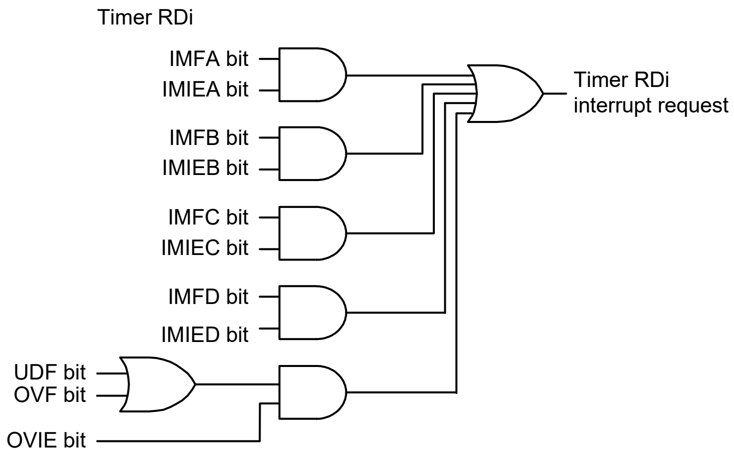
8.6 Timer RD Interrupt

Timer RD generates the timer RDi (i = 0 or 1) interrupt request from six sources for each timer RD0 and timer RD1. Table 8 - 18 lists the Registers Associated with Timer RD Interrupt and Figure 8 - 65 shows the Timer RD Interrupt Block Diagram.

Table 8 - 18 Registers Associated with Timer RD Interrupt

	Timer RD Status Register	Timer RD Interrupt Enable Register	Interrupt Request Flag (Register)	Interrupt Mask Flag (Register)	Priority Specification Flag (Register)
Timer RD0	TRDSR0	TRDIER0	TRDIF0 (IF2H)	TRDMK0 (MK2H)	TRDPR00 (PR02H) TRDPR10 (PR12H)
Timer RD1	TRDSR1	TRDIER1	TRDIF1 (IF2H)	TRDMK1 (MK2H)	TRDPR01 (PR02H) TRDPR11 (PR12H)

Figure 8 - 65 Timer RD Interrupt Block Diagram



i = 0 to 1
IMFA, IMFB, IMFC, IMFD, OVF, UDF: TRDSRi register bit
IMIEA, IMIEB, IMIEC, IMIED, OVIE: TRDIERi register bit

Since the interrupt source (timer RD interrupt) is generated by a combination of multiple interrupt request sources for timer RD, the following differences from other maskable interrupts apply:

- When a bit in the TRDSRi register is 1 and the corresponding bit in the TRDIERi register is 1 (interrupt enabled), the TRDIFi bit in the IF2H register is set to 1 (interrupt requested).
- If multiple bits in the TRDIERi register are set to 1, use the TRDSRi register to determine the source of the interrupt request.
- Since the bits in the TRDSRi register are not automatically set to 0 even if the interrupt is acknowledged, set the corresponding bit to 0 in the interrupt routine.

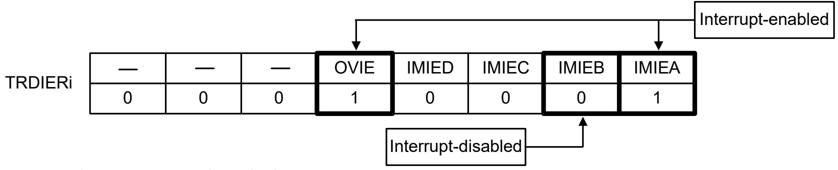
- When status flags of interrupt sources (applicable status flags) of the timer RD are set to 0 and their interrupts are disabled in timer RD interrupt enable register i (TRDIERi), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

(b) When there are bits set to 1 (enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

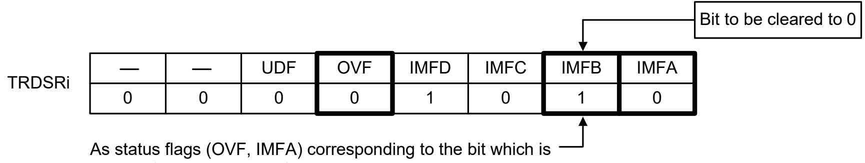
Example: To clear the IMFB bit to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).

• Timer RD Interrupt Enable Register i (TRDIERi)



TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

• Timer RD Status Register i (TRDSRi)



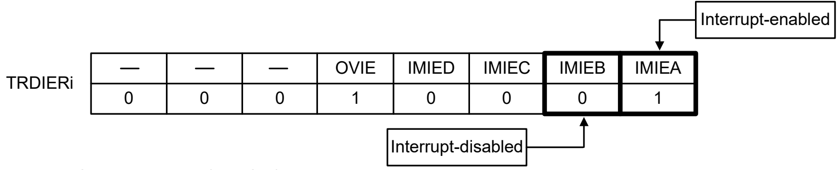
TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	0

As status flags (OVF, IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) are 0, write 0 to the IMFB bit.

(c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

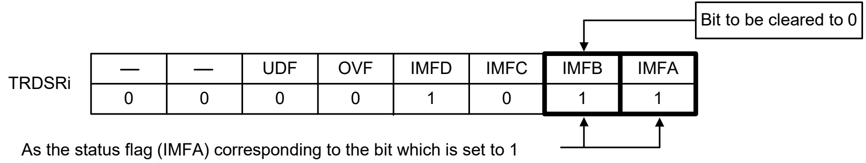
Example: To clear the IMFB bit to 0 when the IMIEA is set to 1 (interrupt-enabled) and the IMIEB is set to 0 (interrupt-disabled).

• Timer RD Interrupt Enable Register i (TRDIERi)



TRDIERi	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
	0	0	0	1	0	0	0	1

• Timer RD Status Register i (TRDSRi)



TRDSRi	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
	0	0	0	0	1	0	1	1

As the status flag (IMFA) corresponding to the bit which is set to 1 (interrupt-enabled) is 1, write 0 to bits IMFB and IFMA at the same time.

8.7 Notes on Timer RD

8.7.1 SFR read/write access

When setting timer RD, set the TRD0EN bit in the PER1 register to 1 first. If the TRD0EN bit is 0, writes to the timer RD control registers are ignored and all the read values are the initial values (except for the port registers and the port mode registers).

The following registers must not be rewritten during count operation:
TRDELC, TRDMR, TRDPMR, TRDFCR, TRDOER1, TRDPTO bit in TRDOER2, TRDOCR, TRDDFi, TRDCRi, TRDIORAi, TRDIORCi, TRDPOCRi

(1) TRDSTR Register

- The TRDSTR register can be set by an 8-bit memory manipulation instruction.
- When the CSELi bit (i = 0 or 1) in the TRDSTR register is set to 0 (count stops at compare match between registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.
  The TSTARTi bit is set to 0 (count stops) only by a compare match with the TRDGRAi register.
  If the CSELi bit is 0 when rewriting the TRDSTR register, write 0 to the TSTARTi bit to change the CSELi bit to 1 without affecting count operation.
  If 1 is written to the TSTARTi bit while the counter is stopped, count may be started.
  To stop counting by a program, set the TSTARTi bit after setting the CSELi bit to 1. Even if 1 is written to the CSELi bit and 0 is written to the TSTARTi bit at the same time (using one instruction), the count cannot be stopped.
- Table 8 - 19 lists the TRDIOji (j = A, B, C, or D) Pin Output Level When Count Stops while using the TRDIOji (j = A, B, C, or D) pin for timer RD output.

Table 8 - 19 TRDIOji (j = A, B, C, or D) Pin Output Level When Count Stops

Count Stop	TRDIOji Pin Output When Count Stops
When the CSELi bit is set to 1, write 0 to the TSTARTi bit and the count stops.	The pin holds the output level immediately before the count stops. (The pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register in timer RD complementary and reset synchronous PWM modes.)
When the CSELi bit is set to 0, the count stops at compare match with registers TRDi and TRDGRAi.	The pin holds the output level after the output changes by compare match. (The pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register in timer RD complementary and reset synchronous PWM modes.)

Remark i = 0 or 1, j = A, B, C, or D

(2) TRDDFi Register (i = 0 or 1)
Set bits DFCK0 and DFCK1 in the TRDDFi register before starting count operation.

(3) TRDi Register (i = 0 or 1)

- If the TRDi register is set to 0000H and a value is written to the TRDi register at the same timing, the value written to the register has priority.

8.7.2 Mode switching

- Set the count to stopped (set bits TSTART0 and TSTART1 to 0) before switching modes during operation.
- Set bits TRDIF0 and TRDIF1 to 0 before changing bits TSTART0 and TSTART1 from 0 to 1. Refer to **CHAPTER 16 INTERRUPT FUNCTIONS** for details.

8.7.3 Count source

- Switch the count source after the count stops.

[Changing procedure]
(1) Set the TSTARTi bit (i = 0 or 1) in the TRDSTR register to 0 (count stops).
(2) Change bits TCK0 to TCK2 in the TRDCRi register.

- When selecting f_{HOCO} (48 MHz) as the count source for timer RD, set f_{CLK} to f_{IH} before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.

8.7.4 Input capture function

- Set the pulse width of the input capture signal to three or more cycles of the timer RD operating clock.
- The value of the TRDi register is transferred to the TRDGRji register two to three cycles of the timer RD operating clock (f_{CLK}) after the input capture signal is applied to the TRDIOji pin (i = 0 or 1, j = A, B, C, or D) (when no digital filter is used).
- In input capture mode, an input capture interrupt request for the active edge of the TRDIOji input is also generated when the TSTARTi bit in the TRDSTR register is 0 (count stops) if the edge selected by bits IOj0 and IOj1 in the TRDIORki register is input to the TRDIOji pin (i = 0 or 1; j = A, B, C, or D, k = A or C).

8.7.5 Procedure for setting pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (i = 0 or 1)

After a reset, the I/O ports multiplexed with pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi function as input ports.

- To output from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi, use the following setting procedure:

Changing procedure

(1) Set the mode and the initial value.
(2) Enable output from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (TRDOER1 register).
(3) Set the port register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to 0.
(4) Set the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to output mode. (Output is started from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi)
(5) Start the count (set bits TSTART0 and TSTART1 to 1).

- To change the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi from output mode to input mode, use the following setting procedure:

Changing procedure

(1) Set the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to input mode (input is started from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi).
(2) Set to the input capture function.
(3) Start the count (set bits TSTART0 and TSTART1 to 1).

- When switching pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi from output mode to input mode, input capture operation may be performed depending on the pin states. When the digital filter is not used, edge detection is performed after two or more cycles of the operation clock have elapsed. When the digital filter is used, edge detection is performed after up to five cycles of the sampling clock.

8.7.6 External clock TRDCLK

Set the pulse width of the external clock applied to the TRDCLK pin to three or more cycles of the timer RD operating clock.

8.7.7 Reset synchronous PWM mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

[Changing procedure]

(1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer mode, PWM mode, and PWM3 mode).
(3) Set bits CMD1 and CMD0 to 01B (reset synchronous PWM mode).
(4) Set the other registers associated with timer RD again.

8.7.8 Complementary PWM mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD0 and CMD1 in the TRDFCR register in the following procedure.

Changing procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

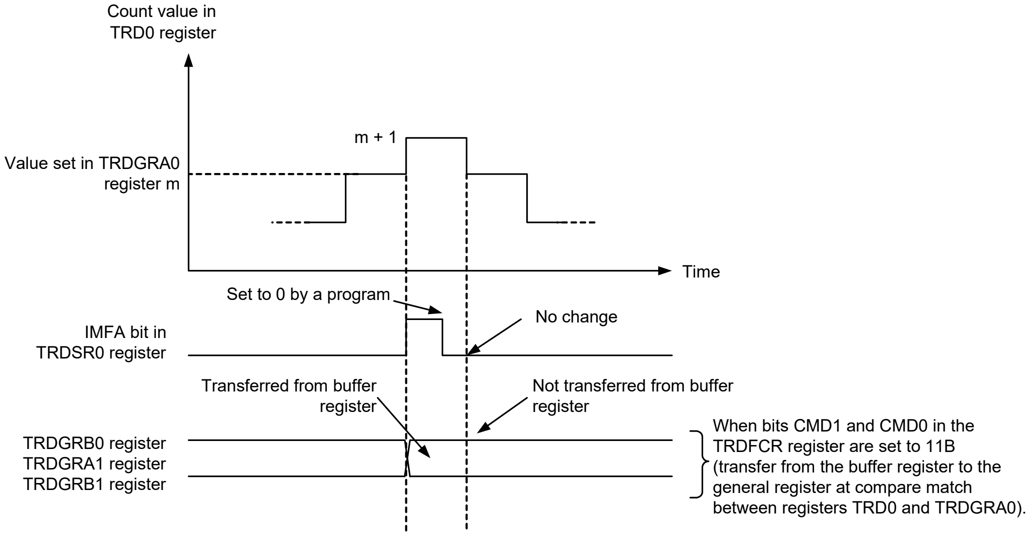
(1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer mode, PWM mode, and PWM3 mode).
(3) Set bits CMD1 and CMD0 to 10B or 11B (complementary PWM mode).
(4) Set the registers associated with other timer RD again.

Changing procedure: When stopping complementary PWM mode

(1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 to CMD0 to 00B (timer mode, PWM mode, and PWM3 mode).

- Do not write to the TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation.
  When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 using the buffer operation.
  However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits TRDBFD0, TRDBFC1, and TRDBFD1 to 0 (general register). After this, bits TRDBFD0, TRDBFC1, and TRDBFD1 may be set to 1 (buffer register).
  The PWM period cannot be changed.
- If the value set in the TRDGRA0 register is assumed to be m, the TRD0 register counts m - 1, m, m + 1, m, m - 1, in that order, when changing from increment to decrement operation.
  When changing from m to m + 1, the IMFA bit in the TRDSRi register is set to 1. Also, bits CMD1 and CMD0 in the TRDFCR register are set to 11B (complementary PWM mode, buffer data transferred at compare match between registers TRD0 and TRDGRA0), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).
  During operation of m + 1, m, and m - 1, the IMFA bit remains unchanged and data is not transferred to registers such as the TRDGRA0 register.

Figure 8 - 66 Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode



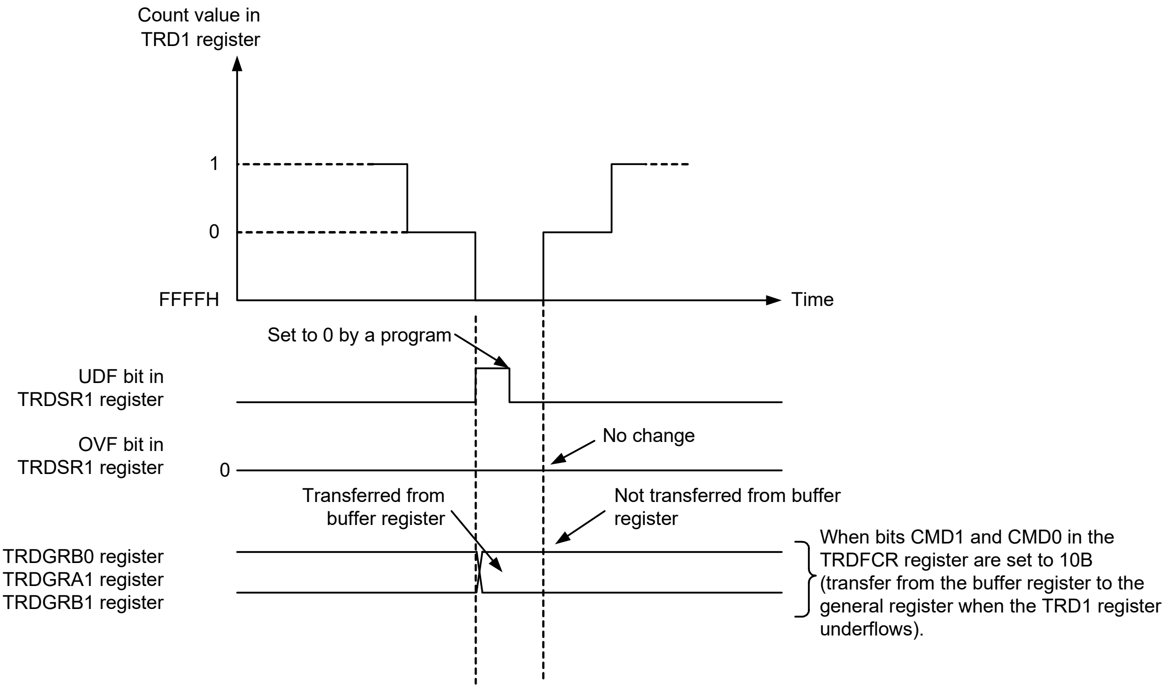
- The TRD1 register counts 1, 0, FFFFH, 0, 1, in that order, when changing from decrement to increment operation.

  Counting from 1, to 0, to FFFFH causes the UDF bit in the TRDSRi register to be set to 1. Also, when bits CMD1 and CMD0 in the TRDFCR register are set to 10B (complementary PWM mode, buffer data transferred at underflow of the TRD1 register), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).

  During operation of FFFFH, 0, and 1, data is not transferred to registers such as the TRDGRB0 register.

  Also, at this time, the OVF bit in the TRDSRi register remains unchanged.

Figure 8 - 67 Operation When TRD1 Register Underflows in Complementary PWM Mode

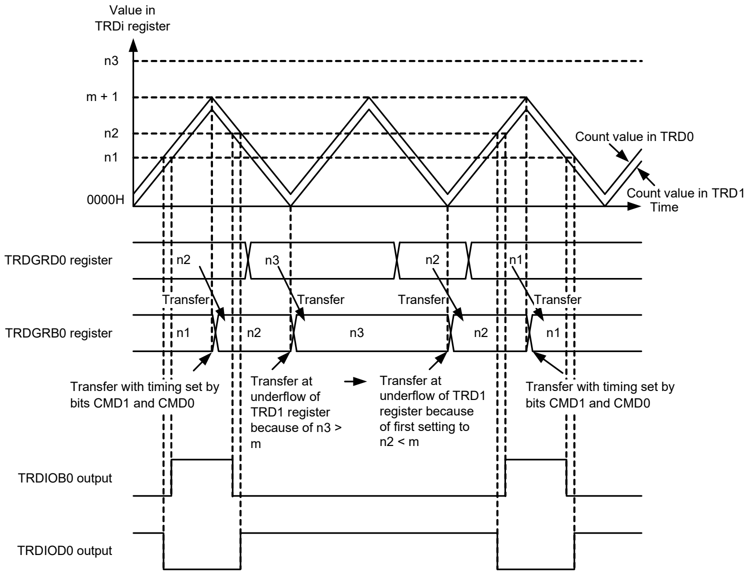


- The timing of data transfer from the buffer register to the general register should be selected using bits CMD0 and CMD1 in the TRDFCR register. However, regardless of the values of bits CMD0 and CMD1, transfer takes place with the following timing when duty cycle is 0% and duty cycle is 100%.

  Value in buffer register ≥ value in TRDGRA0 register (duty cycle is 0%):
  Transfer take place at underflow of the TRD1 register.
  After this, when the buffer register is set to 0001H or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 and CMD0.

  However, no waveform with duty cycle 0% can be generated while the initial value of the buffer register is FFFFH. To generate a waveform with duty cycle 0%, set the value of the buffer register ≥ TRDGRA0 by writing to the buffer register.

Figure 8 - 68 Operation When Value in Buffer Register ≥ Value in TRDGRA0 Register in Complementary PWM Mode



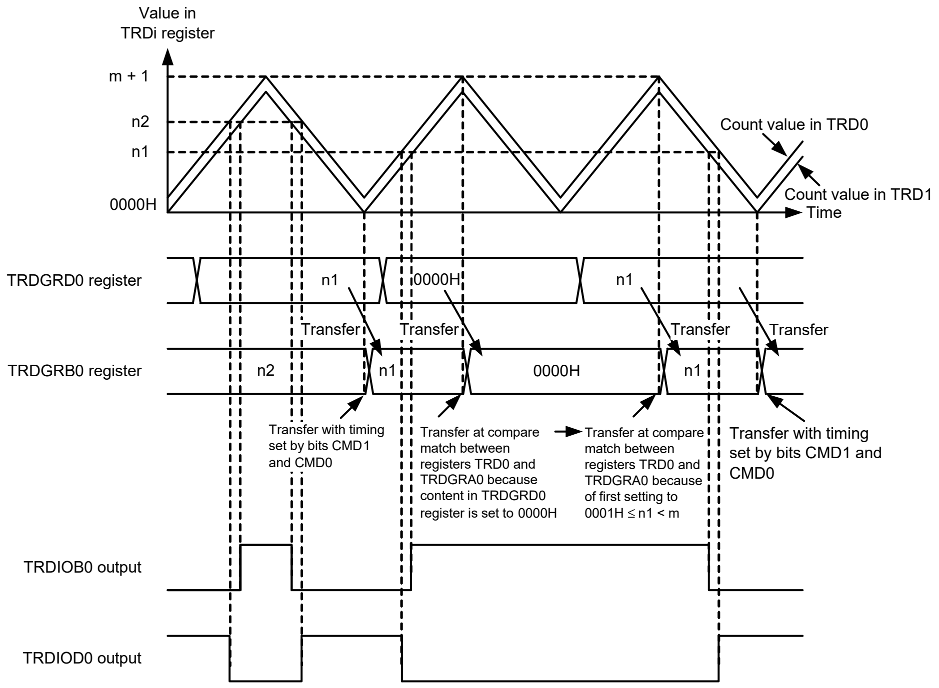
Remark
m: Value set in TRDGRA0 register

The above diagram applies under the following conditions:
- Bits CMD1 and CMD0 in the TRDFCR register are set to 11B
  (data in the buffer register is transferred at compare match between registers TRD0 and TRDGRA0 in complementary PWM mode).
- Both the OSL0 and OLS1 bits in the TRDFCR register are set to 1 (active high for normal-phase and counter-phase).

When a value that is larger than or equal to the value of the TRDGRA0 register is written to the buffer register, the value of the buffer register is transferred to the general register at underflow of the TRD1 counter, and the output level is fixed to normal-phase with 100% duty cycle and counter-phase with 0% duty cycle.
To cancel the fixed output level, write a value that is larger than or equal to the setting value of the TRD0 register and smaller than or equal to (TRDGRA0 setting value minus TRD0 register setting value) to the buffer register. After the value is written to the buffer register, the value of the buffer register is transferred to the general register at underflow of the TRD1 counter, and a PWM waveform is output regardless of the setting of the CMD0 bit. After a PWM waveform is output, the value of the buffer register is transferred to the general register with the timing specified by the CMD0 bit.
However, the initial value FFFFH of the buffer register cannot be used to set normal-phase output with 100% duty cycle and counter-phase output with 0% duty cycle. Also, while the setting is normal-phase output with 100% duty cycle and counter-phase output with 0% duty cycle, the setting cannot be directly changed to normal-phase output with 0% duty cycle and counter-phase output with 100% duty cycle.

When the value in the buffer register is set to 0000H (duty cycle is 100%):
Transfer takes place at compare match between registers TRD0 and TRDGRA0.
After this, when the buffer register is set to 0001H or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD0 and CMD1.

Figure 8 - 69 Operation When Value in Buffer Register is Set to 0000H in Complementary PWM Mode



Remark
m: Value set in TRDGRA0 register

The above diagram applies under the following conditions:
- Bits CMD1 and CMD0 in the TRDFCR register are set to 10B (data in the buffer register is transferred at underflow of the TRD1 register in PWM mode).
- Both the OLS0 and OLS1 bits in the TRDFCR register are set to 1 (active high for normal-phase and counter-phase).

When 0000H is written to the buffer register, the value of the buffer register is transferred to the general register at a compare match between registers TRD0 and TRDGRA0, and the output level is fixed to normal-phase with 0% duty cycle and counter-phase with 100% duty cycle.
To cancel the fixed output level, write a value that is larger than or equal to the setting value of the TRD0 register and smaller than or equal to (TRDGRA0 setting value minus TRD0 register setting value) to the buffer register. After the value is written to the buffer register, the value of the buffer register is transferred to the general register at underflow of the TRD1 counter, and a PWM waveform is output regardless of the setting of the CMD0 bit. After a PWM waveform is output, the value of the buffer register is transferred to the general register with the timing specified by the CMD0 bit.
The setting of normal-phase output with 0% duty cycle and counter-phase output with 100% duty cycle cannot be directly changed to normal-phase output with 100% duty cycle and counter-phase output with 0% duty cycle.

CHAPTER 9 12-BIT INTERVAL TIMER

9.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

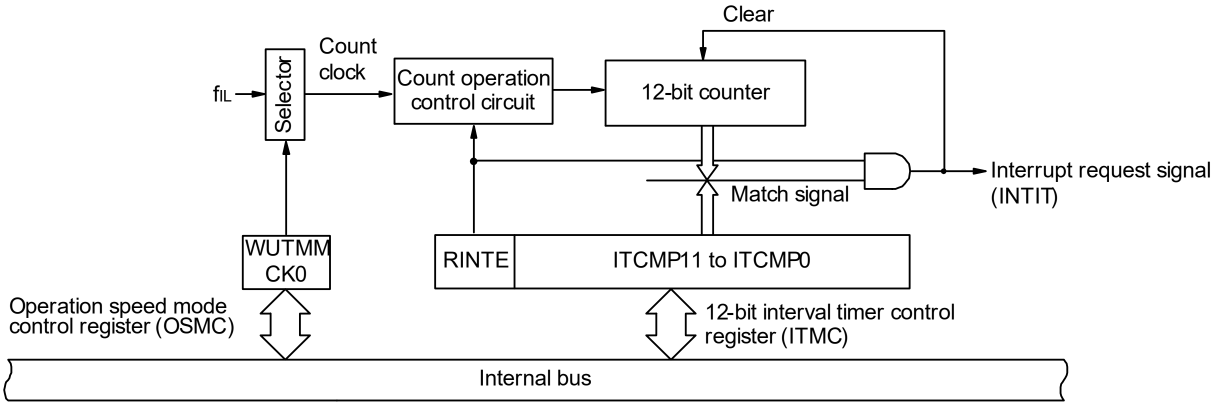
9.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 9 - 1 Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 1 (PER1)
	Operation speed mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 9 - 1 Block Diagram of 12-bit Interval Timer



9.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Operation speed mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

9.3.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the 12-bit interval timer is used, be sure to set bit 7 (TMKAEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
PER1	TMKAEN Note	PWMOPEN	OACMPEN	TRD0EN	0	0	0	TRJ0EN

TMKAEN Note	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read/written.

Note When using the 12-bit interval timer, set bit 4 (WUTMMCK0) in the operation speed mode control register (OSMC) to 1 and start oscillating the low-speed on-chip oscillator clock. Then wait until the low-speed on-chip oscillator stabilizes before setting the TMKAEN bit to 1.

Caution 1. When using the 12-bit interval timer, set TMKAEN = 1 first. If TMKAEN = 0, writing to a control register of the 12-bit interval timer is ignored, and all read values are default values (except for the operation speed mode control register (OSMC)).

Caution 2. Be sure to clear the following bits to 0.
Bits 1 to 3

9.3.2 Operation speed mode control register (OSMC)

The OSMC register can be used to select the low-speed on-chip oscillator as the 12-bit interval timer operating clock or the timer RJ count source.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9 - 3 Format of Operation speed mode control register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0 Notes 1, 2	0	0	0	0

WUTMMCK0 Notes 1, 2	Selection of low-speed on-chip oscillator as 12-bit interval timer operating clock or timer RJ count source
0	• The low-speed on-chip oscillator cannot be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator cannot be selected as the timer RJ count source.
1	• The low-speed on-chip oscillator can be selected as the 12-bit interval timer operating clock. • The low-speed on-chip oscillator can be selected as the timer RJ count source.

Note 1. When using the 12-bit interval timer, be sure to set the WUTMMCK0 bit to 1.

Note 2. When the 12-bit interval timer is operating, do not set the WUTMMCK0 bit to 0.

Caution Be sure to clear bits 0 to 3 and bits 5 to 7 to 0.

9.3.3 12-bit interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 9 - 4 Format of 12-bit interval timer control register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate a fixed-cycle interrupt (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0

- ITCMP11 to ITCMP0 = 001H, count clock: when f_{IL} = 15 kHz
  1/15 [kHz] × (1 + 1) = 0.133333... [ms] ≅ 133.33 [μs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{IL} = 15 kHz
  1/15 [kHz] × (4095 + 1) = 273.066... [ms] ≅ 273.07 [ms]

Caution 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit. When entering HALT mode or STOP mode, confirm the value written to the RINTE bit is applied before entering.

Caution 3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

Caution 4. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0. However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

9.4 12-bit Interval Timer Operation

9.4.1 12-bit interval timer operation timing

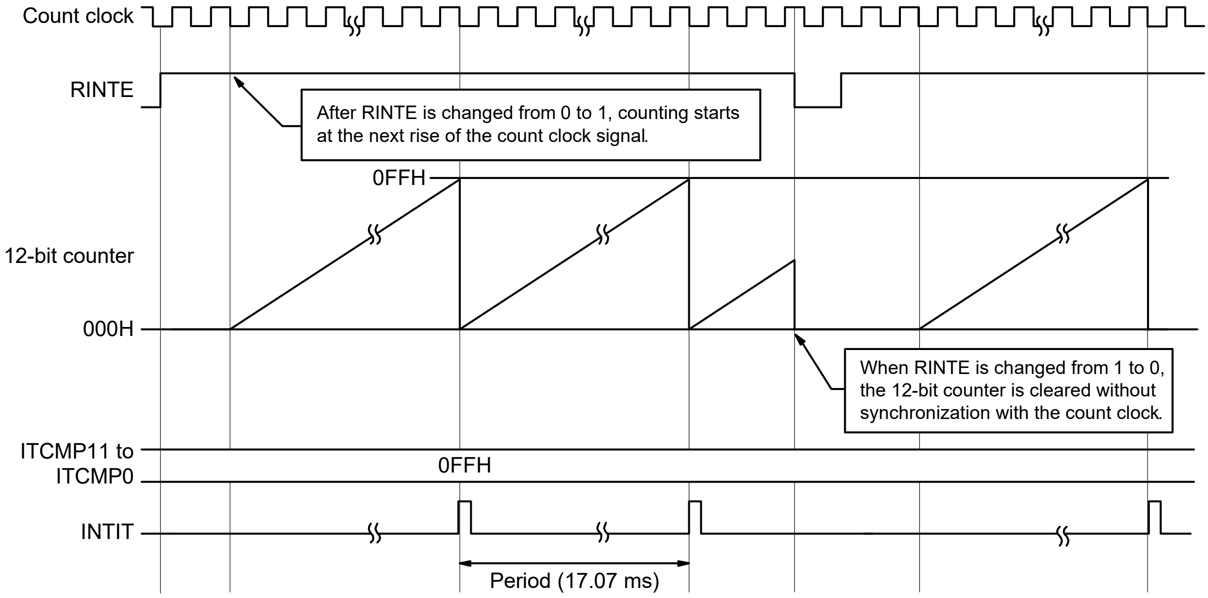
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 9 - 5 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, Count Clock: fIL = 15 kHz)

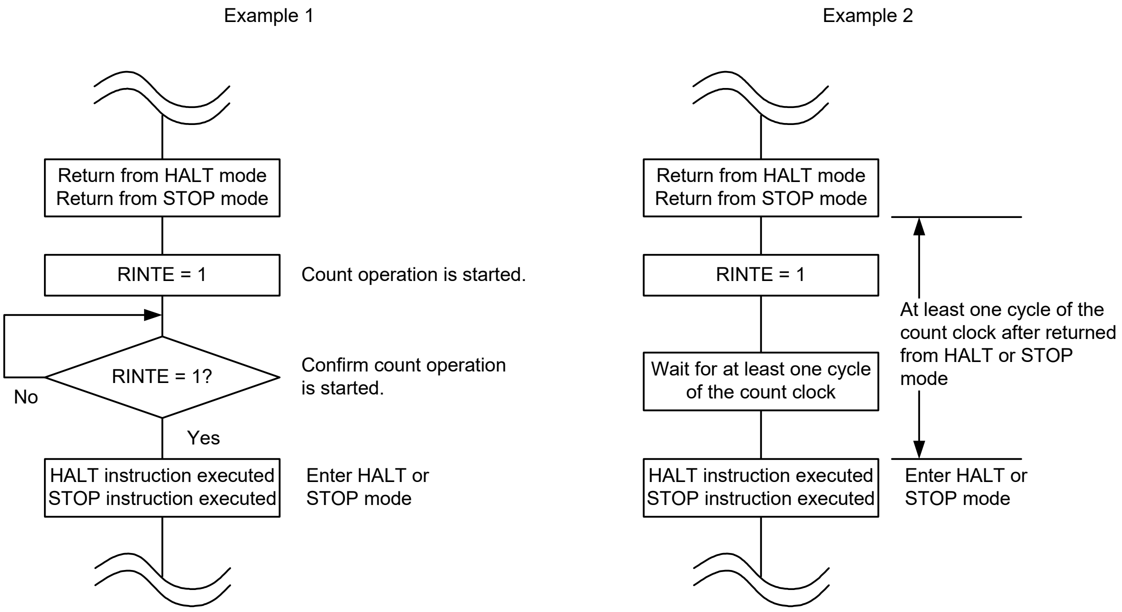


9.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.
Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 9 - 6**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 9 - 6**).

Figure 9 - 6 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

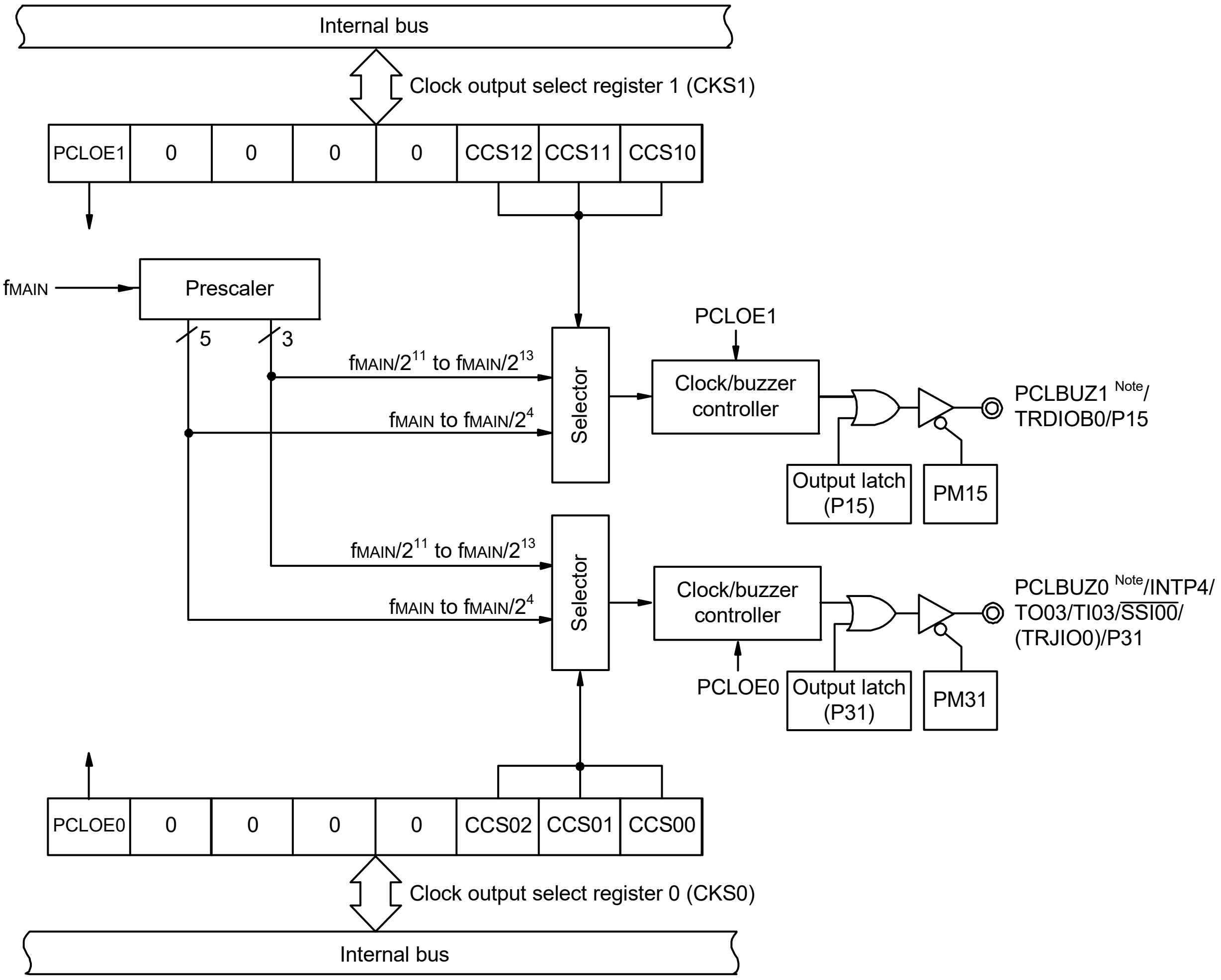
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 10 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Remark n = 0, 1

Figure 10 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **29.5 AC Characteristics**.

10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 10 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Registers controlling port functions of pins to be used for clock or buzzer output Port registers 1, 3 (P1, P3)

10.3 Registers Controlling Clock Output/Buzzer Output Controller

10.3.1 Clock output select registers n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 10 - 2 Format of Clock output select registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	0	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
				f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz
0	0	0	f_{MAIN}	5 MHz	10 MHz Note	Setting prohibited Note	Setting prohibited Note
0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz Note	12 MHz Note
0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz
1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.72 kHz
1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
Other than above			Setting prohibited				

Note Use the output clock within a range of 16 MHz. Furthermore, when using the output clock at 2.7 V ≤ V_{DD} < 4.0 V, can be use it within 8 MHz only. See **29.5 AC Characteristics** for details.

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).

Caution 2. To shift to STOP mode when the main system clock is selected, set PCLOEn = 0 before executing the STOP instruction.

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency

10.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P31/PCLBUZ0/INTP4/TO03/TI03/$\overline{\mathrm{SSI00}}$/(TRJIO0), P15/PCLBUZ1/TRDIOB0) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P31/PCLBUZ0/INTP4/TO03/TI03/$\overline{\mathrm{SSI00}}$/(TRJIO0) is to be used for clock or buzzer output
Set the PM31 bit of port mode register 3 to 0.
Set the P31 bit of port register 3 to 0.

10.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

10.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedures.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Pxx) which correspond to the port which has a pin used as the PCLBUZ0 pin.

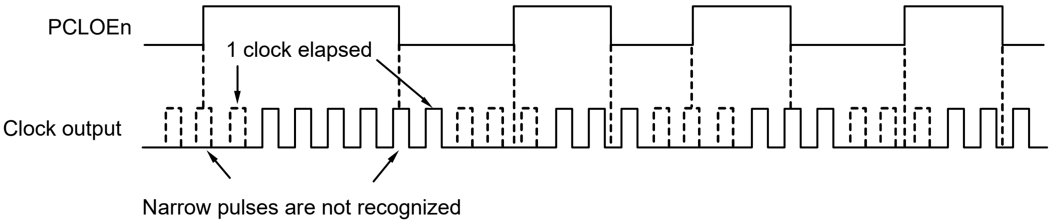
<2> Select the output frequency with bits 0 to 2 (CCSn0 to CCSn2) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 10 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Remark 2. n = 0, 1

Figure 10 - 3 Timing of Outputting Clock from PCLBUZn Pin



10.5 Cautions of Clock Output/Buzzer Output Controller

When the main system clock is selected for the PCLBUZn output (CSEL = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).
The watchdog timer operates on the low-speed on-chip oscillator clock.
The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.
Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1.
For details of the RESF register, see **CHAPTER 19 RESET FUNCTION**.
When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 11 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

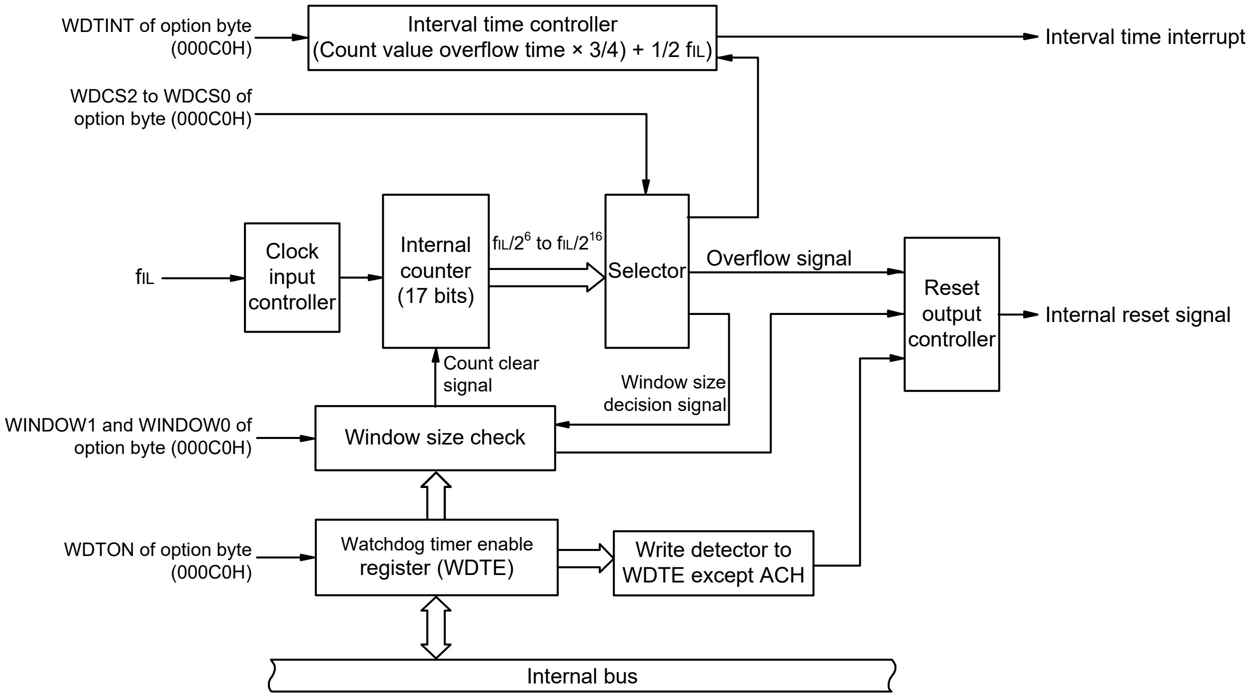
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 11 - 2 Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 24 OPTION BYTE**.

Figure 11 - 1 Block Diagram of Watchdog Timer



11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

11.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 9AH or 1AH [Note].

Figure 11 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 9AH/1AH [Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Caution 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

- Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 24**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

- Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **11.4.2** and **CHAPTER 24**).
- Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **11.4.3** and **CHAPTER 24**).

2. After a reset release, the watchdog timer starts counting.
3. By writing "ACH" to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without "ACH" written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

- If a 1-bit manipulation instruction is executed on the WDTE register
- If data other than "ACH" is written to the WDTE register

Caution 1.When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Caution 2. After "ACH" is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

11.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).
If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.
The following overflow times can be set.

Table 11 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
Other than above			Setting prohibited

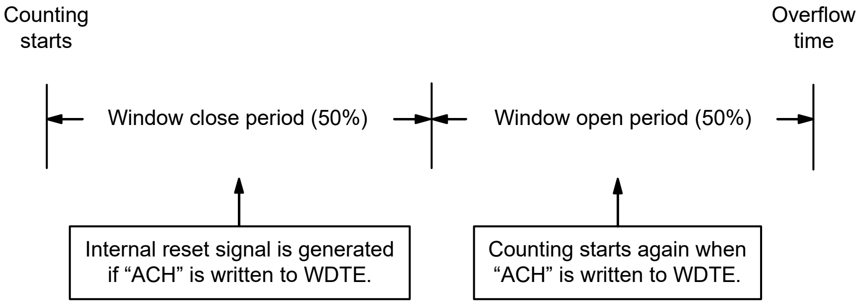
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

11.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution **When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.**

The window open period can be set is as follows.

Table 11 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75% Note
1	1	100%

<R>

<R> **Note** When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Caution **When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.**

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 - 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 - 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

11.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% + 1/2f_{IL} of the overflow time is reached.

Table 11 - 5 Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of overflow time is reached.

Caution **When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.**
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 12 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	30, 32-pin	44-pin
Analog input channels	8 ch (ANI0 to ANI3, ANI16 to ANI19)	12 ch (ANI0 to ANI7, ANI16 to ANI19)

12.1 Function of A/D Converter

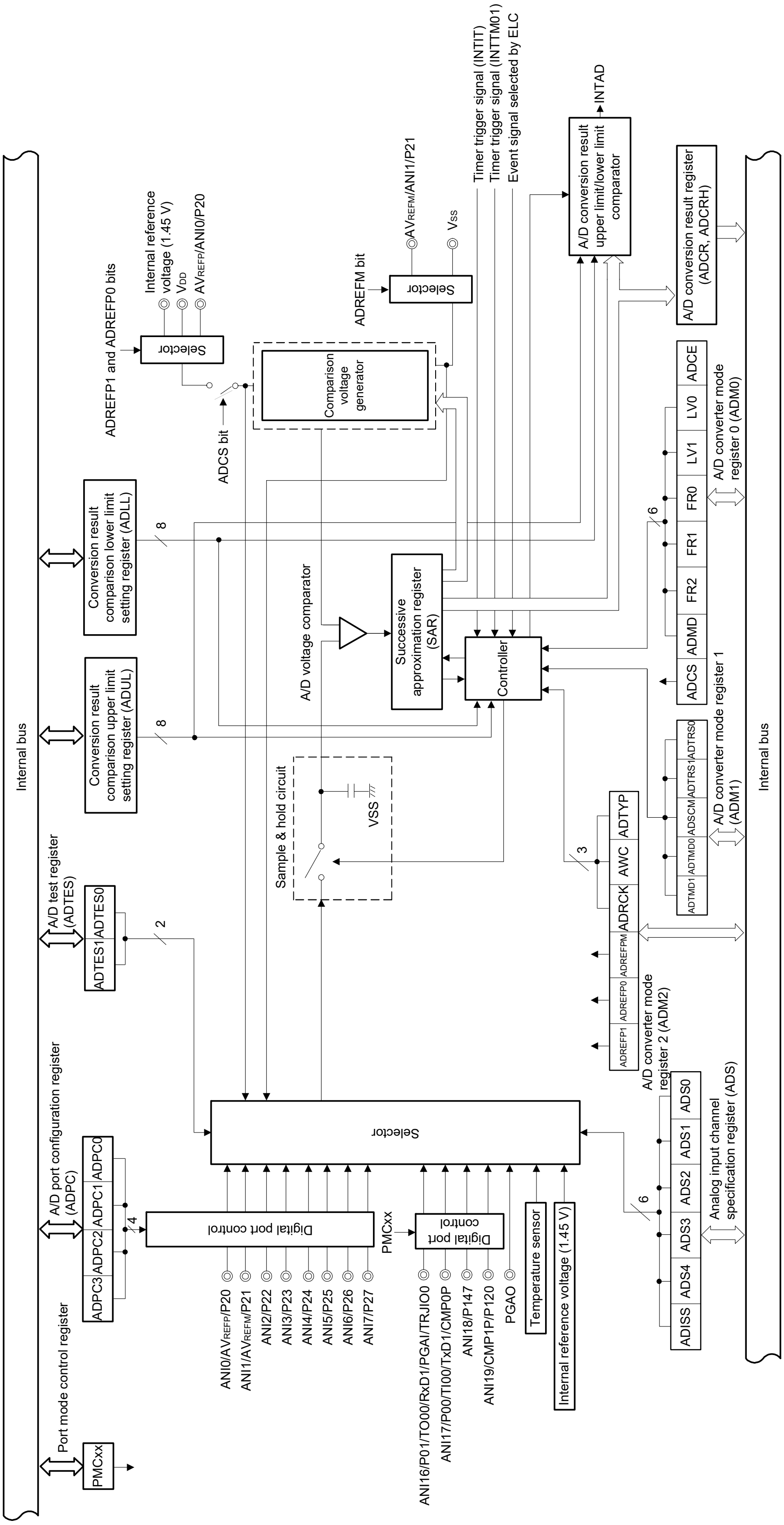
The A/D converter is a converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to twelve channels of A/D converter analog inputs (ANI0 to ANI7 and ANI16 to ANI19).
10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

- 10-bit or 8-bit resolution A/D conversion
  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI7 and ANI16 to ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI7 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of $2.7 V \leq V_{DD} \leq 5.5 V$.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 12 - 1 Block Diagram of A/D Converter



Remark Analog input pin for Figure 12 - 1 when a 44-pin product is used.

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 and ANI16 to ANI19 pins

These are the analog input pins of the 12 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) PGAO

This is an internal output pin of the programmable gain amplifier. In the A/D converter, an output signal from the programmable gain amplifier can be selected as analog input and A/D converted.

(3) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(4) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AVREF) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AVREF), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AVREF)
Bit 9 = 1: (3/4 AVREF)

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage (1.45 V), and VDD.

(5) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(6) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(7) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(8) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(9) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.

(10) AVREFP pin

This pin inputs an external reference voltage (AVREFP).

If using AVREFP as the + side reference voltage of the A/D converter, set the ADREFP1 bit of A/D converter mode register 2 (ADM2) to 0 and the ADREFP0 bit to 1.

The analog signals input to ANI0 to ANI7 and ANI16 to ANI19 are converted to digital signals based on the voltage applied between AVREFP and the - side reference voltage (AVREFM/VSS).

In addition to AVREFP, it is possible to select VDD or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(11) AVREFM pin

This pin inputs an external reference voltage (AVREFM). If using AVREFM as the - side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AVREFM, it is possible to select VSS as the - side reference voltage of the A/D converter.

12.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14)
- Port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)

12.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 2 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PER0	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read/written.

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 0, 2, 12 and 14 (PM0, PM2, PM12, PM14), port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14), and A/D port configuration register (ADPC)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 4, 6, 7

12.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 3 Format of A/D converter mode register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control	
0	Stops conversion operation [When read] Conversion stopped/standby status	
1	Enables conversion operation [When read] While in the software trigger mode: While in the hardware trigger wait mode:	 Conversion operation status A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Tables 12 - 3 to 12 - 6 A/D Conversion Time Selection**.

Note 2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, LV0, and ADCE bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

Caution 3. Do not change the ADCE and ADCS bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 12.7 A/D Converter Setup Flowchart.

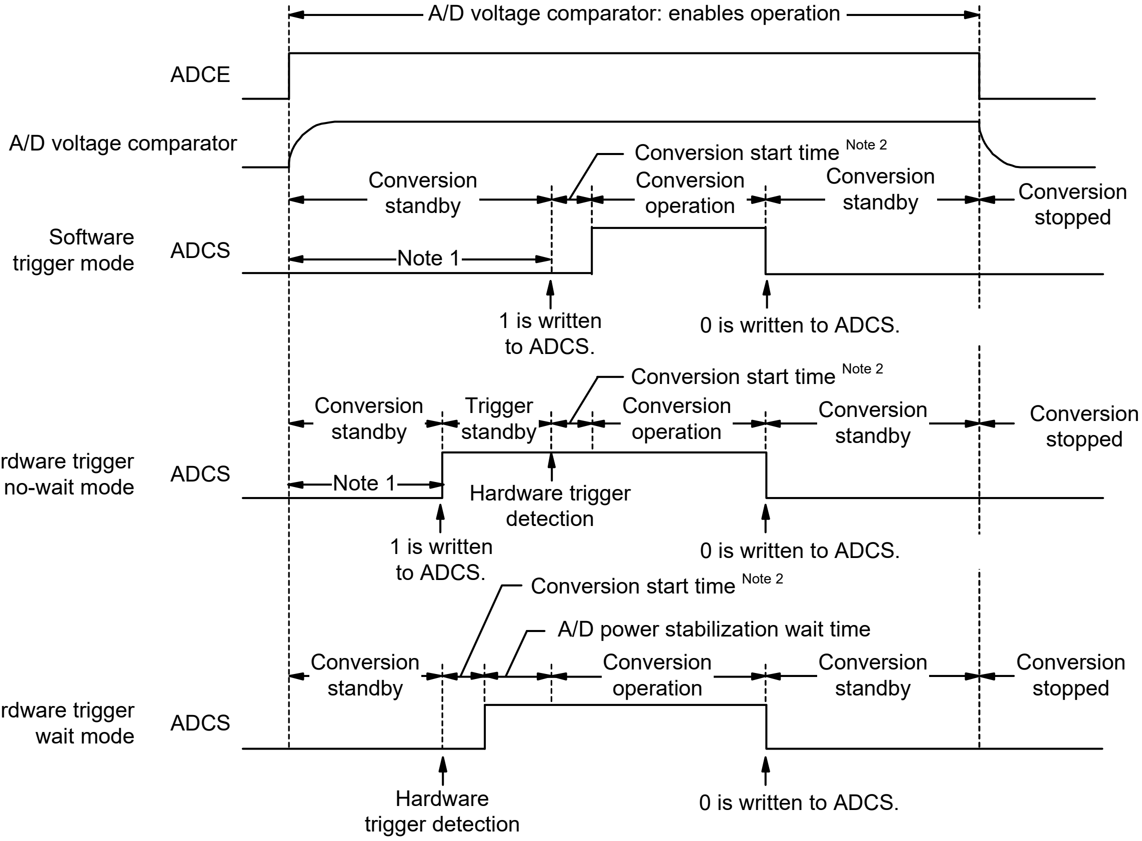
Table 12 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 12 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 12 - 4 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

Note 2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (fAD)	Conversion Start Time (Number of fCLK Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	fCLK/64	63	1
0	0	1	fCLK/32	31	
0	1	0	fCLK/16	15	
0	1	1	fCLK/8	7	
1	0	0	fCLK/6	5	
1	0	1	fCLK/5	4	
1	1	0	fCLK/4	3	
1	1	1	fCLK/2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

(**Cautions** and **Remark** are listed on the next page.)

Caution 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + Conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + Conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (1/4)

(1) When there is no A/D power supply stabilization wait time Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D converter mode register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution					
									2.7 V ≤ V_{DD} ≤ 5.5 V					
FR 2	FR 1	FR 0	LV 1	LV 0					f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	$f_{CLK}/64$	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				$f_{CLK}/32$		608/f_{CLK}					38 μs	25.33 μs
0	1	0				$f_{CLK}/16$		304/f_{CLK}				38 μs	19 μs	12.67 μs
0	1	1				$f_{CLK}/8$		152/f_{CLK}			38 μs	19 μs	9.5 μs	6.33 μs
1	0	0				$f_{CLK}/6$		114/f_{CLK}			28.5 μs	14.25 μs	7.125 μs	4.75 μs
1	0	1				$f_{CLK}/5$		95/f_{CLK}			23.75 μs	11.875 μs	5.938 μs	3.96 μs
1	1	0				$f_{CLK}/4$		76/f_{CLK}		38 μs	19 μs	9.5 μs	4.75 μs	3.17 μs
1	1	1				$f_{CLK}/2$		38/f_{CLK}	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	$f_{CLK}/64$	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				$f_{CLK}/32$		544/f_{CLK}					34 μs	22.67 μs
0	1	0				$f_{CLK}/16$		272/f_{CLK}				34 μs	17 μs	11.33 μs
0	1	1				$f_{CLK}/8$		136/f_{CLK}			34 μs	17 μs	8.5 μs	5.67 μs
1	0	0				$f_{CLK}/6$		102/f_{CLK}			25.5 μs	12.75 μs	6.375 μs	4.25 μs
1	0	1				$f_{CLK}/5$		85/f_{CLK}			21.25 μs	10.625 μs	5.3125 μs	3.54 μs
1	1	0				$f_{CLK}/4$		68/f_{CLK}		34 μs	17 μs	8.5 μs	4.25 μs	2.83 μs
1	1	1				$f_{CLK}/2$		34/f_{CLK}	34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 29.7.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 4 A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D converter mode register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution					
									2.7 V ≤ V_{DD} ≤ 5.5 V					
FR 2	FR 1	FR 0	LV 1	LV 0					f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low-voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32		608/f_{CLK}					38 µs	25.33 µs
0	1	0				f_{CLK}/16		304/f_{CLK}				38 µs	19 µs	Setting prohibited
0	1	1				f_{CLK}/8		152/f_{CLK}			38 µs	19 µs	Setting prohibited	
1	0	0				f_{CLK}/6		114/f_{CLK}			28.5 µs	Setting prohibited		
1	0	1				f_{CLK}/5		95/f_{CLK}			23.75 µs			
1	1	0				f_{CLK}/4		76/f_{CLK}		38 µs	19 µs			
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs	19 µs	Setting prohibited			
0	0	0	1	1	Low-voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32		544/f_{CLK}					34 µs	22.67 µs
0	1	0				f_{CLK}/16		272/f_{CLK}				34 µs	17 µs	Setting prohibited
0	1	1				f_{CLK}/8		136/f_{CLK}			34 µs	17 µs	Setting prohibited	
1	0	0				f_{CLK}/6		102/f_{CLK}			25.5 µs	Setting prohibited		
1	0	1				f_{CLK}/5		85/f_{CLK}			21.25 µs			
1	1	0				f_{CLK}/4		68/f_{CLK}		34 µs	17 µs			
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs	17 µs	Setting prohibited			

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 29.7.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 5 A/D Conversion Time Selection (3/4)

(3) When there is A/D power supply stabilization wait time Normal mode 1, 2 (hardware trigger wait mode Note 1)

A/D converter mode register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Stabilization Wait Clock	Number of Conversion Clock Note 4	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution					
										2.7 V ≤ VDD ≤ 5.5 V					
FR2	FR1	FR0	LV1	LV0						fCLK = 1 MHz	fCLK = 2 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	0	0	Normal 1	fCLK/64	8 fAD	19 fAD (number of sampling clock: 7 fAD)	1728/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			864/fCLK					54 µs	36 µs
0	1	0				fCLK/16			432/fCLK				54 µs	27 µs	18 µs
0	1	1				fCLK/8			216/fCLK			54 µs	27 µs	13.5 µs	9 µs
1	0	0				fCLK/6			162/fCLK			40.5 µs	20.25 µs	10.125 µs	6.75 µs
1	0	1				fCLK/5			135/fCLK			33.75 µs	16.875 µs	8.4375 µs	5.63 µs
1	1	0				fCLK/4			108/fCLK		54 µs	27µs	13.5 µs	6.75 µs	4.5 µs Note 2
1	1	1				fCLK/2			54/fCLK	54 µs	27µs	13.5 µs	6.75 µs	3.375 µs Notes 2, 3	Setting prohibited
0	0	0	0	1	Normal 2	fCLK/64	8 fAD	17 fAD (number of sampling clock: 5 fAD)	1600/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			800/fCLK					50 µs	33.33 µs
0	1	0				fCLK/16			400/fCLK				50 µs	25 µs	16.67 µs
0	1	1				fCLK/8			200/fCLK			50 µs	25 µs	12.5 µs	8.33 µs
1	0	0				fCLK/6			150/fCLK			37.5 µs	18.75 µs	9.375 µs	6.25 µs
1	0	1				fCLK/5			125/fCLK			31.25 µs	15.625 µs	7.8125 µs	5.21 µs
1	1	0				fCLK/4			100/fCLK		50 µs	25 µs	12.5 µs	6.25 µs	4.17 µs Note 2
1	1	1				fCLK/2			50/fCLK	50 µs	25 µs	12.5 µs	6.25 µs	3.125 µs Notes 2, 3	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 3**).

Note 2. Setting is prohibited when VDD < 3.6 V.

Note 3. This value is prohibited when using the temperature sensor.

Note 4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 29.7.1 A/D converter characteristics.

Note that the conversion time (tCONV) does not include the A/D power supply stabilization wait time.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark fCLK: CPU/peripheral hardware clock frequency

Table 12 - 6 A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time Low-voltage mode 1, 2 Note 1 (hardware trigger wait mode Note 2)

A/D converter mode register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 3	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution					
										2.7 V ≤ VDD ≤ 5.5 V					
FR 2	FR 1	FR 0	LV 1	LV 0						fCLK = 1 MHz	fCLK = 2 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	1	0	Low-voltage 1	fCLK/64	2 fAD	19 fAD (number of sampling clock: 7 fAD)	1344/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			672/fCLK					42 μs	28 μs
0	1	0				fCLK/16			336/fCLK				42 μs	27 μs	Setting prohibited
0	1	1				fCLK/8			168/fCLK			42 μs	21 μs	Setting prohibited	
1	0	0				fCLK/6			126/fCLK			31.5 μs	Setting prohibited		
1	0	1				fCLK/5			105/fCLK			26.25 μs			
1	1	0				fCLK/4			84/fCLK		42 μs	21 μs			
1	1	1				fCLK/2			42/fCLK	42 μs	21 μs	Setting prohibited			
0	0	0	1	1	Low-voltage 2	fCLK/64	2 fAD	17 fAD (number of sampling clock: 5 fAD)	1216/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			608/fCLK					38 μs	25.33 μs
0	1	0				fCLK/16			304/fCLK				38 μs	19 μs	Setting prohibited
0	1	1				fCLK/8			152/fCLK			38 μs	19 μs	Setting prohibited	
1	0	0				fCLK/6			114/fCLK			28.5 μs	Setting prohibited		
1	0	1				fCLK/5			96/fCLK			23.75 μs			
1	1	0				fCLK/4			76/fCLK		38 μs	19 μs			
1	1	1				fCLK/2			38/fCLK	38 μs	19 μs	Setting prohibited			

Note 1. This mode is prohibited when using the temperature sensor.

Note 2. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 4**).

Note 3. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 29.7.1 A/D converter characteristics.

Note that the conversion time (tCONV) does not include the A/D power supply stabilization wait time.

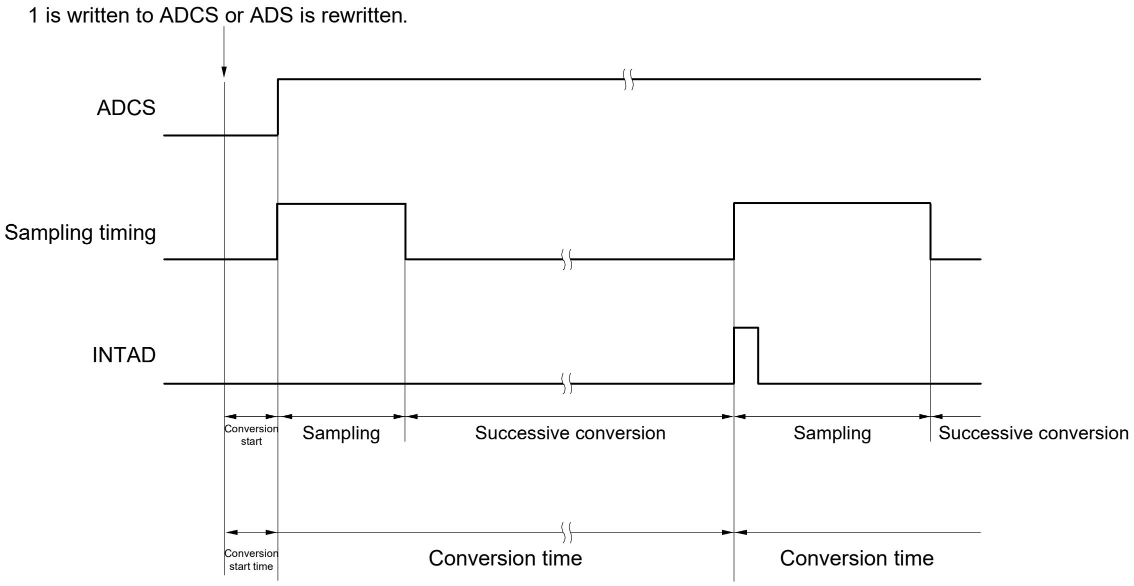
Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark fCLK: CPU/peripheral hardware clock frequency

Figure 12 - 5 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



12.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 6 Format of A/D converter mode register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	0	Software trigger mode
0	1	
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	Event signal selected by ELC
1	1	12-bit interval timer interrupt signal (INTIT)
Other than above		Setting prohibited

Caution 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE mode, input of the next INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTIT is input.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

12.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 7 Format of A/D converter mode register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V) Note
1	1	Setting prohibited
• When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures. (1) Set ADCE = 0 (2) Change the values of ADREFP1 and ADREFP0 (3) Reference voltage stabilization wait time (A) (4) Set ADCE = 1 (5) Reference voltage stabilization wait time (B) When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs. When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs. • When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output and internal reference voltage output. Be sure to perform A/D conversion while ADISS = 0.		

ADREFM	Selection of the - side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 12 - 9 shows the generation range of the interrupt signal (INTAD) for AREA1 to AREA3.	

Note Operation is possible only in HS (high-speed main) mode.

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. When entering STOP mode, do not set ADREFP1 to 1. When selecting internal reference voltage (ADREFP1, ADREFP0 = 1, 0), the current value of A/D converter reference voltage current (I_{ADREF}) shown in 29.4.2 Supply current characteristics is added.

Caution 3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 12 - 8 Format of A/D converter mode register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode Note + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 f_{CLK} clock”
- Even when using the SNOOZE function, set AWC to 0 in normal operating mode and set AWC to 1 immediately before shifting to STOP mode.

After returning to normal operating mode from STOP mode, be sure to change AWC to 0.

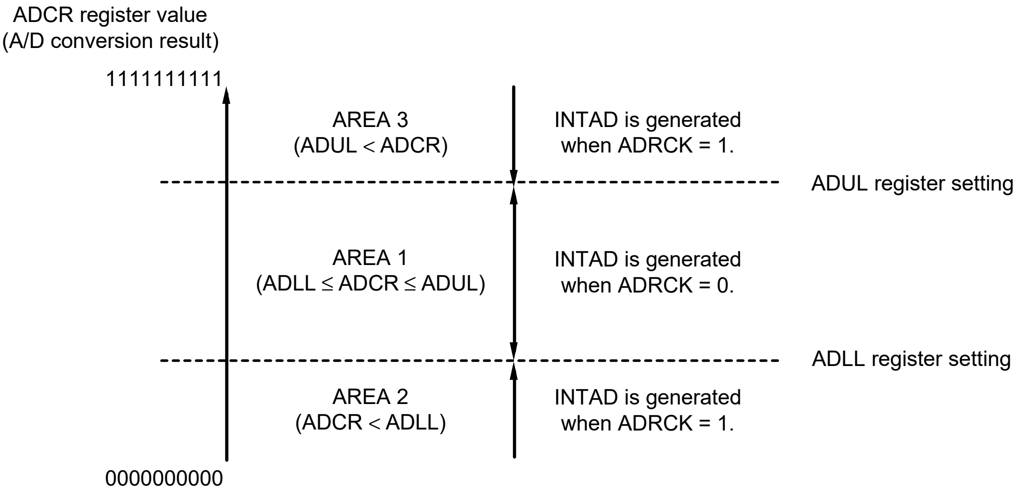
While AWC = 1, A/D conversion will not start normally regardless of subsequent SNOOZE mode or normal operating mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to “Transition time from STOP mode to SNOOZE mode” in **18.3.3 SNOOZE mode**.

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS and ADCE bits of A/D converter mode register 0 (ADM0) being 0).

Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

12.3.5 10-bit A/D conversion result register (ADCR)

The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH [Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12 - 9**), the result is not stored.

Figure 12 - 10 Format of 10-bit A/D conversion result register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (bits 7 and 6 of the ADCR register).

Caution 2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

12.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored [Note].

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12 - 9**), the result is not stored.

Figure 12 - 11 Format of 8-bit A/D conversion result register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), Analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

12.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 12 Format of Analog input channel specification register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

- Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	0	1	0	0	—	PGAO (Programmable gain amplifier output)
1	0	0	0	0	0	—	Temperature sensor output voltage Note
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note Operation is possible only in HS (high-speed main) mode.

(**Cautions** are listed on the next page.)

Figure 12 - 13 Format of Analog input channel specification register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

- Scan mode (ADMD = 1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	1	0	0	0	0	ANI16	ANI17	ANI18	ANI19
0	1	0	0	0	1	ANI17	ANI18	ANI19	PGAO
Other than the above						Setting prohibited			

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Set a channel to be set the analog input by ADPC and PMCx registers in the input mode by using port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14).

Caution 3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

Caution 4. Do not set the pin that is set by Port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14) as digital I/O by the ADS register.

Caution 5. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 6. If using AVREFP as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 7. If using AVREFM as the - side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

Caution 9. Do not set the ADISS bit to 1 when shifting to STOP mode when the ADISS bit is set to 1, the A/D converter reference voltage current (IADREF) indicated in 29.4.2 Supply current characteristics will be added.

12.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9**).

The ADUL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Caution 1.When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

Figure 12 - 14 Format of Conversion result comparison upper limit setting register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

12.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9**).

The ADLL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 15 Format of Conversion result comparison lower limit setting register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

12.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the converter, an analog input channel (ANIxx), the temperature sensor output voltage, or the internal reference voltage (1.45 V) as the target for A/D conversion.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the - side reference voltage as the target for conversion.
- For full-scale measurement, select the + side reference voltage as the target for conversion.

Figure 12 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage Note/internal reference voltage (1.45 V) Note (This is specified using the analog input channel specification register (ADS).)
1	0	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

Note The temperature sensor output voltage and internal reference voltage (1.45 V) can be selected only in the HS (high-speed main) mode.

12.3.11 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.6 Port mode control registers 0, 12, 14 (PMCxx)**, and **4.3.7 A/D port configuration register (ADPC)**.

When using the ANI0 to ANI7 pins for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1 and select analog input through the A/D port configuration register (ADPC).

When using the ANI16 to ANI19 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

12.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched Note 1.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 Note 2.

To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 12 - 9**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

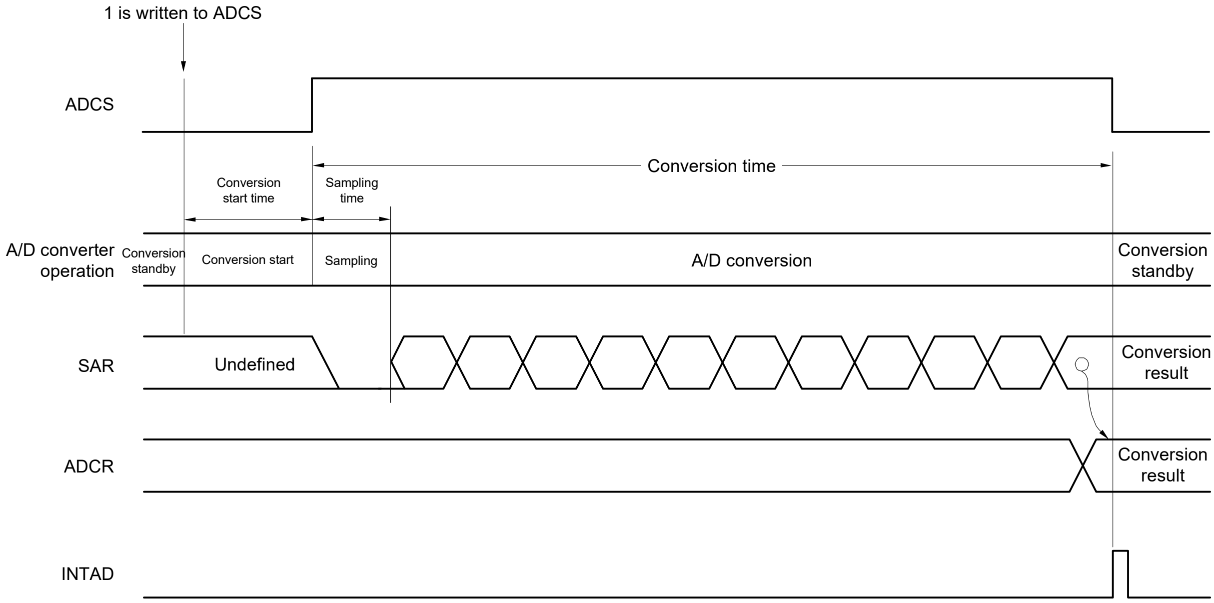
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 12 - 17 Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7, ANI16 to ANI19) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\text{SAR} = \text{INT}\left(\frac{V_{\text{AIN}}}{AV_{\text{REF}}} \times 1024 + 0.5\right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

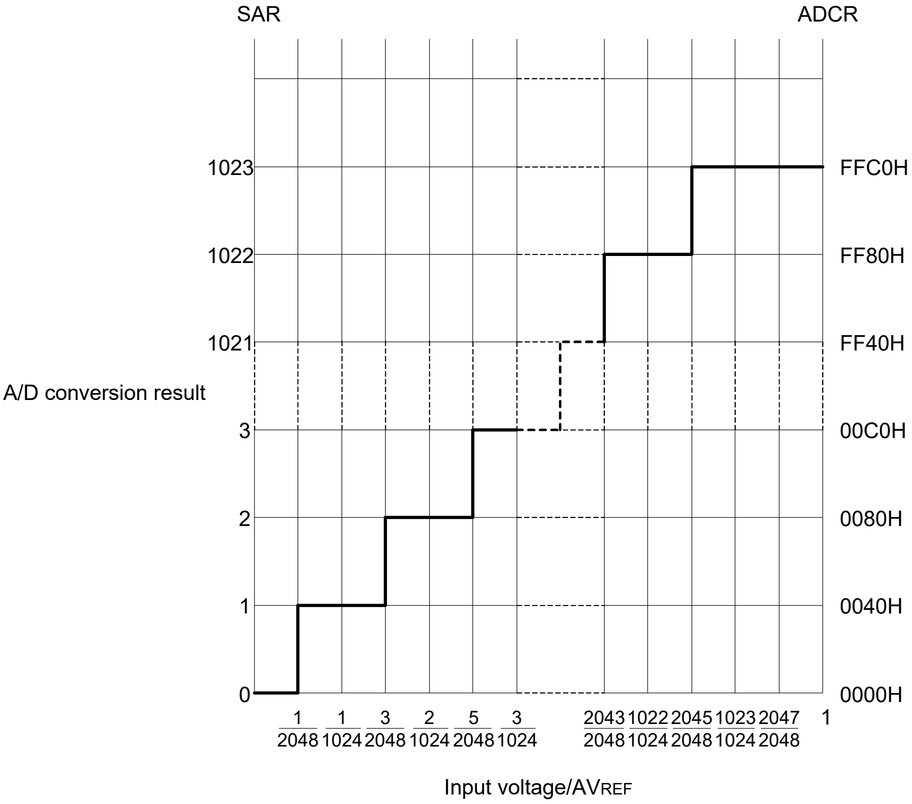
or

$$\left(\frac{\text{ADCR}}{64} - 0.5\right) \times \frac{AV_{\text{REF}}}{1024} \leq V_{\text{AIN}} < \left(\frac{\text{ADCR}}{64} + 0.5\right) \times \frac{AV_{\text{REF}}}{1024}$$

where,	INT( ):	Function which returns integer part of value in parentheses
	V_{AIN}:	Analog input voltage
	AV_{REF}:	AV_{REF} pin voltage
	ADCR:	A/D conversion result register (ADCR) value
	SAR:	Successive approximation register

Figure 12 - 18 shows the Relationship between Analog Input Voltage and A/D Conversion Result.

Figure 12 - 18 Relationship between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

12.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **12.7 A/D Converter Setup Flowchart**.

12.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

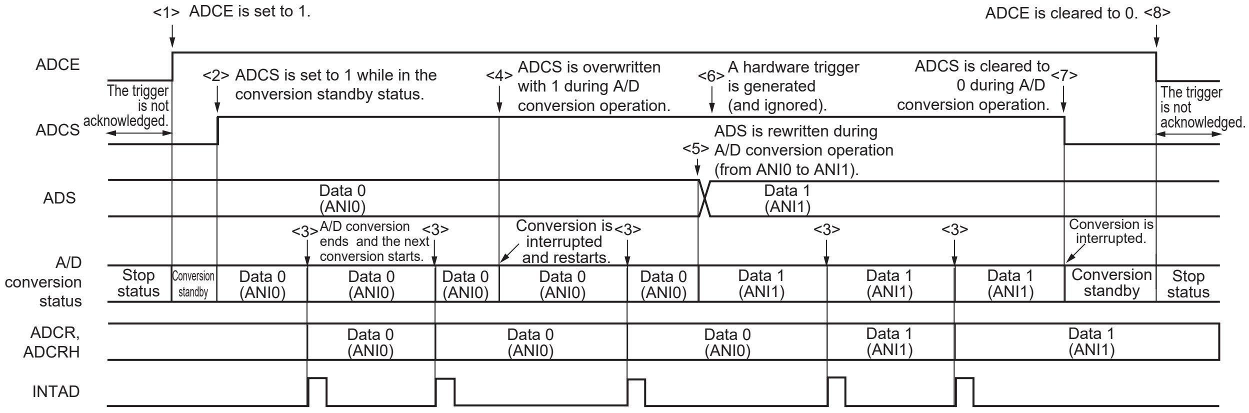
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

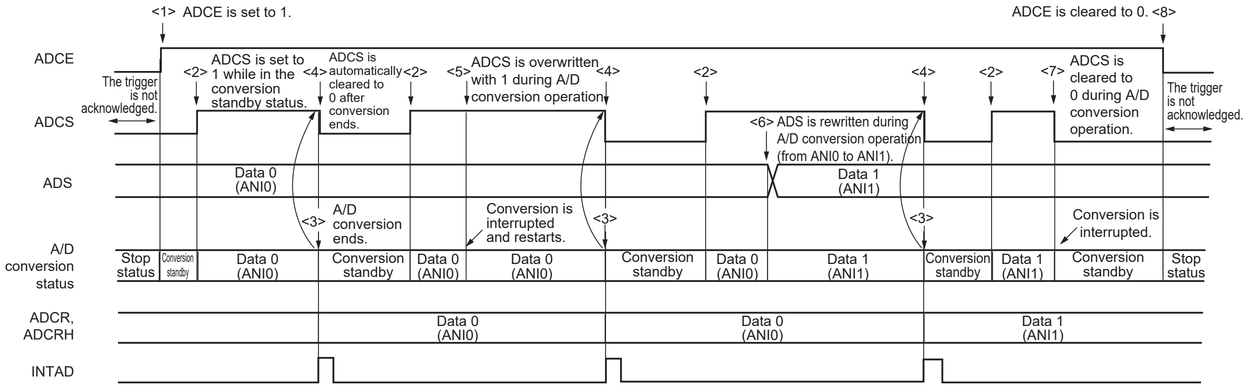
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 12 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

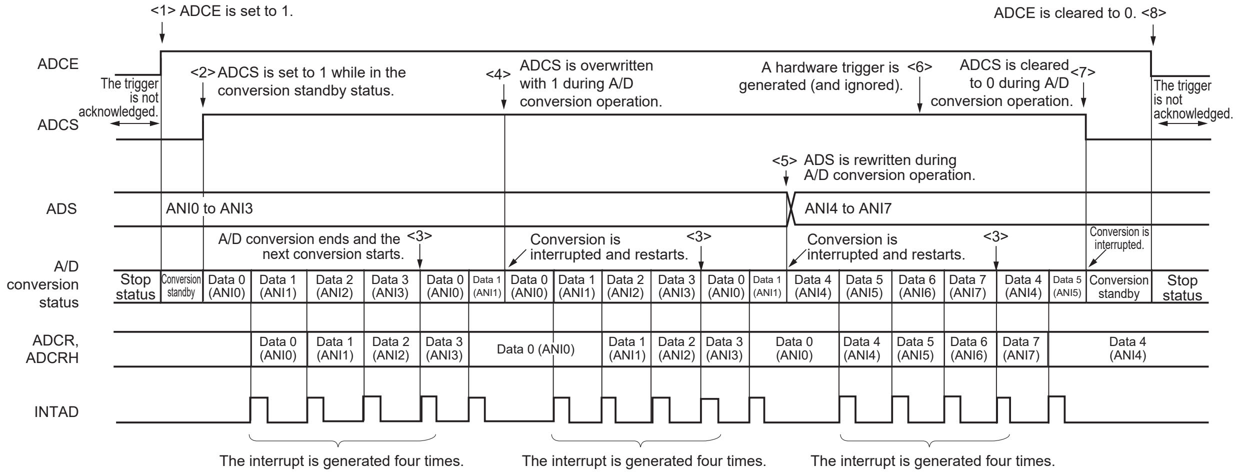
<5> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

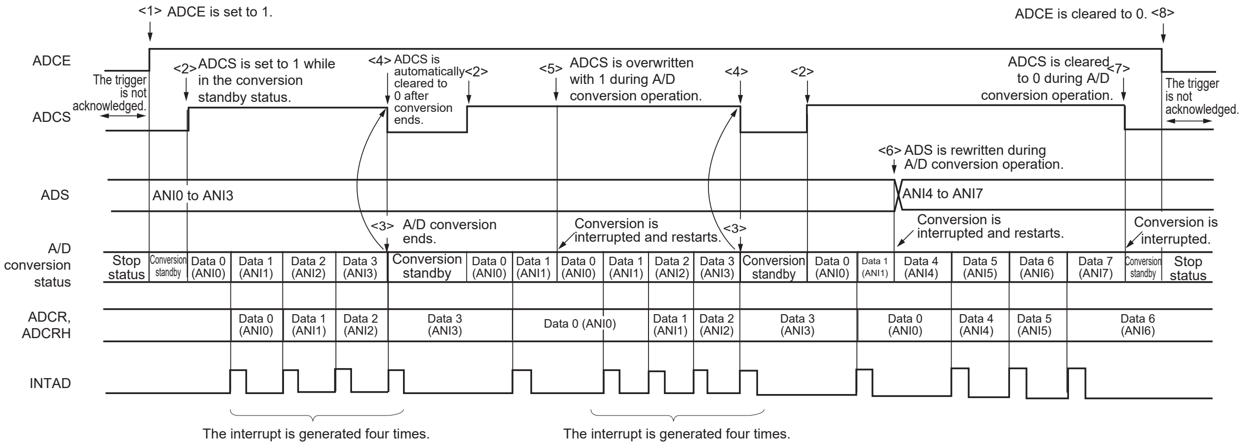
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 12 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

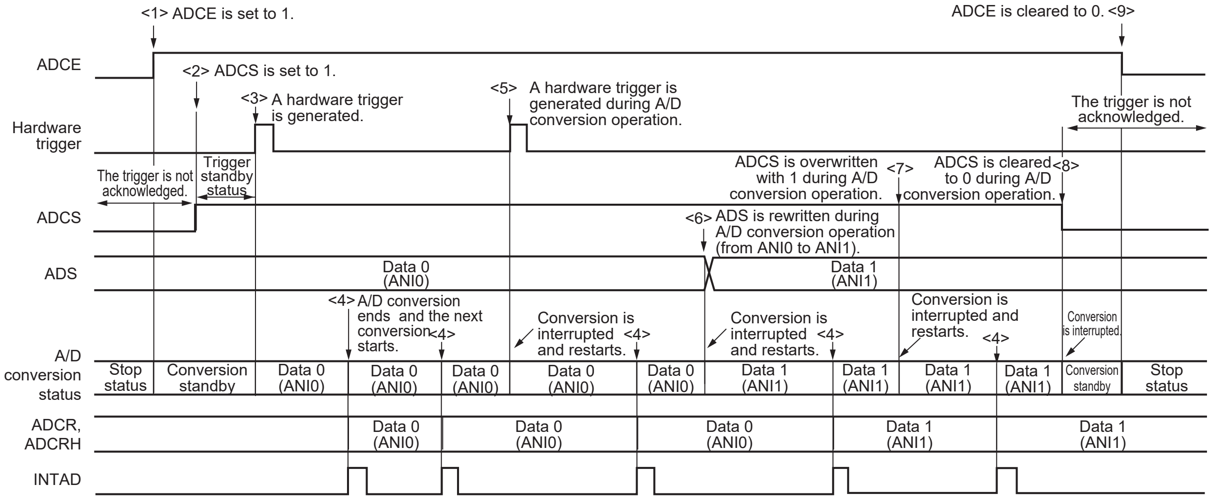
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

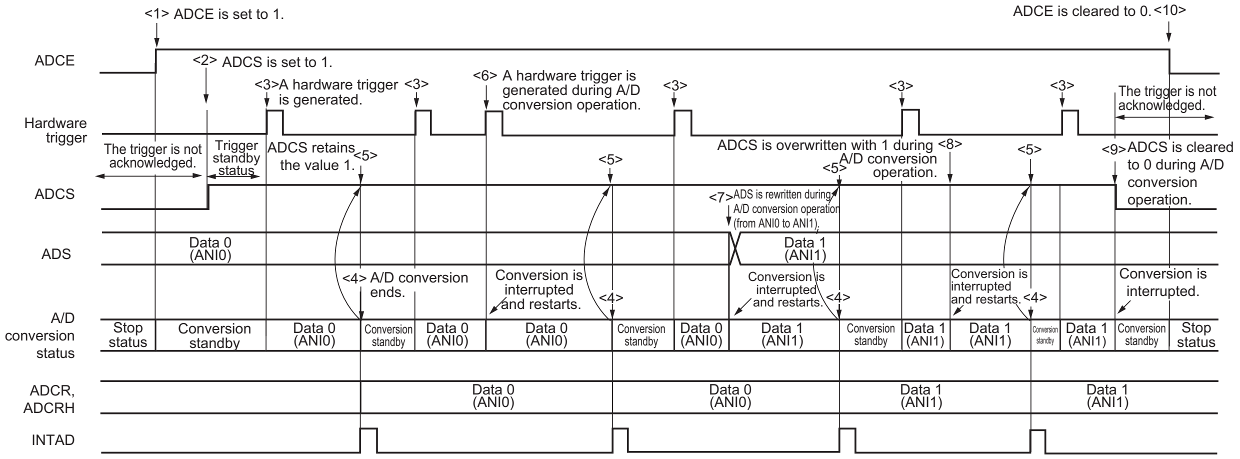
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

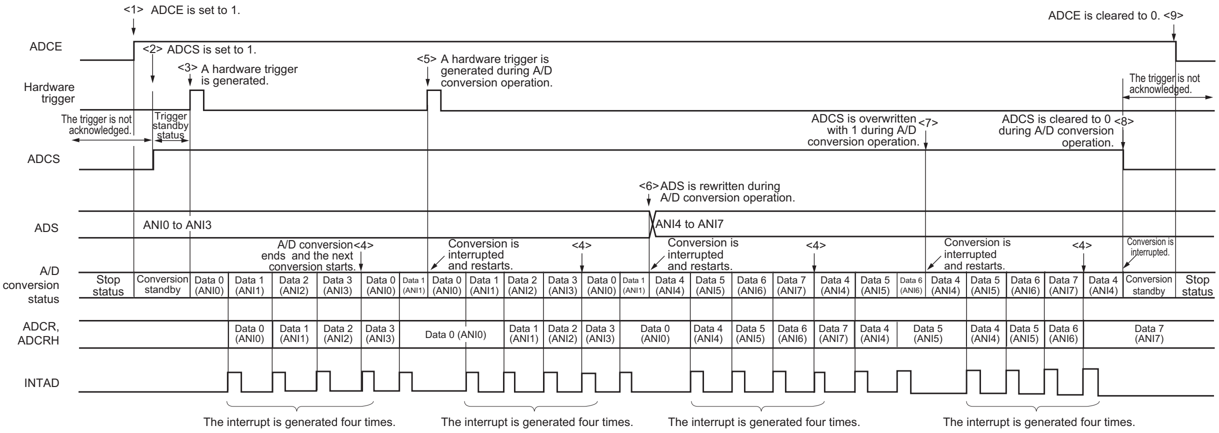
<6> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

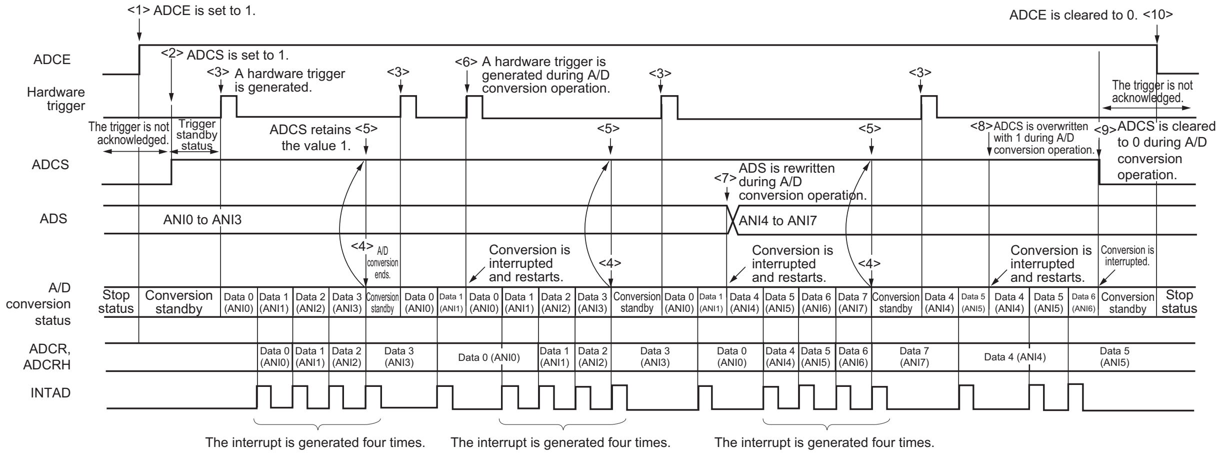
<7> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 26 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

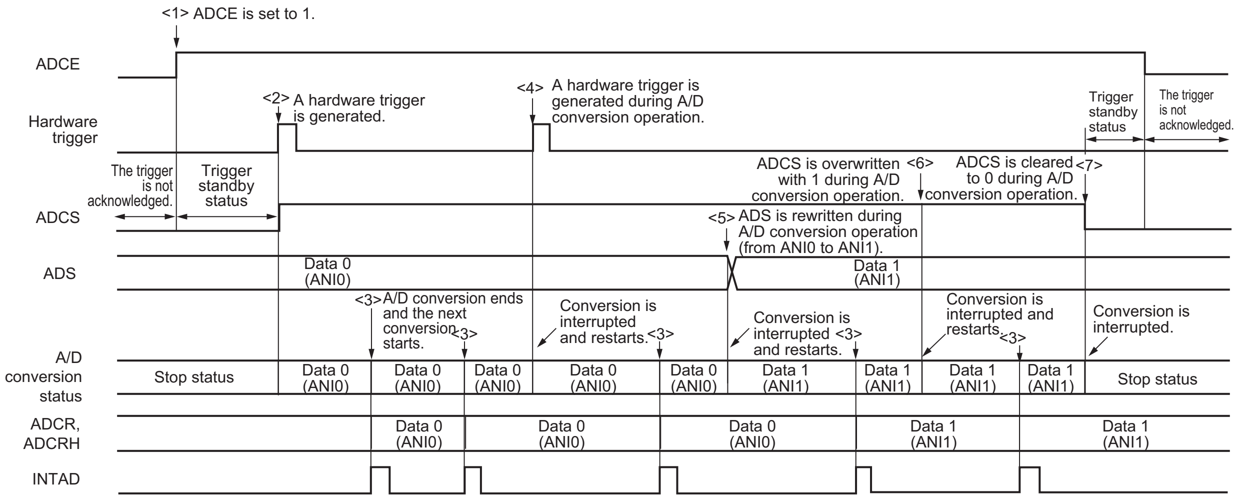
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

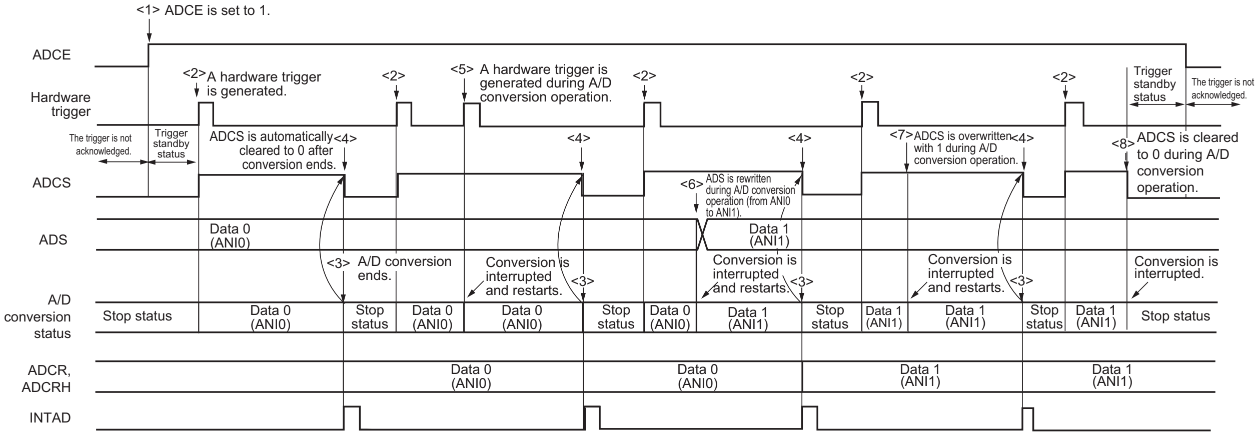
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

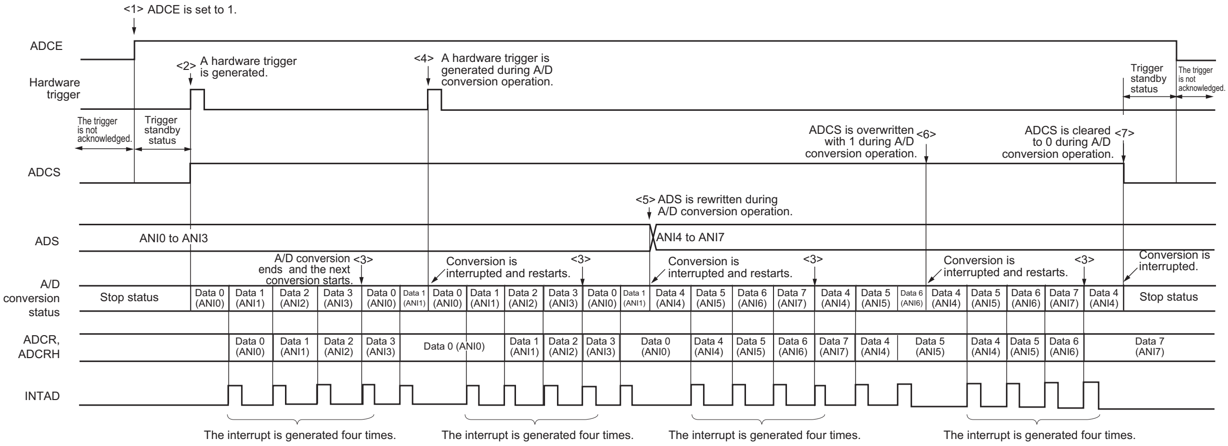
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

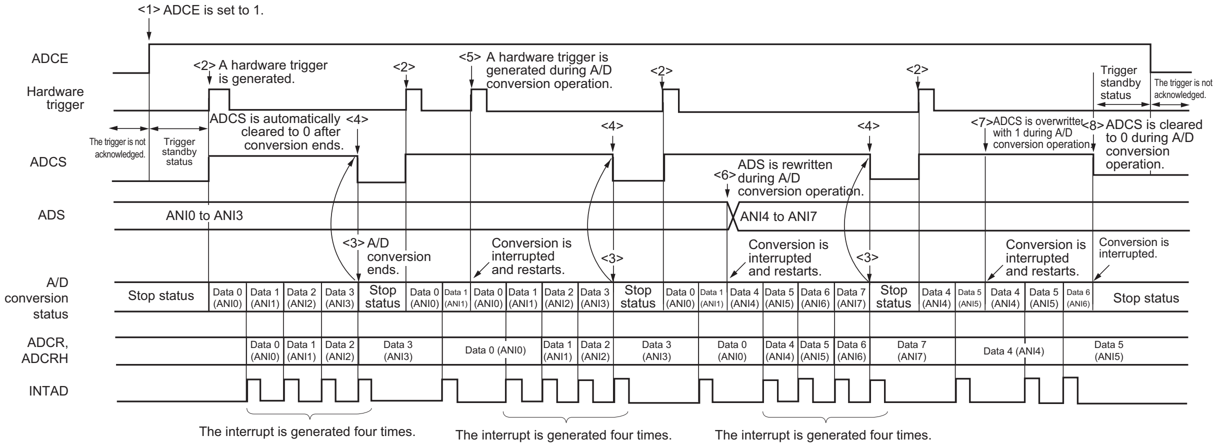
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

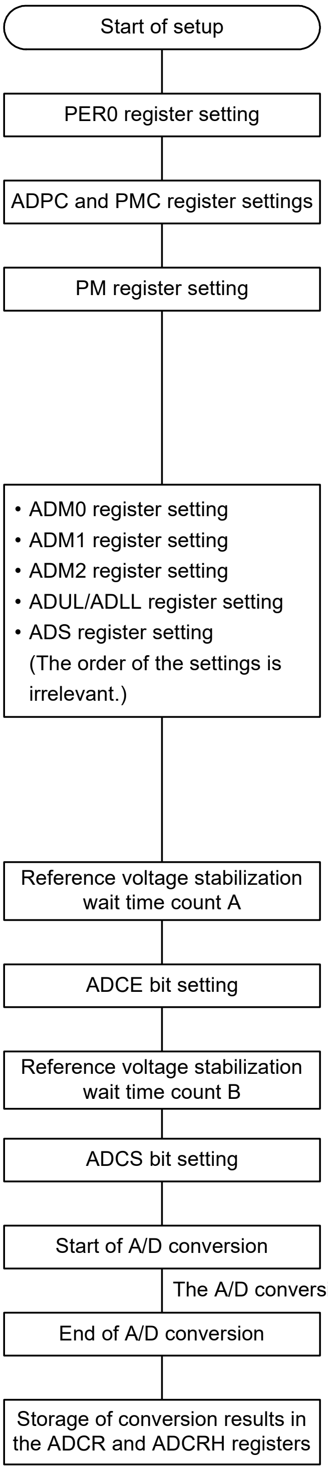


12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

12.7.1 Setting up software trigger mode

Figure 12 - 31 Setting Up Software Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits:
  These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits:
  These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits:
  These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

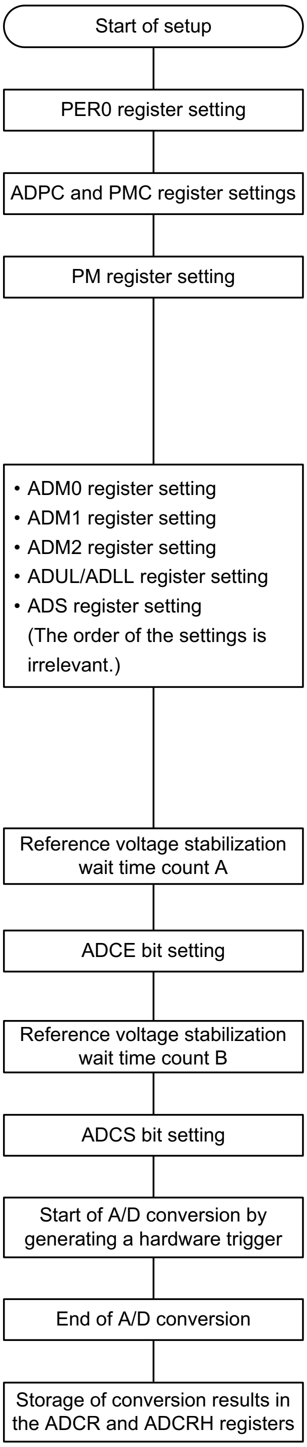
The A/D conversion end interrupt (INTAD) is generated. Note

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

12.7.2 Setting up hardware trigger no-wait mode

Figure 12 - 32 Setting Up Hardware Trigger No-Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits:
  These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits:
  These are used to specify the hardware trigger no-wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits:
  These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and the system enters the hardware trigger standby status.
Hardware trigger standby status

The A/D conversion operations are performed.

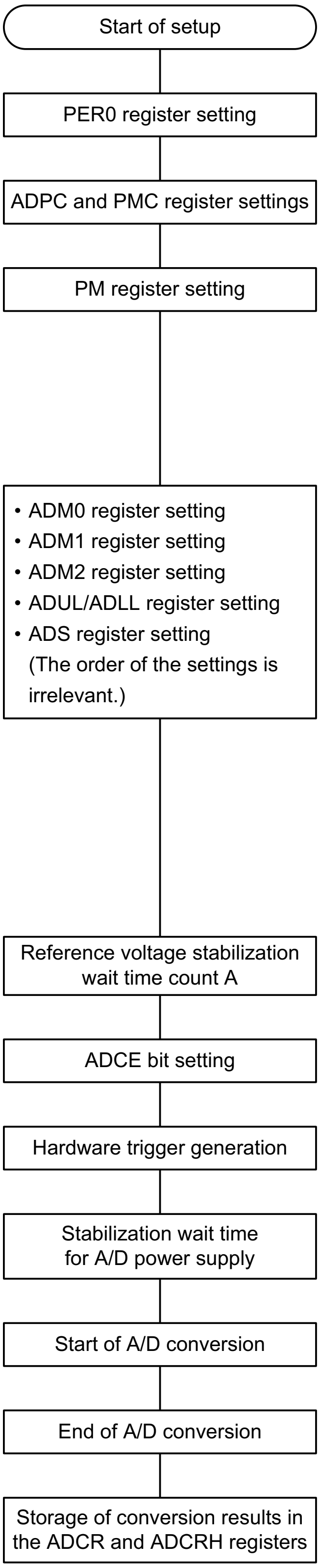
The A/D conversion end interrupt (INTAD) is generated. Note

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

12.7.3 Setting up hardware trigger wait mode

Figure 12 - 33 Setting Up Hardware Trigger Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits:
  These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits:
  These are used to specify the hardware trigger wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits:
  These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  AWC bit: This is used to set up the SNOOZE mode function.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The system automatically counts up to the stabilization wait time for A/D power supply.

After counting up to the reference voltage stabilization wait time ends, A/D conversion starts
The A/D conversion operations are performed.

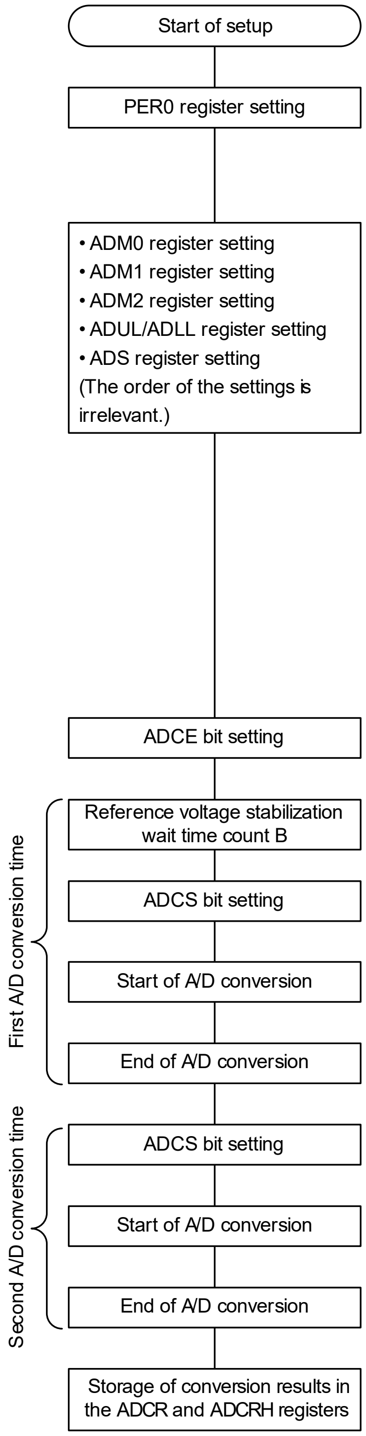
The A/D conversion end interrupt (INTAD) is generated. Note

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)

Figure 12 - 34 Setup when temperature sensor output voltage/internal reference voltage is selected



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADISS and ADS4 to ADS0 bits: These are used to select temperature sensor 0 output or internal reference voltage output.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion end interrupt (INTAD) will be generated.
After ADISS is set (1), the initial conversion result cannot be used.

The ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion end interrupt (INTAD) is generated. Note

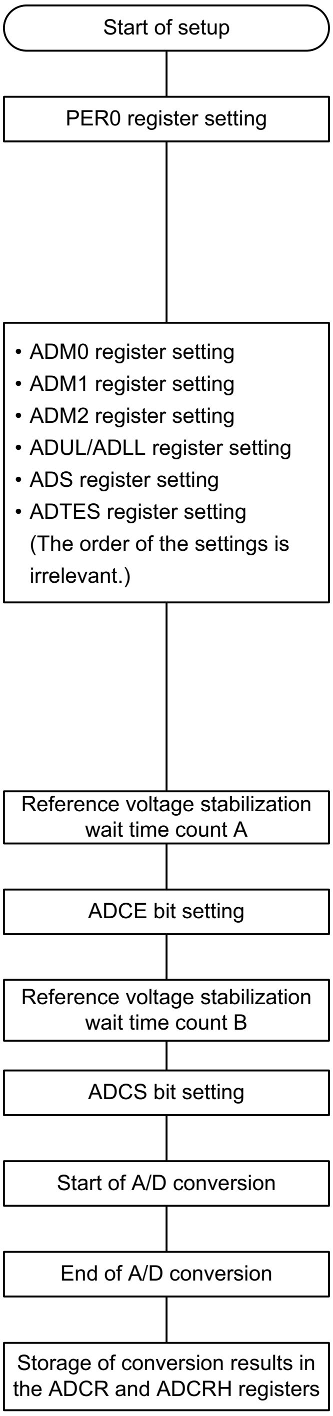
The conversion results are stored in the ADCR and ADCRH registers

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

Caution This setting can be used only in HS (high-speed main) mode.

12.7.5 Setting up test mode

Figure 12 - 35 Setting Up Test Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: This is used to specify the one-shot conversion mode.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select for the reference voltage source.
  ADRCK bit: This is used to set the range for the A/D conversion result comparison value generated by the interrupt signal to AREA2.
  ADTYP bit: This is used to specify 10-bit resolution.
- ADUL/ADLL register
  These set ADUL to FFH and ADLL to 00H (initial values).
- ADS register
  ADS4 to ADS0 bits: These are used to set to ANI0.
- ADTES register
  ADTES1, ADTES0 bits: AVREFM/AVREFP

The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated. Note

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

Caution For the procedure for testing the A/D converter, see 22.10 A/D Test Function.

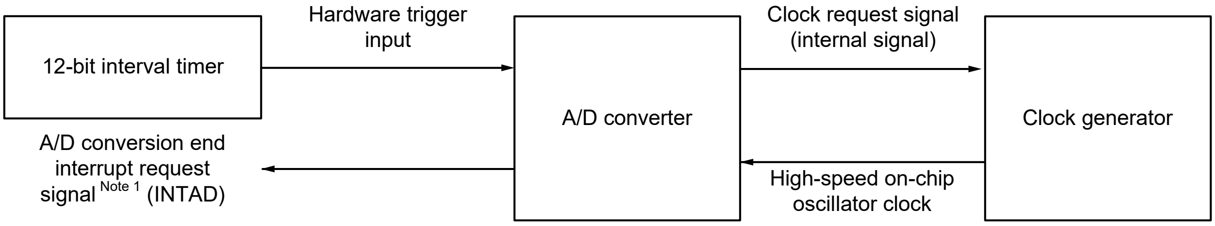
12.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:
- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for fCLK.

Figure 12 - 36 Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. (For details about these settings, see **12.7.3 Setting up hardware trigger wait mode** Note 2.) At this time, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.
If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.
The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

Note 2. Be sure to set the ADM1 register to E1H, E2H or E3H.

Remark The hardware trigger is event selected by ELC or INTIT.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

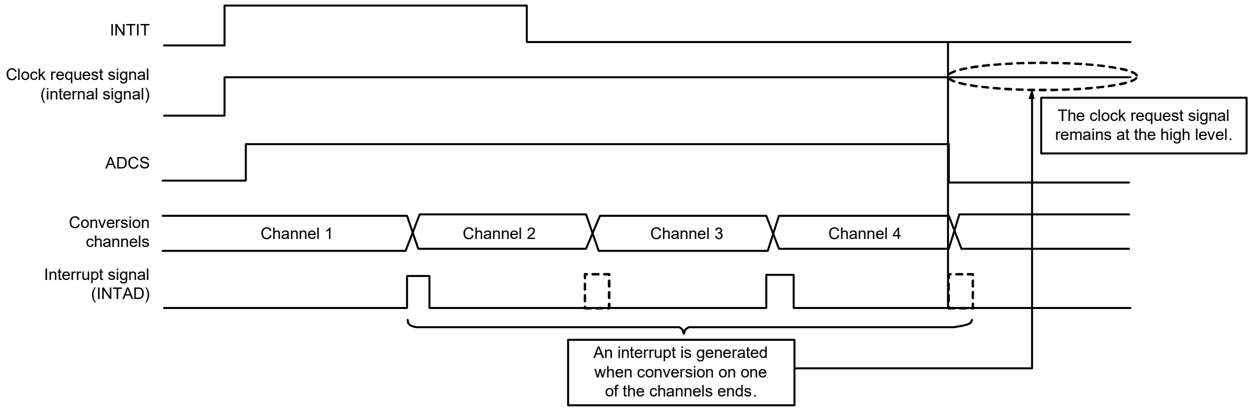
- While in the select mode

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 12 - 37 Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 12 - 38 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

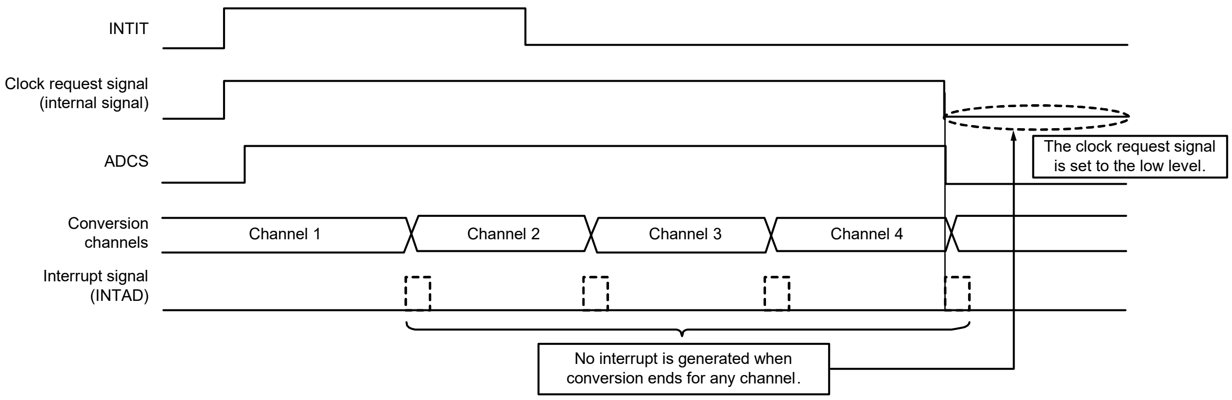
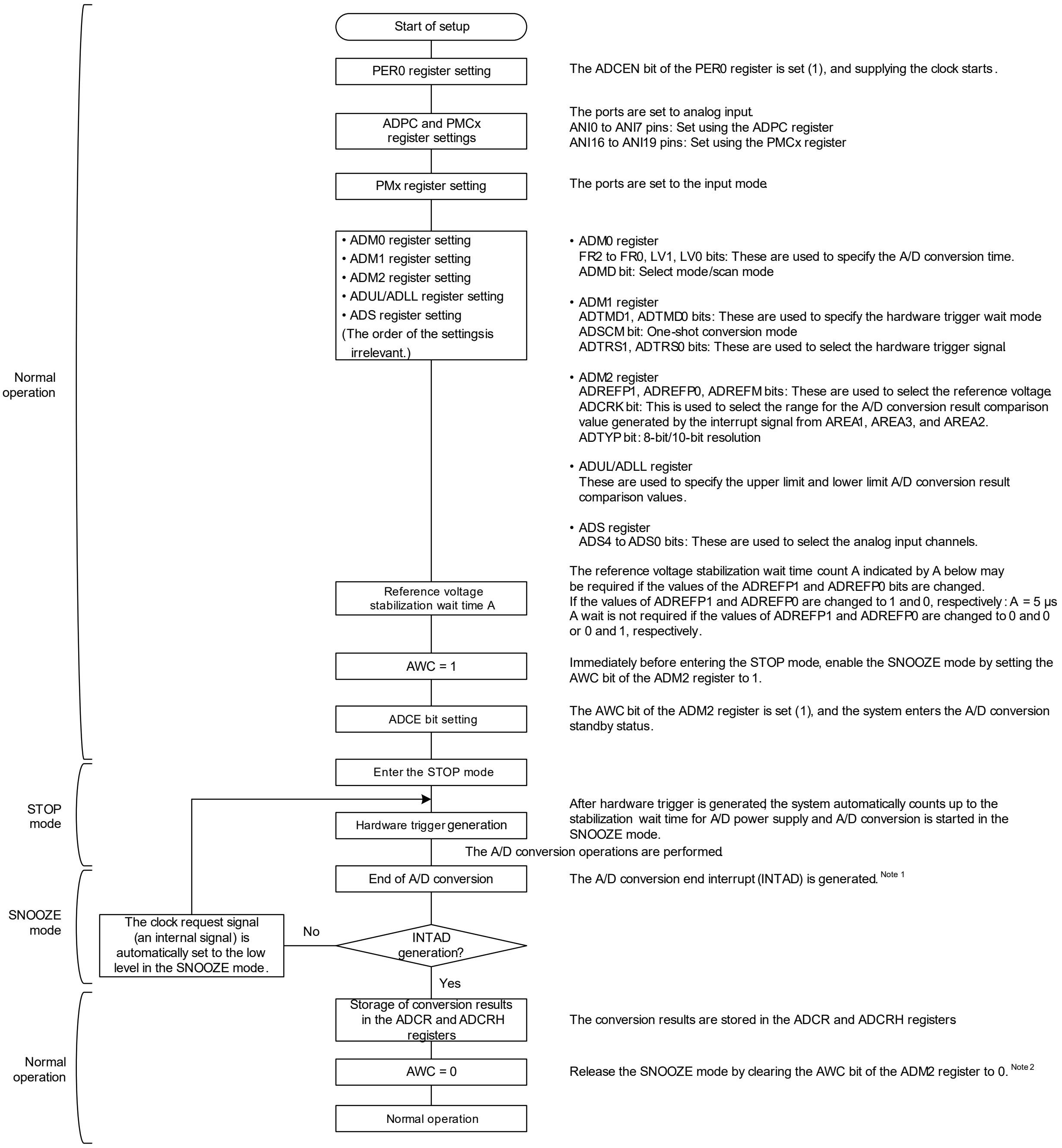


Figure 12 - 39 Flowchart for Setting up SNOOZE Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMCx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1, ADTMD0 bits: These are used to specify the hardware trigger wait mode
  ADSCM bit: One-shot conversion mode
  ADTRS1, ADTRS0 bits: These are used to select the hardware trigger signal
- ADM2 register
  ADREFP1, ADREFP0, ADREFM bits: These are used to select the reference voltage
  ADCRK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time count A indicated by A below may be required if the values of the ADREFP1 and ADREFP0 bits are changed.
If the values of ADREFP1 and ADREFP0 are changed to 1 and 0, respectively: A = 5 μs
A wait is not required if the values of ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1, respectively.

Immediately before entering the STOP mode, enable the SNOOZE mode by setting the AWC bit of the ADM2 register to 1.

The AWC bit of the ADM2 register is set (1), and the system enters the A/D conversion standby status.

After hardware trigger is generated the system automatically counts up to the stabilization wait time for A/D power supply and A/D conversion is started in the SNOOZE mode.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated. Note 1

The conversion results are stored in the ADCR and ADCRH registers

Release the SNOOZE mode by clearing the AWC bit of the ADM2 register to 0. Note 2

Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1 LSB (Least Significant Bit). The percentage of 1 LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1 LSB is as follows when the resolution is 10 bits.

1 LSB = $1/2^{10}$ = 1/1024
= 0.098%FSR

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2 LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2 LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12 - 40 Overall Error

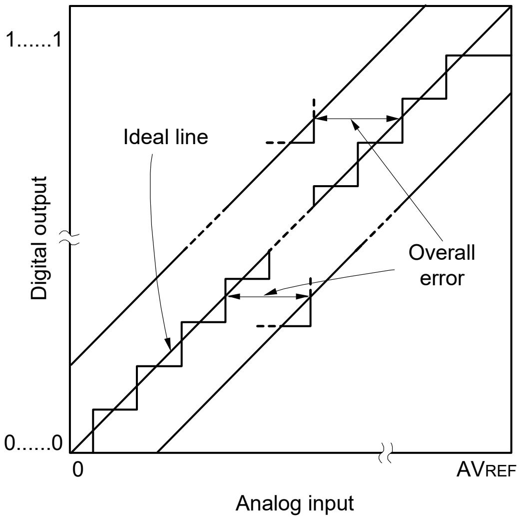
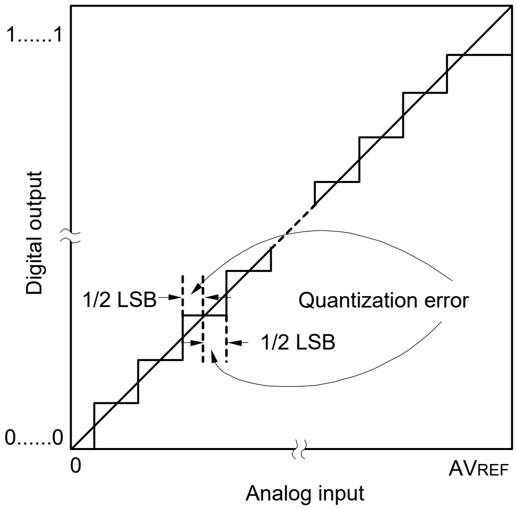


Figure 12 - 41 Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2 LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale - 3/2 LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1 LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 12 - 42 Zero-Scale Error

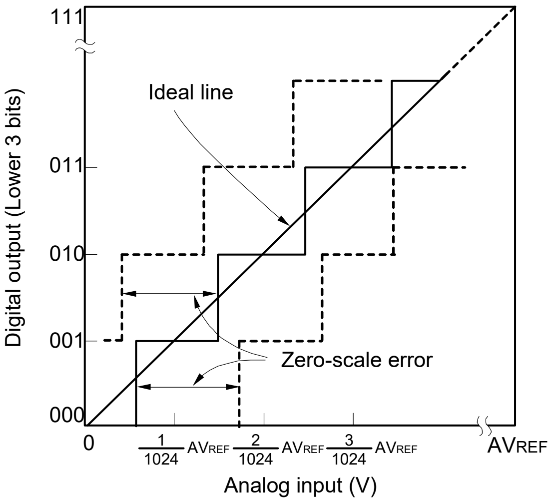


Figure 12 - 43 Full-Scale Error

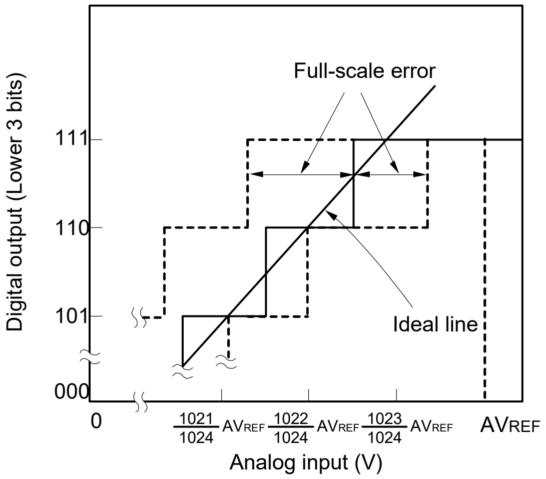


Figure 12 - 44 Integral Linearity Error

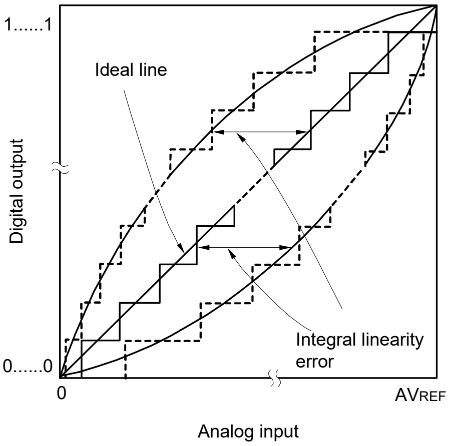
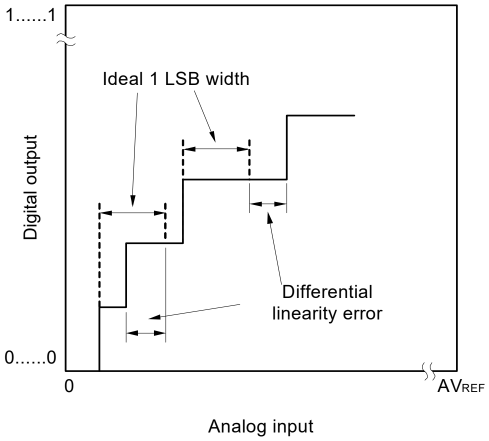


Figure 12 - 45 Differential Linearity Error



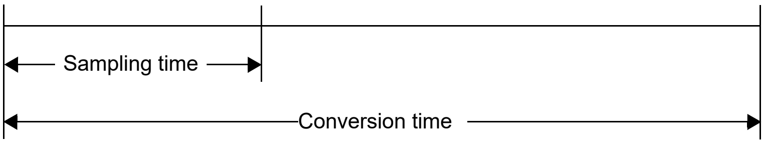
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI7 and ANI16 to ANI19 pins

Observe the rated range of the ANI0 to ANI7 and ANI16 to ANI19 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage source for the + side of the A/D converter, do not input exceeding internal reference voltage (1.45 V) to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage exceeding the internal reference voltage (1.45 V).

Caution Internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI7, and ANI16 to ANI19 pins.

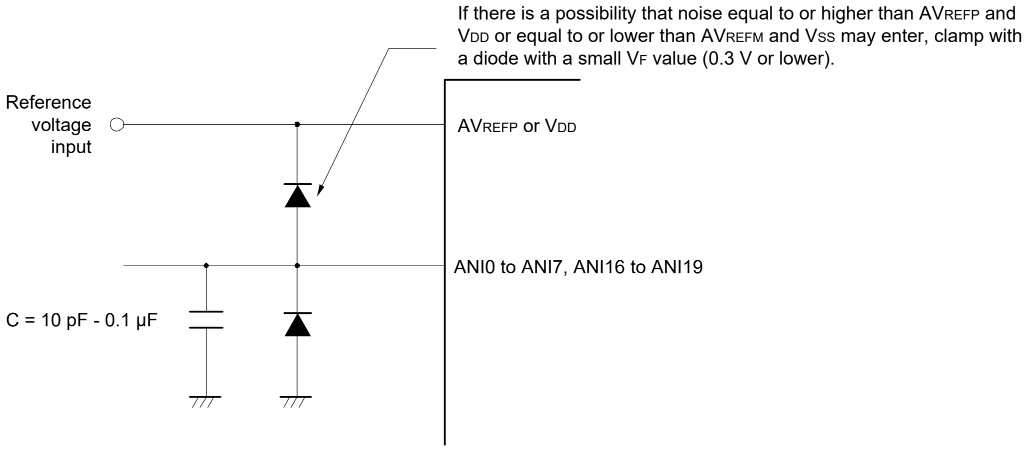
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 12 - 46 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 12 - 46 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI7, ANI16 to ANI19) are also used as input port pins (P20 to P23, P01, P00, P147, P120).
When A/D conversion is performed with any of the ANI0 to ANI7 and ANI16 to ANI19 pins selected, do not change to output value P20 to P23, P01, P00, P147, and P120 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

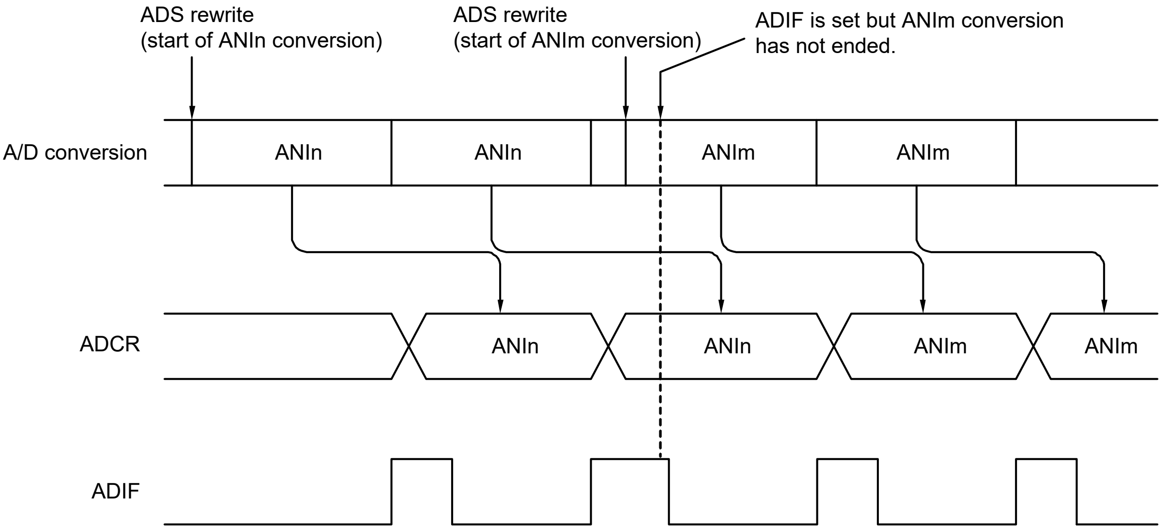
(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI7 and ANI16 to ANI19 to which the source is connected (see **Figure 12 - 46**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.
Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.
When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 12 - 47 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12 - 48 Internal Equivalent Circuit of ANIn Pin

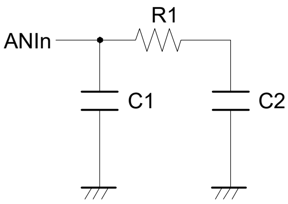


Table 12 - 7 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AVREFP, VDD	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ VDD ≤ 5.5 V	ANI0 to ANI7	14	8	2.5
	ANI16 to ANI19	18	8	7.0
2.7 V ≤ VDD < 3.6 V	ANI0 to ANI7	39	8	2.5
	ANI16 to ANI19	53	8	7.0

Remark The resistance and capacitance values shown in Table 12 - 7 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AVREFP and VDD voltages stabilize.

CHAPTER 13 COMPARATORS/PROGRAMMABLE GAIN AMPLIFIERS

The comparators and programmable gain amplifiers have the following functions.

13.1 Overview

○ Comparators

- A comparator is equipped with two channels (CMP0, CMP1).
- Positive-side input pins (CMP0P, CMP1P) can be connected.
- The output signal of a programmable gain amplifier can be used as the positive-side input signal of a comparator.
- The voltage from a dedicated 8-bit DAC (resolution of 256 with V_{DD}/AV_{REFP} or V_{SS}/AV_{REFM} as the internally generated reference voltage) can be selected as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event link controller (ELC) event signal can be output by detecting an active edge of the comparator output.
- When connected to the PWM option unit, the output signal from the comparator forcibly cuts the output from the timer output pin off by placing it in the high-Z state. When connected to the ELC, the output signal from the comparator forcibly cuts the output from the timer output pin off by fixing it to the high-Z, high output, or low output state. For details, see **8.4.4 Pulse Output Forced Cutoff**.
- The comparison result of the reference input voltage and the analog input voltage can be read by software.

○ Programmable gain amplifiers

- A programmable gain amplifier amplifies an input analog voltage. One among four amplification factors can be selected.
- The output signal of a programmable gain amplifier can be used as the positive-side input signal of a comparator.
- The output signal of a programmable gain amplifier can be selected as the analog input of an A/D converter.

Figure 13 - 1 lists the Block Diagram of Comparator and Programmable Gain Amplifier and Table 13 - 1 shows the Pin Configuration.

Figure 13 - 1 Block Diagram of Comparator and Programmable Gain Amplifier

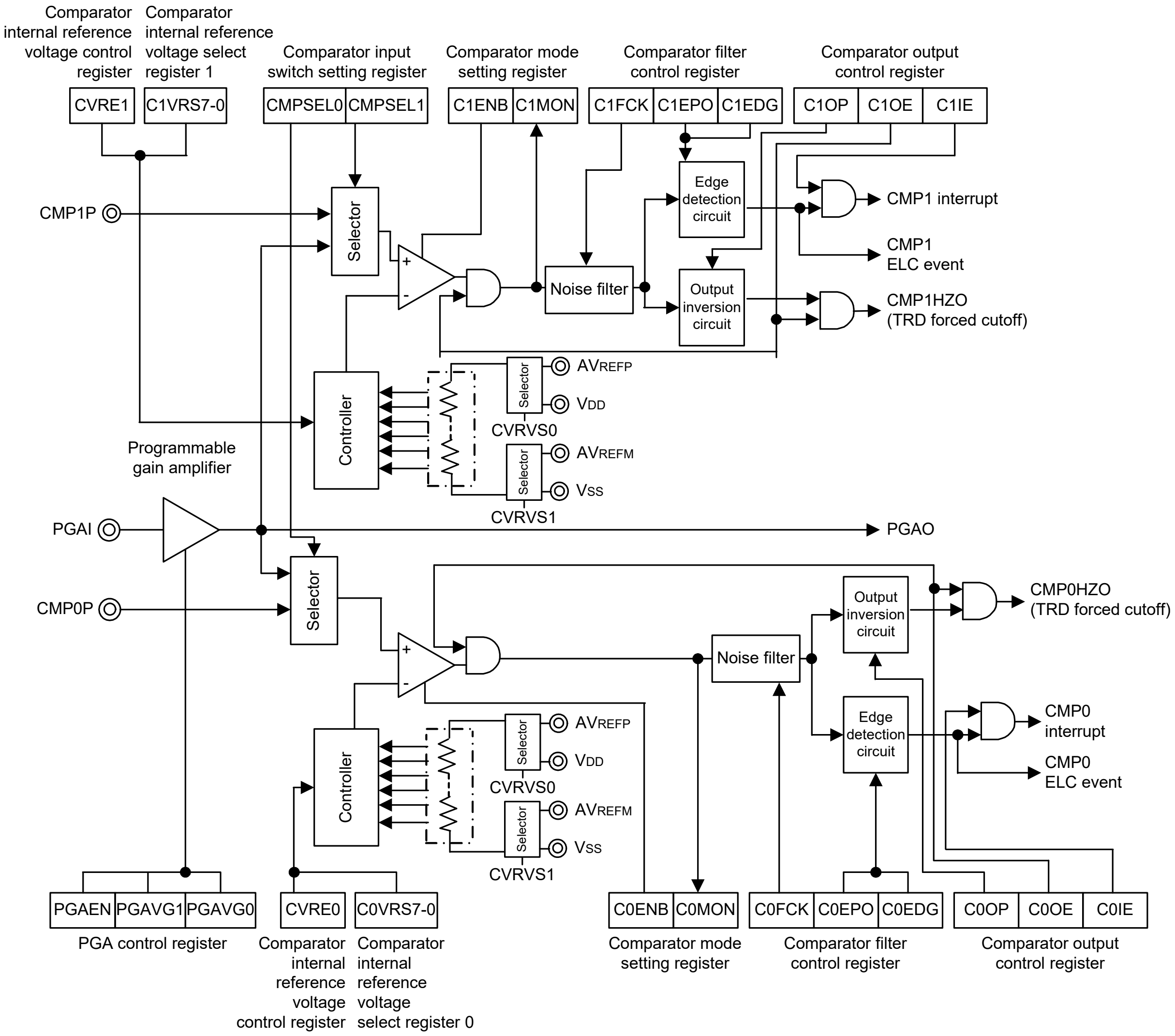


Table 13 - 1 Pin Configuration

Pin Name	I/O	Function
CMP0P	Input	Comparator 0 analog pin
CMP1P	Input	Comparator 1 analog pin
PGAI	Input	Programmable gain amplifier input pin

13.2 Registers

Registers lists the registers used for the comparators/programmable gain amplifiers.

Table 13 - 2 Registers

Register Name	Symbol	After Reset	Address	Access Size
Comparator mode setting register	COMPMDR	00H	F0340H	1 or 8
Comparator filter control register	COMPFIR	00H	F0341H	1 or 8
Comparator output control register	COMPOCR	00H	F0342H	1 or 8
Comparator internal reference voltage control register	CVRCTL	00H	F0343H	1 or 8
Comparator internal reference voltage select register 0	C0RVM	00H	F0344H	8
Comparator internal reference voltage select register 1	C1RVM	00H	F0345H	8
PGA control register	PGACTL	00H	F0346H	1 or 8
6-phase PWM option mode register	OPMR	00H	F026CH	8
6-phase PWM option status register	OPSR	00H	F026DH	8
6-phase PWM option Hi-Z start trigger register	OPHS	00H	F026EH	8
6-phase PWM option Hi-Z stop trigger register	OPHT	00H	F026FH	8
Peripheral enable register 1	PER1	00H	F007AH	8
Port mode control register 0	PMC0	FFH	F0060H	8
Port mode control register 12	PMC12	FFH	F006CH	8
Port mode register 0	PM0	FFH	FFF20H	8
Port mode register 12	PM12	FFH	FFF2CH	8

13.2.1 Comparator mode setting register (COMPMDR)

This register is used to enable or stop operation of the comparator and monitor comparator output.

The CiENB bit can be read or written in 1-bit or 8-bit units.

The CiMON bit can be read only.

Reset signal generation clears this register to 00H.

Figure 13 - 2 Format of Comparator mode setting register (COMPMDR)

Address: F0340H After reset: 00H R/W

Symbol	<7>	6	5	<4>	<3>	2	1	<0>
COMPMDR	C1MON	0	0	C1ENB Note	C0MON	0	0	C0ENB Note

C1MON	Comparator 1 monitor flag
0	CMP1 < comparator 1 reference voltage
1	CMP1 > comparator 1 reference voltage

C1ENB Note	Comparator 1 operation enable
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

C0MON	Comparator 0 monitor flag
0	CMP0 < comparator 0 reference voltage
1	CMP0 > comparator 0 reference voltage

C0ENB Note	Comparator 0 operation enable
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Note Do not set the CiENB bit to 0 when comparator i output is enabled (the CiOE bit in the COMPOCR register = 1). Also, do not set this bit to 1 (comparator i operation enabled) under the following conditions:

- When operation of the internal reference voltage i is stopped (the CVREi bit in the CVRCTL register = 0)
- When PGA operation is disabled (the CMPSELi bit in the CVRCTL register = 1 and the PGAEN bit in the PGACTL register = 0) while comparator input = PGA output is selected

Remark i = 0, 1

13.2.2 Comparator filter control register (COMPFIR)

This register is used to select the sampling clock for the digital filer and the edge detection for comparator output.

This register can be read or written in 1-bit or 8-bit units.

Reset signal generation clears this register to 00H.

Figure 13 - 3 Format of Comparator filter control register (COMPFIR)

Address: F0341H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK		C0EDG	C0EPO	C0FCK	

C1EDG	Comparator 1 edge detection selection Note 1
0	Interrupt request by comparator 1 one-edge detection
1	Interrupt request by comparator 1 both-edge detection

C1EPO	Comparator 1 edge polarity switching Note 1
0	Interrupt request at comparator 1 rising edge
1	Interrupt request at comparator 1 falling edge

C1FCK		Comparator 1 filter selection Note 1
Bit 5	Bit 4	
0	0	No comparator 1 filter
0	1	Comparator 1 filter enabled, sampling at f_{CLK}
1	0	Comparator 1 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 1 filter enabled, sampling at $f_{CLK}/32$

C0EDG	Comparator 0 edge detection selection Note 2
0	Interrupt request by comparator 0 one-edge detection
1	Interrupt request by comparator 0 both-edge detection

C0EPO	Comparator 0 edge polarity switching Note 2
0	Interrupt request at comparator 0 rising edge
1	Interrupt request at comparator 0 falling edge

C0FCK		Comparator 0 filter selection Note 2
Bit 1	Bit 0	
0	0	No comparator 0 filter
0	1	Comparator 0 filter enabled, sampling at f_{CLK}
1	0	Comparator 0 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 0 filter enabled, sampling at $f_{CLK}/32$

(**Notes** are listed on the next page.)

Note 1. If bits C1FCK1, C1FCK0, C1EPO, and C1EDG are changed, a comparator 1 interrupt and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR17 register for the ELC to 0 (not linked to comparator 1 output). In addition, clear bit 0 (CMPIF1) in interrupt request flag register 2H (IF2H) to 0.

If bits C1FCK1 to C1FCK0 are changed from 00B (no comparator 1 filter) to a value other than 00B (comparator 1 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal to the ELC.

Note 2. If bits C0FCK1, C0FCK0, C0EPO, and C0EDG are changed, a comparator 0 interrupt and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR16 register for the ELC to 0 (not linked to comparator 0 output). In addition, clear bit 7 (CMPIF0) in interrupt request flag register 2L (IF2L) to 0.

If bits C0FCK1 to C0FCK0 are changed from 00B (no comparator 0 filter) to a value other than 00B (comparator 0 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal to the ELC.

13.2.3 Comparator output control register (COMPOCR)

This register is used to control the polarity of comparator output, enable or disable output, and enable or disable interrupt output.

This register can be read or written in 1-bit or 8-bit units.

Reset signal generation clears this register to 00H.

Figure 13 - 4 Format of Comparator output control register (COMPOCR)

Address: F0342H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
COMPOCR	0	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

C1OP	Comparator 1 output polarity selection Note 1
0	Comparator 1 output is non-inverted output
1	Comparator 1 output is inverted output

C1OE	Comparator 1 output enable Notes 1, 2
0	Comparator 1 output stopped (CMP1HZO output fixed at low level)
1	Comparator 1 output enabled (CMP1HZO output enabled)

C1IE	Comparator 1 interrupt request enable Note 3
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	Comparator 0 output polarity selection Note 1
0	Comparator 0 output is non-inverted output
1	Comparator 0 output is inverted output

C0OE	Comparator 0 output enable Notes 1, 4
0	Comparator 0 output stopped (CMP0HZO output fixed at low level)
1	Comparator 0 output enabled (CMP0HZO output enabled)

C0IE	Comparator 0 interrupt request enable Note 5
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

(**Notes**, **Caution**, and **Remark** are listed on the next page.)

Note 1. The CiOE and CiOP bits are used to forcibly cut off PWM output by inputting the result of comparator i to the PWM option unit. These bits are not used to control output to ports.

Note 2. If the C1OE bit is changed, a comparator 1 interrupt request and an event signal to the ELC may be generated. Change this bit only after setting the ELSELR17 register for the ELC to 0 (not linked to comparator 1 output).
In addition, clear bit 0 (CMPIF1) in interrupt request flag register 2H (IF2H).

Note 3. If C1IE is changed from 0 (comparator 1 interrupt request disabled) to 1 (comparator 1 interrupt request enabled), the flag bit CMPIF1 in the interrupt control register may be set to 1 (interrupt requested), so clear bit 0 (CMPIF1) in interrupt request flag register 2H (IF2H) to 0 before using an interrupt.

Note 4. If the C0OE bit is changed, a comparator 0 interrupt request and an event signal to the ELC may be generated. Change this bit only after setting the ELSELR16 register for the ELC to 0 (not linked to comparator 0 output).
In addition, clear bit 7 (CMPIF0) in interrupt request flag register 2L (IF2L).

Note 5. If C0IE is changed from 0 (comparator 0 interrupt request disabled) to 1 (comparator 0 interrupt request enabled), the flag bit CMPIF0 in the interrupt control register may be set to 1 (interrupt requested), so clear bit 7 (CMPIF0) in interrupt request flag register 2L (IF2L) to 0 before using an interrupt.

Caution **Do not set the CiOE bit to 1 (comparator i output enabled) under the following conditions:**

- **When comparator i operation is disabled (the CiENB bit in the COMPMDR register = 0)**
- **When operation of the internal reference voltage i is stopped (the CVREi bit in the CVRCTL register = 0)**
- **When PGA operation is disabled (the CMPSELi bit in the CVRCTL register = 1 and the PGAEN bit in the PGACTL register = 0) while comparator input = PGA output is selected**

Remark i = 0, 1

13.2.4 Comparator internal reference voltage control register (CVRCTL)

This register is used to set the input signal of the comparator and enable or stop operation of the internal reference voltage.

This register can be read or written in 1-bit or 8-bit units.

Reset signal generation clears this register to 00H.

Figure 13 - 5 Format of Comparator internal reference voltage control register (CVRCTL)

Address: F0343H After reset: 00H R/W

Symbol	7	6	<5>	4	3	2	<1>	0
CVRCTL	0	CMPSEL1 Note 1	CVRE1	CVRVS1 Note 2	0	CMPSEL0 Note 1	CVRE0	CVRVS0 Note 2

CMPSEL1 Note 1	Function
0	CMP1P pin is selected as the positive-side input of comparator 1
1	PGA output is selected as the positive-side input of comparator 1

CVRE1	Function
0	Operation of internal reference voltage 1 stopped
1	Operation of internal reference voltage 1 enabled

CVRVS1 Note 2	Function
0	V_{SS} is selected as the GND of the internal reference voltage and the GND of the PGA feedback resistor
1	AV_{REFM} is selected as the GND of the internal reference voltage and the GND of the PGA feedback resistor

CMPSEL0 Note 1	Function
0	CMP0PI pin is selected as the positive-side input of comparator 0
1	PGA output is selected as the positive-side input of comparator 0

CVRE0	Function
0	Operation of internal reference voltage 0 stopped
1	Operation of internal reference voltage 0 enabled

CVRVS0 Note 2	Function
0	V_{DD} is selected as the power supply of the internal reference voltage
1	AV_{REFP} is selected as the power supply of the internal reference voltage

Note 1. Rewrite the CMPSELi bit while comparator i operation is disabled (CiENB = 0).

Note 2. Rewrite the CVRVSi bit while operation of the internal reference voltage i is stopped (CVREi = 0).

Remark i = 0, 1

13.2.5 Comparator internal reference voltage selection register i (CiRVM)

This register is used to set the internal reference voltage of the comparator.

This register can be read or written in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 13 - 6 Format of Comparator internal reference voltage selection register i (CiRVM)

Address: F0344H (C0RVM), F0345H (C1RVM) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CiRVM	CiVRS7	CiVRS6	CiVRS5	CiVRS4	CiVRS3	CiVRS2	CiVRS1	CiVRS0

CiVRS7 to CiVRS0 Note	Comparator i internal reference voltage selection
00000000	{(AV$_{REFP}$ or V$_{DD}$)/256} × 0
00000001	{(AV$_{REFP}$ or V$_{DD}$)/256} × 1
00000010	{(AV$_{REFP}$ or V$_{DD}$)/256} × 2
⋮	⋮
11111101	{(AV$_{REFP}$ or V$_{DD}$)/256} × 253
11111110	{(AV$_{REFP}$ or V$_{DD}$)/256} × 254
11111111	{(AV$_{REFP}$ or V$_{DD}$)/256} × 255

Note Rewrite the CiRVM register while operation of the internal reference voltage i is stopped (CVREi = 0).

Remark i = 0, 1

13.2.6 PGA control register (PGACTL)

This register is used to enable or stop PGA operation and set the amplification factor.
This register can be read or written in 1-bit or 8-bit units.
Reset signal generation clears this register to 00H.
The bits other than PGAEN in this register must be rewritten while PGA operation is stopped (PGAEN = 0).

Figure 13 - 7 Format of PGA control register (PGACTL)

Address: F0346H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PGACTL	PGAEN	0	0	0	0	0	PGAVG1 Note	PGAVG0 Note

PGAEN	Function
0	PGA operation stopped
1	PGA operation enabled

PGAVG1 Note	PGAVG0 Note	Function
0	0	× 4 is selected as the gain
0	1	× 8 is selected as the gain
1	0	× 16 is selected as the gain
1	1	× 32 is selected as the gain

Note Rewrite the PGAVG1 and PGAVG0 bits while PGA operation is stopped (PGAEN = 0).

13.2.7 6-phase PWM option mode register (OPMR)

The OPMR register is used to set the operating mode of the PWM option unit.

TRDIOB0, TRDIOD0, TRDIOA1, TRDIOC1, TRDIOB1, and TRDIOD1 are set as the 6-phase PWM output of timer RD while the Hi-Z of these pins can be controlled using a comparator. Set the operating mode used to control the conditions for setting and canceling the Hi-Z of these pins simultaneously.

The OPMR register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13 - 8 Format of 6-phase PWM option mode register (OPMR)

Address: F026CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OPMR	0	0	0	0	0	0	HDM	0

HDM	PWM option operating mode
0	Two-stage overcurrent detection mode
1	Overcurrent/induced current detection mode

13.2.8 6-phase PWM option status register (OPSR)

The OPSR register is used to monitor various statuses of the PWM option unit.

The OPSR register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13 - 9 Format of 6-phase PWM option status register (OPSR)

Address: F026DH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	HZOF	HZIF1	HZIF0

HZOF	Hi-Z control signal operating state
0	Normal timer output
1	Hi-Z output state

HZIF1	On-chip comparator 1 output signal state
0	On-chip comparator 1 output signal is low level
1	On-chip comparator 1 output signal is high level

HZIF0	On-chip comparator 0 output signal state
0	On-chip comparator 0 output signal is low level
1	On-chip comparator 0 output signal is high level

13.2.9 6-phase PWM option Hi-Z start trigger register (OPHS)

The OPHS register is used to set a software trigger for the Hi-Z control circuit. Set the OPHS0 bit to 1 when starting the Hi-Z output of the 6-phase PWM output pins of timer RD. Since OPHS0 is a trigger bit, it is cleared immediately when the pins are set to the Hi-Z state (HZOF = 1).

The OPHS register can be written by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13 - 10 Format of 6-phase PWM option Hi-Z start trigger register (OPHS)

Address: F026EH After reset: 00H W

Symbol	7	6	5	4	3	2	1	0
OPHS	0	0	0	0	0	0	0	OPHS0

OPHS0	Hi-Z control signal of on-chip comparator output
0	Invalid
1	Hi-Z control signal is high level (the pins are set to Hi-Z output)

Remark The read value of the OPHS register is always 00H.

13.2.10 6-phase PWM option Hi-Z stop trigger register (OPHT)

The OPHT register is used to set a software trigger for the Hi-Z control circuit. Set the OPHT0 bit to 1 when canceling the Hi-Z of the 6-phase PWM output pins of timer RD. Since OPHT0 is a trigger bit, it is cleared immediately when the Hi-Z cancel request signal is output.

The OPHT register can be written by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13 - 11 Format of 6-phase PWM option Hi-Z stop trigger register (OPHT)

Address: F026FH After reset: 00H W

Symbol	7	6	5	4	3	2	1	0
OPHT	0	0	0	0	0	0	0	OPHT0

OPHT0	HDM	HZIF1	HZIF0	Hi-Z cancel control of on-chip comparator output
0	—	—	—	Invalid
1	0	0	0	The Hi-Z of the 6-phase output is canceled in synchronization with the PWM period.
	Other than above			The Hi-Z of the 6-phase output cannot be canceled.

Remark 1. The read value of the OPHT register is always 00H.

Remark 2. HDM: Bit 1 in 6-phase PWM option mode register (OPMR)
HZIF0, HZIF1: Bits 0 and 1 in 6-phase PWM option status register (OPSR)

13.2.11 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware.

Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the PWM option unit, the comparator, and the programmable gain amplifier, be sure to set bit 6 (PWMOPEN) and bit 5 (OACMPEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13 - 12 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>
PER1	TMKAEN	PWMOPEN	OACMPEN	TRD0EN	0	0	0	TRJ0EN

PWMOPEN	Control of PWM option unit input clock supply
0	Stops input clock supply. • SFR used by the PWM option unit cannot be written. • The PWM option unit is in the reset state.
1	Enables input clock supply. • SFR used by the PWM option unit can be read and written.

OACMPEN	Control of input clock supply for comparators 0 and 1 and the programmable gain amplifier
0	Stops input clock supply. • SFR used by comparators 0 and 1 and the programmable gain amplifier cannot be written. • Comparators 0 and 1 and the programmable gain amplifier are in the reset state.
1	Enables input clock supply. • SFR used by Comparators 0 and 1 and the programmable gain amplifier can be read and written.

Caution 1. When setting the PWM option unit, be sure to set the PWMOPEN bit to 1 first.
If PWMOPEN = 0, writing to a control register of the PWM option unit is ignored, and all read values are default values.

Caution 2. When setting comparators 0 and 1 and the programmable gain amplifier, be sure to set the OACMPEN bit to 1 first.
If OACMPEN = 0, writing to a control register of the comparators 0 and 1 and the programmable gain amplifier is ignored, and all read values are default values.

Caution 3. Be sure to clear bits 1 to 3 to 0.

13.2.12 Registers controlling port functions of analog input pins

When using the CMP0P, CMP1P, and PGAI pins for input of comparators/programmable gain amplifiers, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit to 1.

13.3 Operation

Comparator 0 and comparator 1 operate independently. Their setting method and operation are the same.
Table 13 - 3 lists the Procedure for Setting Comparator Associated Registers.

Table 13 - 3 Procedure for Setting Comparator Associated Registers

Step	Register	Bit	Setting Value
1	PER1	CMPEN	1 (input clock supply)
2	PMC0, PMC12	PMC00, PMC01, PMC120	Select the function of pins CMP0P, CMP1P, and PGAI. Set the PMC00, PMC01, PMC120 bit to 1 (analog input). Set the PM00, PM01, PM120 bit to 1 (input mode).
	PM0, PM12	PM00, PM01, PM120	
3	PGACTL	PGAVG0, PGCVG1	Select the gain. Note 3
4	PGACTL	PGAEN	1 (PGA operation enabled) Note 3
5	Wait for the PGA stabilization time (min. 10 μs).		
6	CVRCTL	CMPSELi	Select the positive-side input of comparator i. Note 3
7	CiRVM	CiVRSn	Set the value of the internal reference voltage.
8	CVRCTL	CVRVSi	Select the power supply and the GND of the internal reference voltage.
9	CVRCTL	CVREi	1 (operation of internal reference voltage i enabled)
10	Wait for the reference voltage stabilization time (min. 20 μs).		
11	Select the function of CMPiP (input) and PGAI (input). Note 3		
12	COMPMDR	CiENB	1 (comparator i operation enabled)
13	Wait for the comparator stabilization time (min. 3 μs).		
14	COMPFIR	CiFCK	Select whether the digital filter is used or not and the sampling clock.
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).
15	COMPOCR	CiOP, CiOE	Set the comparator i output (select the polarity and output enabled).
		CiIE	Set the interrupt request output enabled or disabled.
16	PRi2L, PRi2H Note 1	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.
17	MK2L, MK2H Note 1	CMPMKi	When using an interrupt: Select the interrupt masking.
18	IF2L, IF2H Note 1	CMPIFi	When using an interrupt: 0 (no interrupt requested) Note 2

Note 1. PRi2L, MK2L, and IF2L are the interrupt control registers for comparator 0. PRi2H, MK2H, and IF2H are the interrupt control registers for comparator 1.

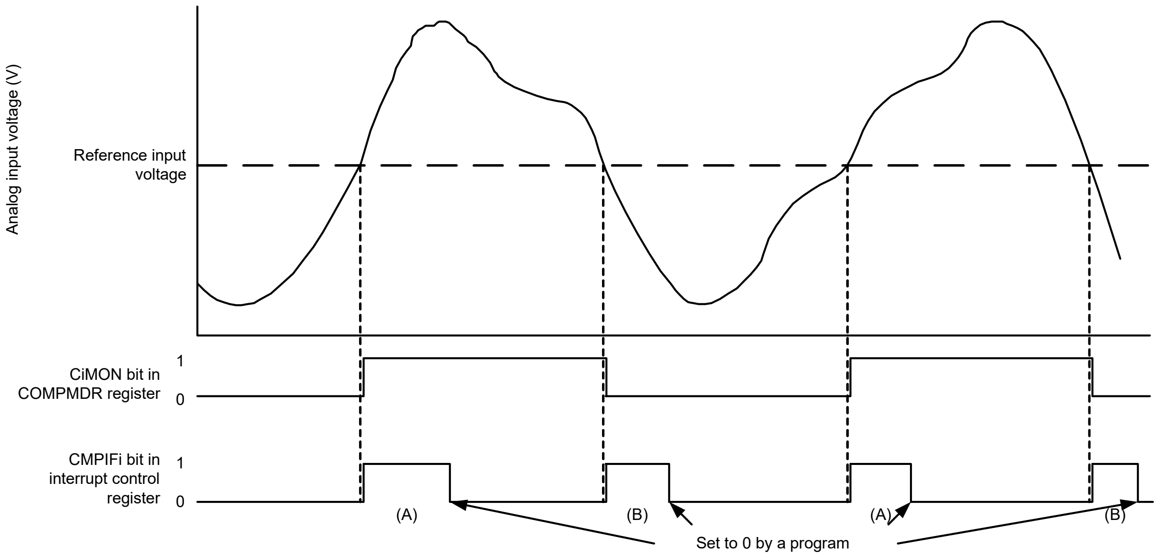
Note 2. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Note 3. Set this bit when using the PGA.

Figure 13 - 13 shows a Comparator i (i = 0, 1) Operation Example. The CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage.
When using the comparator i interrupt, set CiIE bit in the COMPOCR register to 1 (comparator i interrupt request enabled). If the comparison result changes at this time, a comparator i interrupt request is generated. For details on interrupt requests, refer to **13.3.2 Comparator i interrupts**.

Remark i = 0, 1; n = 0 to 7

Figure 13 - 13 Comparator i (i = 0, 1) Operation Example



Caution **The above diagram applies when CiFCK = 00B (no comparator i filter) and CiEDG = 1B (interrupt request by comparator i both-edge detection) in the COMPFIR register. When CiEDG = 0 (interrupt request by comparator i one-edge detection) and CiEPO = 0 (interrupt request at comparator i rising edge), CMPIFi changes only as shown by (A). When CiEDG = 0 (interrupt request by comparator i one-edge detection) and CiEPO = 1 (interrupt request at comparator i falling edge), CMPIFi changes only as shown by (B).**

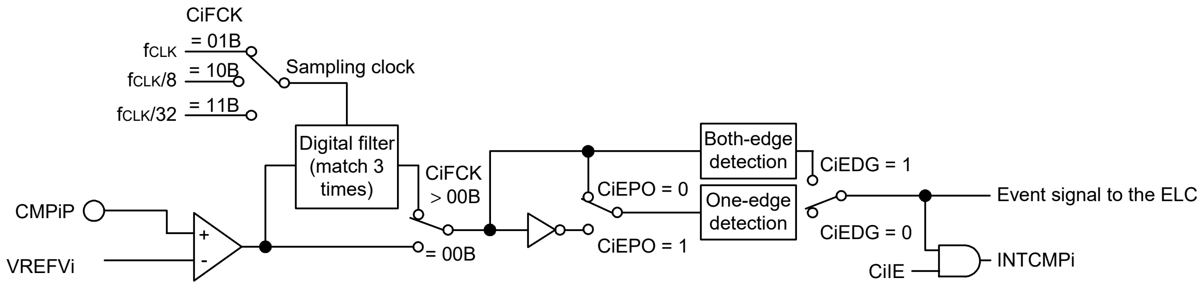
Remark i = 0, 1

13.3.1 Comparator i digital filter (i = 0, 1)

Comparator i contains a digital filter. The sampling clock can be selected by the CiFCK bit in the COMPFIR register. The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock.

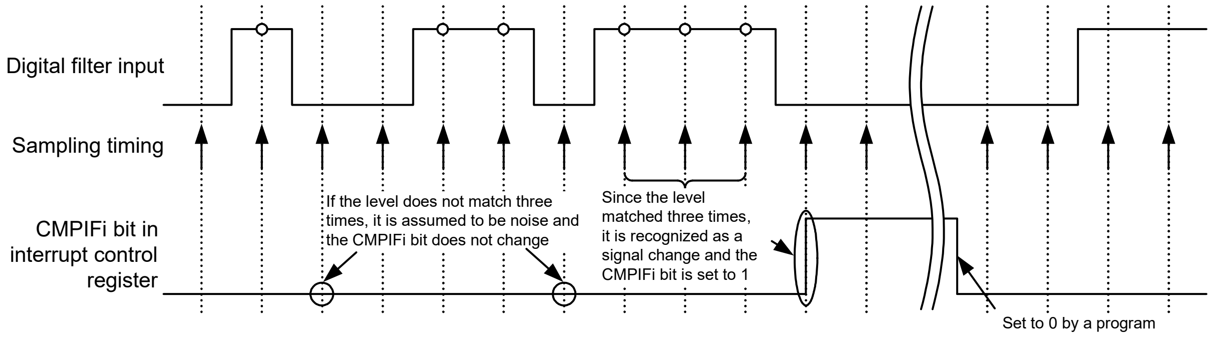
Figure 13 - 14 shows the Comparator i Digital Filter and Edge Detection Configuration, and Figure 13 - 15 shows the Comparator i Digital Filter and Interrupt Operation Example.

Figure 13 - 14 Comparator i Digital Filter and Edge Detection Configuration



Remark CiFCK, CiEPO, CiEDG: Bits in COMPFIR register
CiIE: Bit in COMPOCR register

Figure 13 - 15 Comparator i Digital Filter and Interrupt Operation Example



Caution **The above operation example applies when the CiFCK bit (i = 0, 1) in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).**

13.3.2 Comparator i interrupts

The comparator generates interrupt requests from two sources, comparator 0 and comparator 1. The comparator i interrupt each uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and a single vector.

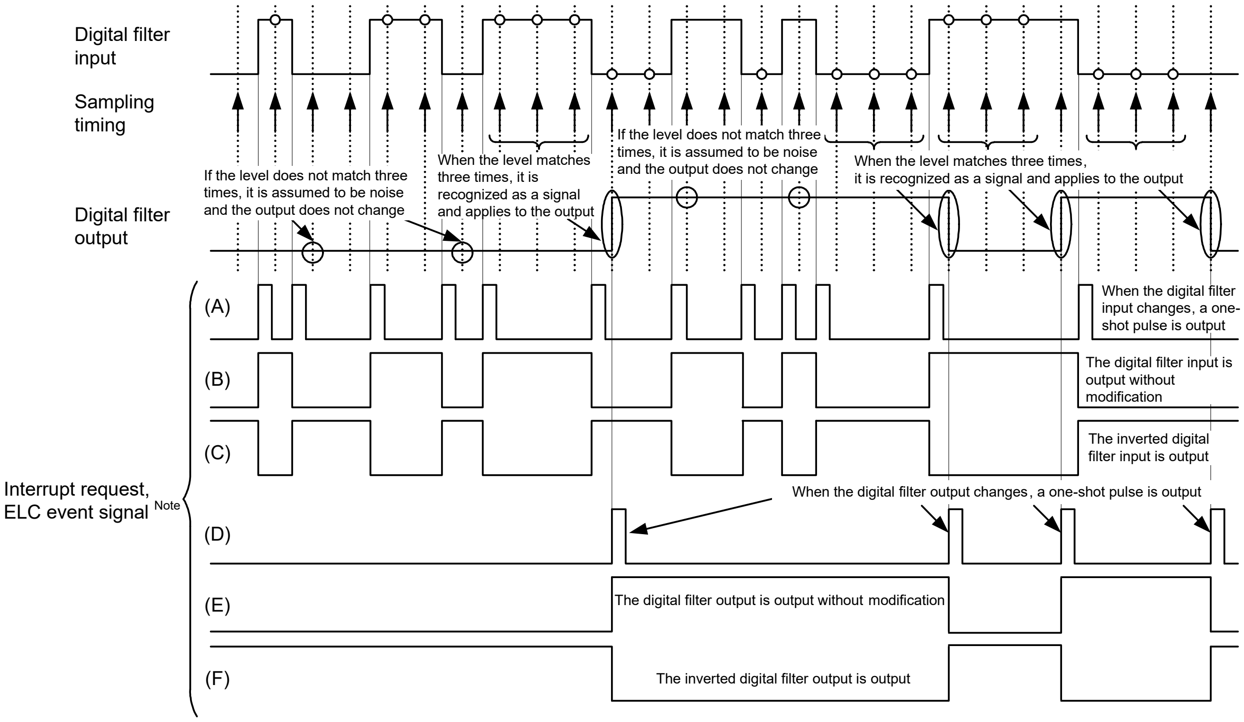
When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (comparator i interrupt request enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, refer to **13.2.2 Comparator filter control register (COMPFIR)** and **13.2.3 Comparator output control register (COMPOCR)**.

13.3.3 Event signal output to event link controller (ELC)

An event signal to the ELC is generated by detecting the edge for the digital filter output set by the COMPFIR register, which is the same as the condition for interrupt request generation. However, unlike interrupt requests, the event signal to the ELC are always output regardless of the CiIE bit in the COMPOCR register. Set registers ELSELR17 and ELSELR18 for the ELC to select the event output destination and to stop linking events.

Figure 13 - 16 Digital Filter and Interrupt Request/Event Signal Output to the ELC Operation



Note When the CiIE bit (i = 0, 1) is 1, the same waveform is generated for an interrupt request and an ELC event signal. When the CiIE bit (i = 0, 1) is 0, the value is fixed at 0 for an interrupt request only.

The waveforms of (A), (B), and (C) are shown for an operation example when the CiFCK bits (i = 0, 1) in the COMPFIR register are 00B (no digital filter). The waveforms (D), (E), and (F) are shown for an operation example when the CiFCK bits (i = 0, 1) in the COMPFIR register are 01B, 10B, or 11B (digital filter enabled).
(A) and (D) apply when the CiEDG bit is set to 1 (both edges), (B) and (E) when the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), and (C) and (F) when the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge).

13.3.4 Stopping or supplying comparator clock

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

<1> Set the CiOE bit in the COMPMDR register to 0 (comparator i output stopped).
<2> Set the CMPIFi bit in the IF2L and IF2H registers to 0 (clear any unnecessary interrupt before stopping the comparator).
<3> Set the OACMPEN bit in the PER1 register to 0.

When the clock is stopped by setting PER1, all the internal registers in the comparator are initialized. To use the comparator again, follow the procedure in Table 13 - 3 to set the registers.

Remark i = 0, 1

13.3.5 Using comparator output to forcibly cut off timer RD output

In the PWM mode, reset synchronous mode, compulsory PWM mode, or PWM3 mode, pulse output can be cut off in response to comparator output.
See **8.4.4 Pulse Output Forced Cutoff**.

Remark ij = B0, D0, A1, B1, C1, D1

13.4 PWM Option Unit

There are the following two overcurrent detection functions.

(1) Two-stage overcurrent detection function

- When the comparator 0 reference voltage < the input signal voltage < the comparator 1 reference voltage
- When the comparator 1 reference voltage < the input signal voltage

(2) Overcurrent/induced current detection function

- When the input signal voltage (induced current) < the comparator 0 reference voltage
- When the comparator 1 reference voltage < the input signal voltage (overcurrent)

13.4.1 Two-stage overcurrent detection function

The 6-phase PWM output pins are set to the Hi-Z state when the comparator 0 reference voltage < the input signal voltage.
There are the following two methods to restart timer output.

<1> After the 6-phase PWM output pins are set to Hi-Z while the comparator 0 reference voltage < the input signal voltage < the comparator 1 reference voltage, when the input signal voltage < the comparator 0 reference voltage, timer output is automatically restarted in synchronization with the timer period.
<2> When the input signal voltage is lower than the comparator 0 reference voltage after the comparator 1 reference voltage < the input signal voltage, timer output is restarted in synchronization with the timer period by setting the register.

13.4.2 Overcurrent/induced current detection function

The 6-phase PWM output pins are set to the Hi-Z state when the input signal voltage (induced current) < the comparator 0 reference voltage or the comparator 1 reference voltage < the input signal voltage (overcurrent).
When the input signal voltage is higher than the comparator 0 reference voltage and lower than the comparator 1 reference voltage, timer output is automatically restarted in synchronization with the timer period.

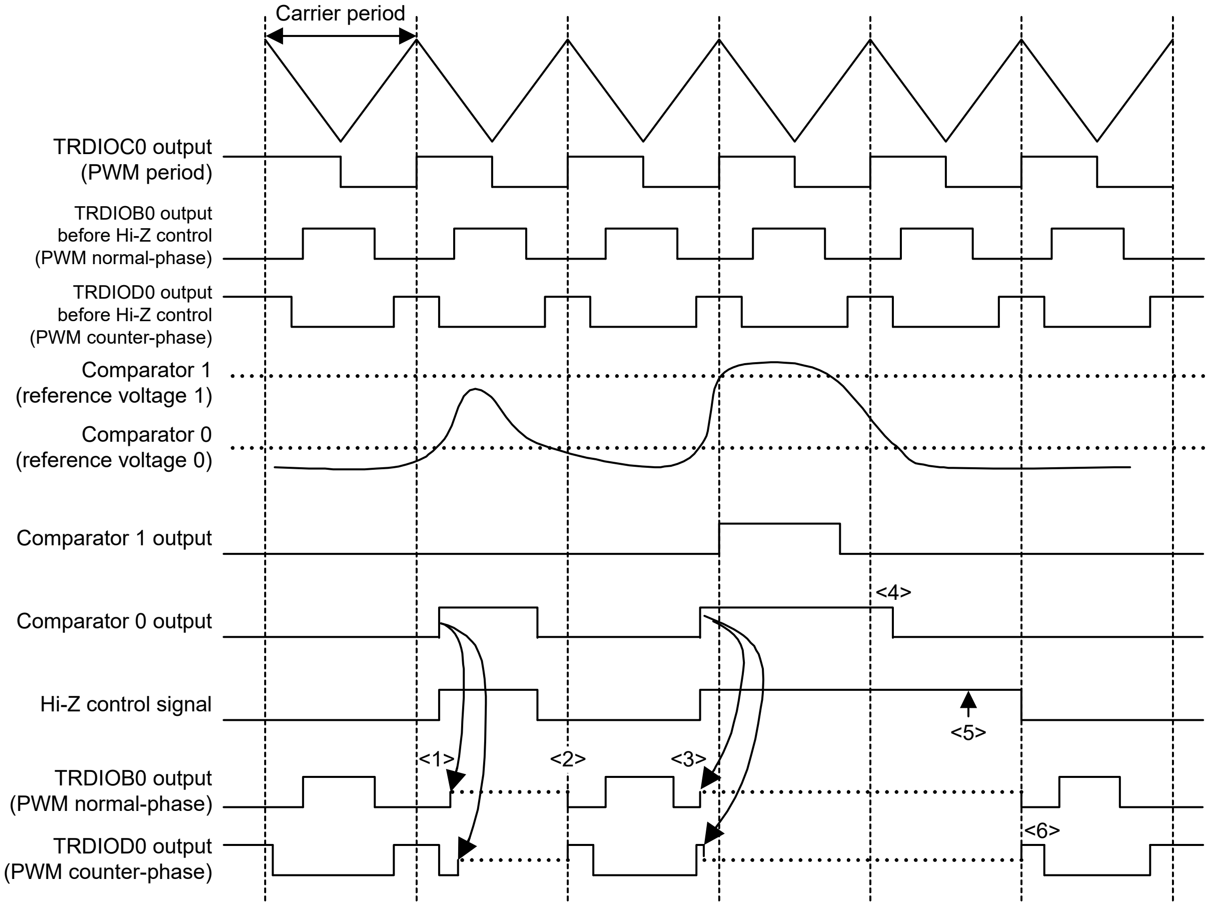
13.4.3 Operation example of overcurrent detection function

The Hi-Z of the port pins is controlled by setting the comparator 0 reference voltage < the comparator 1 reference voltage and inputting the same signal to the positive-side input of comparators 0 and 1.
The overcurrent detection function is set by the OPMR register. Table 13 - 4 lists the Relationship between Register Settings and Overcurrent Detection Function Operation. Figure 13 - 17 shows an Operation Example of Two-Stage Overcurrent Detection Function (TRDIOB0, TRDIOD0). Figure 13 - 18 shows an Operation Example of Overcurrent/Induced Detection Function (TRDIOB0, TRDIOD0).

Table 13 - 4 Relationship between Register Settings and Overcurrent Detection Function Operation

OPMR Register	Input Signal	Software Start Trigger OPHS0	Software Stop Trigger OPHT0	Hi-Z Output Pin	Operation Example
HDM Bit					
0	Comparator 0 output signal, comparator 1 output signal	Valid	Valid	6-phase PWM output	Figure 13 - 17
1			Invalid		Figure 13 - 18

Figure 13 - 17 Operation Example of Two-Stage Overcurrent Detection Function (TRDIOB0, TRDIOD0)



<1> The output of the TRDIOB0 and TRDIOD0 pins is set to the Hi-Z state when the rising edge of the comparator 0 output signal is detected.

<2> After the falling edge of the comparator 0 output signal is detected, the Hi-Z state of the TRDIOB0 and TRDIOD0 pins is canceled in synchronization with the timer carrier period.

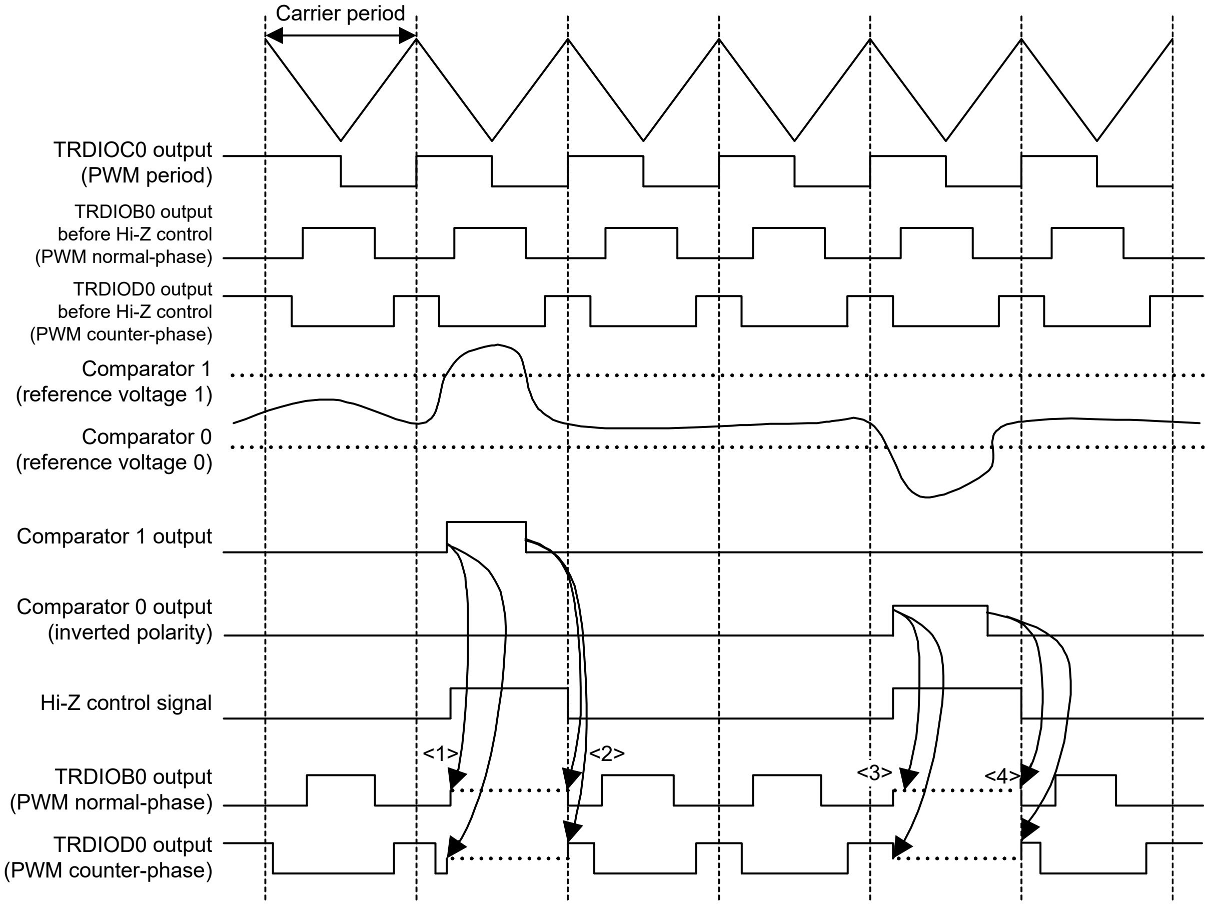
<3> The output of the TRDIOB0 and TRDIOD0 pins is set to the Hi-Z state when the rising edge of the comparator 1 output signal or comparator 0 output signal is detected.

<4> The Hi-Z state of the TRDIOB0 and TRDIOD0 pins is not canceled even when the falling edge of the comparator 1 output signal or comparator 0 output signal is detected.

<5> Write 1 to the OPHT0 bit after both the comparator 1 output signal and the comparator 0 output signal become inactive level.

<6> The Hi-Z state of the TRDIOB0 and TRDIOD0 pins is canceled in synchronization with the carrier period.

Figure 13 - 18 Operation Example of Overcurrent/Induced Detection Function (TRDIOB0, TRDIOD0)



<1> The output of the TRDIOB0 and TRDIOD0 pins is set to the Hi-Z state when the rising edge of the comparator 1 output signal is detected.

<2> After the falling edge of the comparator 1 output signal is detected, the Hi-Z state of the TRDIOB0 and TRDIOD0 pins is canceled in synchronization with the timer carrier period.

<3> The output of the TRDIOB0 and TRDIOD0 pins is set to the Hi-Z state when the rising edge of the comparator 0 output signal is detected.

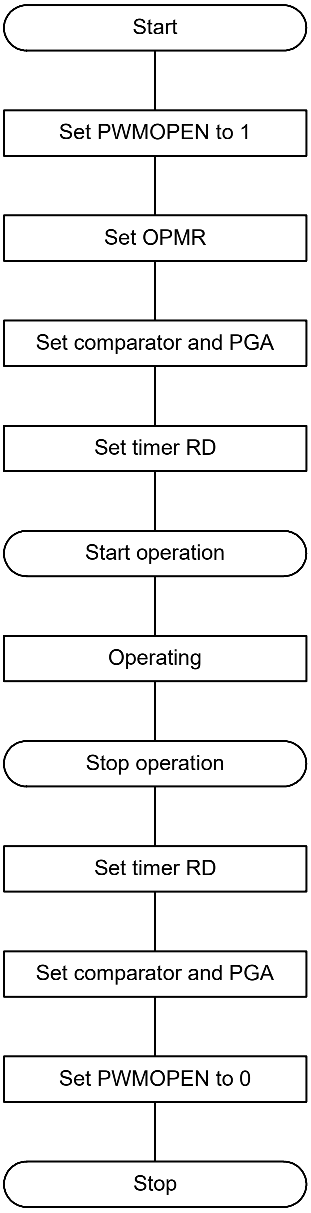
<4> After the falling edge of the comparator 0 output signal is detected, the Hi-Z state of the TRDIOB0 and TRDIOD0 pins is canceled in synchronization with the timer carrier period.

Caution Comparator 0 output is set to high-level output when the input signal voltage is lower than reference voltage 0, and low-level output when the input signal voltage is higher than reference voltage 0. Set the C0OP bit in the comparator output control register (COMPOCR) to 1 (comparator 0 output is inverted output).

13.4.4 Setting procedure

Figures 13 - 19 shows the Setting Procedure for Using Comparator Output to Forcibly Cut Off Timer RD Output.

Figure 13 - 19 Setting Procedure for Using Comparator Output to Forcibly Cut Off Timer RD Output



Set the corresponding bit in the register used to enable or stop clock supply to 1, cancel the reset state of the PWM option unit, and start clock supply.

Use the on-chip comparator output signal as the Hi-Z control signal, and select two-stage overcurrent detection mode or overcurrent/induced current detection mode.

Start operation of the comparator and the PGA Note.

Set timer RD to reset PWM mode or complementary PWM mode, and start PWM output.

Stop the PWM output of timer RD.

Stop operation of the comparator and the PGA.

Set the corresponding bit in the register used to enable or stop clock supply to 0, set the PWM option unit to the reset state, and stop clock supply.

Note In overcurrent/induced current detection mode, set the C0OP bit in the comparator output control register (COMPOCR) to 1 (comparator 0 output is inverted output).

Remark To use the PWM option unit again, set the above procedure one more time from the beginning.

13.4.5 Usage Note on PWM Option Unit

When using the PWM option unit, do not use the cutoff function of timer RD through INTP0 at the same time.

CHAPTER 14 SERIAL ARRAY UNIT

Serial array unit has four serial channels. All channels can achieve UART, and only channel 0 can achieve simplified SPI (CSI^Note) and simplified I^{2}C.
Function assignment of each channel supported by the RL78/G1G is as shown below.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

- 30, 32, 44-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and IIC00 cannot be used.

Caution **Most of the following descriptions in this chapter use the units and channels of the 44-pin products as an example.**

14.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G1G has the following features.

14.1.1 Simplified SPI (CSI00)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.
Simplified SPI is a clock synchronous communications function that uses three lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).
For details about the settings, see **14.5 Operation of Simplified SPI (CSI00) Communication**.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]
- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate Note
  During master communication: Max. $f_{MCK}/2$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only following simplified SPI (CSI) can be specified for asynchronous reception.
- CSI00

In addition, CSI00 supports the slave select input function.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**.

14.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]
- Data length of 7, 8, or 9 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

In addition, UART of following channels supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following UART can be specified when FRQSEL4 in the option byte (000C2H) = 0 in the SNOOZE mode.
- UART0

14.1.3 Simplified I^2C (IIC00)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^2C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **14.8 Operation of Simplified I^2C (IIC00) Communication**.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function [Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]
- Transfer end interrupt

[Error detection flag]
- Parity error (ACK error), or overrun error

* [Functions not supported by simplified I^2C]
- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **14.8.3 (2)** for details.

Remark m: Unit number (m = 0),
n: Channel number (n = 0)

14.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 14 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits Note 1
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) Notes 1, 2
Serial clock I/O	SCK00 pin (for simplified SPI), SCL00 pin (for simplified I^{2}C)
Serial data input	SI00 pin (for simplified SPI), RxD0 pin, RxD1 pin
Serial data output	SO00 pin (for simplified SPI), TxD0 pin, TxD1 pin, output controller
Serial data I/O	SDA00 pin (for simplified I^{2}C)
Slave select input	$\overline{\text{SSI00}}$ pin (for slave select input function)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Registers controlling port functions of serial input/output pins • Port output mode registers 0, 3, 5 (POM0, POM3, POM5) • Port mode registers 0, 3, 5, 6 (PM0, PM3, PM5, PM6) • Port registers 0, 3, 5, 6 (P0, P3, P5, P6)

(**Notes** and **Remark** are listed on the next page.)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.

• 30 to 44-pin products and mn = 00, 01:	lower 9 bits
• Other than above:	lower 8 bits

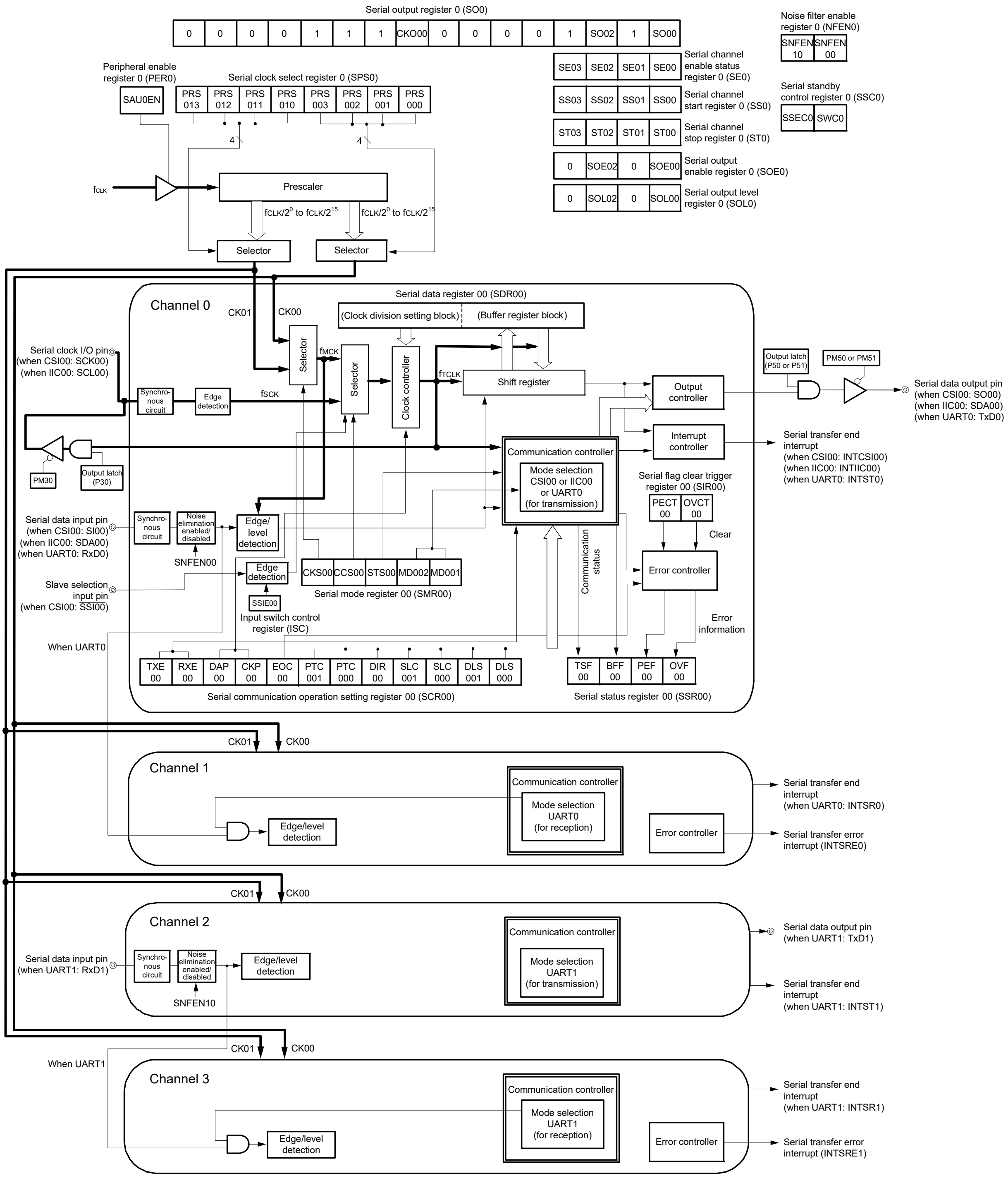
Note 2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication........SIOp (CSIp data register)
- UARTq reception..............RXDq (UARTq receive data register)
- UARTq transmission ....TXDq (UARTq transmit data register)
- IICr communication ....SIOr (IICr data register)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00),
q: UART number (q = 0, 1), r: IIC number (r = 00)

Figure 14 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 14 - 1 Block Diagram of Serial Array Unit 0



14.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used Note 1.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

14.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) Note 1 or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register) Note 1

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written Note 2 as the following SFR, depending on the communication mode.

- CSIp communication........... SIOp (CSIp data register)
- UARTq reception .... RXDq (UARTq receive data register)
- UARTq transmission .... TXDq (UARTq transmit data register)
- IICr communication............. SIOr (IICr data register)

Reset signal generation clears the SDRmn register to 0000H.

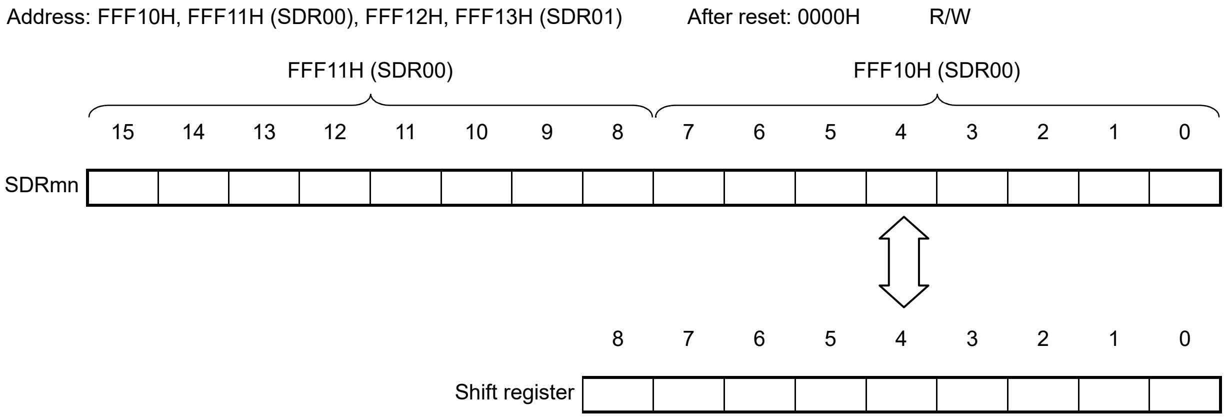
Note 1. Only following UART can be specified for the 9-bit data length.
- 30 to 44-pin products: UART0

Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, "0" is stored in bits 0 to 8 in bit portions that exceed the data length.

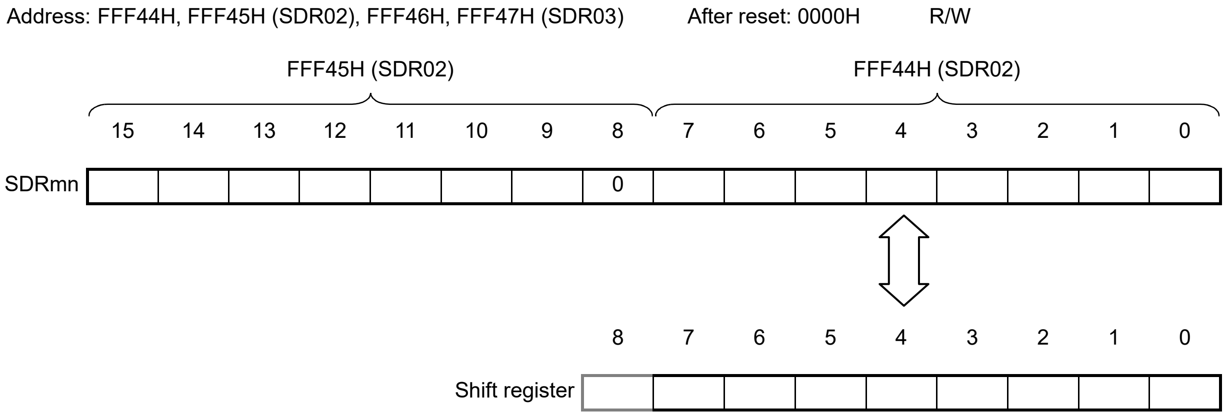
Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00), q: UART number (q = 0, 1), r: IIC number (r = 00)

Figure 14 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)



Remark For the function of the higher 7 bits of the SDRmn register, see **14.3 Registers Controlling Serial Array Unit**.

Figure 14 - 3 Format of Serial data register mn (SDRmn) (mn = 02, 03)



Caution **Be sure to clear bit 8 to “0”.**

Remark For the function of the higher 7 bits of the SDRmn register, see **14.3 Registers Controlling Serial Array Unit**.

14.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Registers controlling port functions of serial input/output pins
- Port output mode registers 0, 3, 5 (POM0, POM3, POM5)
- Port mode registers 0, 3, 5, 6 (PM0, PM3, PM5, PM6)
- Port registers 0, 3, 5, 6 (P0, P3, P5, P6)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

14.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

Figure 14 - 4 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PER0	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops supply of input clock. • SFR used by serial array unit 0 cannot be written. • Serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit 0 can be read/written.

Caution 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode registers 0, 3, 5 (PIM0, PIM3, PIM5), port output mode registers 0, 3, 5 (POM0, POM3, POM5), port mode registers 0, 3, 5, 6 (PM0, PM3, PM5, PM6), port mode control register 0 (PMC0), and port registers 0, 3, 5, 6 (P0, P3, P5, P6).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 4, 6, 7

14.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 14 - 5 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk) Note					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	0.75 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.75 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.88 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.44 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.72 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.47 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array units (SAUs).

Caution Be sure to clear bits 15 to 8 to “0”.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. m: Unit number (m = 0)

Remark 3. k = 0, 1

14.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (simplified SPI (CSI), UART, or I^{2}C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 14 - 6 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI (CSI), UART transmission, and simplified I^{2}C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 and SMR03 registers only.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, or SMR02 register) to “0”. Be sure to set bit 5 to “1”.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00),
q: UART number (q = 0, 1), r: IIC number (r = 00)

Figure 14 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03)   After reset: 0020H   R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^{2}C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 and SMR03 registers only.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, or SMR02 register) to "0". Be sure to set bit 5 to "1".**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00),
q: UART number (q = 0, 1), r: IIC number (r = 00)

14.3.4 Serial communication operation setting register mn (SCRmn)

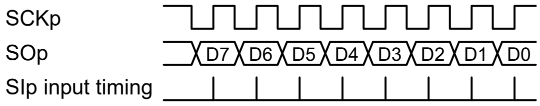
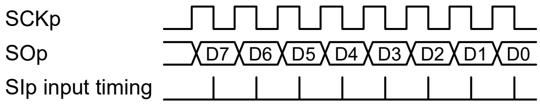
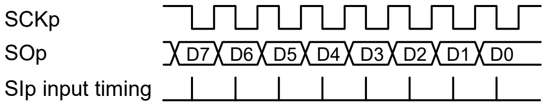
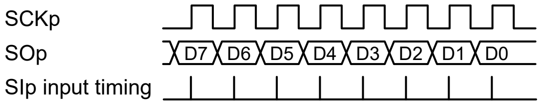
The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.
Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).
The SCRmn register can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets the SCRmn register to 0087H.

Figure 14 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1 Note 2	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 0, 1))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOCmn = 0 in the simplified SPI (CSI) mode, simplified I^2C mode, and during UART transmission Note 3.	

Note 1. The SCR00 and SCR02 registers only.

Note 2. The SCR00 and SCR01 registers only.
Others are fixed to 1.

Note 3. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution **Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 or SCR03 register to 0, as well as bit 1 of the SCR02 or SCR03 registers.). Be sure to set bit 2 to "1".**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00))

Figure 14 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1 Note 2	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 3.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the simplified SPI (CSI) mode and simplified I^2C mode.			

DIR mn	Selection of data transfer sequence in simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLC mn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1 Note 2	DLS mn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^2C mode.		

Note 1. The SCR00, and SCR02 registers only.

Note 2. The SCR00 and SCR01 registers only.
Others are fixed to 1.

Note 3. 0 is always added regardless of the data contents.

Caution **Be sure to clear bits 3, 6, and 11 to “0”. (Also clear bit 5 of the SCR01 or SCR03 register to 0, as well as bit 1 of the SCR02 or SCR03 registers.). Be sure to set bit 2 to “1”.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00)

14.3.5 Higher 7 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02 and SDR03 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 and SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, 0 is always read.

Reset signal generation clears the SDRmn register to 0000H.

Figure 14 - 10 Format of Serial data register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock set by dividing the operating clock
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

(**Cautions** and **Remarks** are listed on the next page.)

Caution 1. Be sure to clear bit 8 of the SDR02 or SDR03 register to “0”.

Caution 2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^{2}C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. Do not write eight bits to the lower eight bits if operation is stopped (SEmn = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **14.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

14.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 14 - 11 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn Note	PEC Tmn	OVC Tmn

FEC Tmn Note	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01 and SIR03 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 or SIR02 register) to “0”.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

14.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 14 - 12 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn Note 3	BFF mn Note 3	0	0	FEF mn Note 1	PEF mn Note 2	OVF mn Note 2

TSF mn Note 3	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn Note 3	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note 1. The SSR01 and SSR03 registers only.

Note 2. The SSR00, SSR01, and SSR03 registers only.

Note 3. The SSR00 and SSR02 registers only.

Caution **If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 14 - 13 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn Note 3	BFF mn Note 3	0	0	FEF mn Note 1	PEF mn Note 2	OVF mn Note 2

FEF mn Note 1	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEF mn Note 2	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I²C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during I²C transmission (ACK is not detected).	

OVF mn Note 2	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in simplified SPI (CSI) mode.	

Note 1. The SSR01 and SSR03 registers only.
Note 2. The SSR00, SSR01, and SSR03 registers only.
Note 3. The SSR00 and SSR02 registers only.

Caution **When simplified SPI (CSI) reception proceeds in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

14.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 14 - 14 Format of Serial channel start register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution 1. Be sure to clear bits 15 to 4 of the SS0 register to “0”.

Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more fMCK clocks have elapsed.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. When the SSm register is read, 0000H is always read.

14.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 14 - 15 Format of Serial channel stop register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation [Note].

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution **Be sure to clear bits 15 to 4 of the ST0 register to "0".**

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. When the STm register is read, 0000H is always read.

14.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 14 - 16 Format of Serial channel enable status register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

14.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 14 - 17 Format of Serial output enable register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 02	0	SOE 00

SOE mn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution **Be sure to clear bits 15 to 3 and 1 of the SOE0 register to “0”.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

14.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use a pin for the serial interface as a port function pin other than a serial interface function pin, set the corresponding the CKOmn and SOmn bits to 1.

The SOm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0F0FH.

Figure 14 - 18 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	1	1	CKO 00	0	0	0	0	1	SO 02	1	SO 00

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution **Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register to “0”.**
Be sure to set bits 11 to 9, 3, and 1 of the SO0 register to “1”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

14.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the simplified SPI (CSI) mode and simplified I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 14 - 19 Format of Serial output level register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

SOL mn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

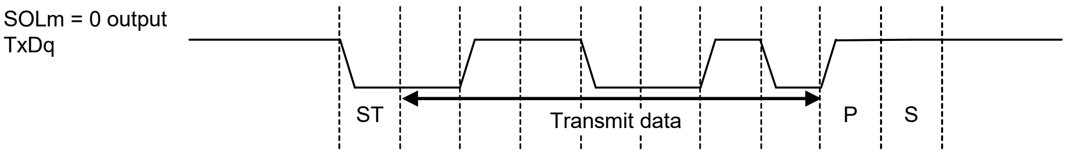
Caution Be sure to clear bits 15 to 3, and 1 of the SOL0 register to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

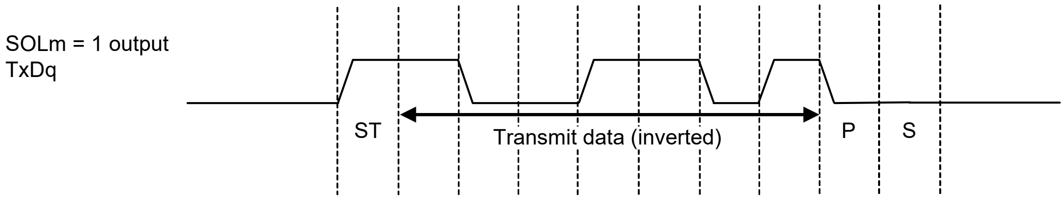
Figure 14 - 20 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 14 - 20 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

14.3.14 Serial standby control register m (SSCm)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSCm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.

Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00: Up to 1 Mbps**
- **When using UART0: 4800 bps only**
  (Can be used when FRQSEL4 in the option byte (000C2H) is set to 0.)

Figure 14 - 21 Format of Serial standby control register m (SSCm)

Address: F0138H, F0139H (SSC0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSECm	SWCm

SSECm	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0)
1	Disable the generation of error interrupts (INTSRE0)
• The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0. • Setting SSECm, SWCm = 1, 0 is prohibited.	

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and simplified SPI (CSI) or UART reception is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.	

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Figure 14 - 22 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSRx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

14.3.15 Input switch control register (ISC)

The SSIE0 bit controls the $\overline{\text{SSI00}}$ pin input of channel 0 during CSI00 communication and in slave mode.

While a high level is being input to the $\overline{\text{SSI00}}$ pin, no transmission/reception operation is performed even if a serial clock is input. While a low level is being input to the $\overline{\text{SSI00}}$ pin, a transmission/reception operation is performed according to each mode setting if a serial clock is input.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ISC register to 00H.

Figure 14 - 23 Format of Input switch control register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	0	0

SSIE00	Channel 0 $\overline{\text{SSI00}}$ input setting in CSI00 communication and slave mode
0	Disables $\overline{\text{SSI00}}$ pin input.
1	Enables $\overline{\text{SSI00}}$ pin input.

Caution Be sure to clear bits 6 to 0 to “0”.

14.3.16 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for simplified SPI (CSI) or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral hardware clock (f_{CLK}) is synchronized with 2-clock match detection. When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}) Note.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

Figure 14 - 24 Format of Noise filter enable register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	SNFEN10	0	SNFEN00

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution **Be sure to clear bits 7 to 3 and 1 to "0".**

14.3.17 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, **4.3.5 Port output mode registers (POMxx)**, and **4.3.6 Port mode control registers 0, 12, 14 (PMCxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P00/TI00/TxD1/CMP0P/ANI17/(TRJO0)) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMCxx) and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (2.5 V or 3 V), see **4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers**.

Example When P00/TI00/TxD1/CMP0P/ANI17/(TRJO0) is to be used for serial data output
Set the PMC00 bit of port mode control register 0 to 0.
Set the PM00 bit of port mode register 0 to 0.
Set the P00 bit of port register 0 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P01/TO00/RxD1/PGAI/ANI16/TRJIO0) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1.

When connecting an external device operating on a different potential (2.5 V or 3 V), see **4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers**.

Example When P01/TO00/RxD1/PGAI/ANI16/TRJIO0 is to be used for serial data input
Set the PMC01 bit of port mode control register 0 to 0.
Set the PM01 bit of port mode register 0 to 1.
Set the P01 bit of port register 0 to 0 or 1.

14.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the pin for serial interface can be used as port function pins in this mode.

14.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

Figure 14 - 25 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAU0 to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	0	0	ADCEN ×	0	0	SAU0EN 0/1	0	TAU0EN ×

Control of SAU0 input clock
0: Stops supply of input clock
1: Supplies input clock

Caution 1. If SAU0EN = 0, writing to a control register of serial array unit 0 is ignored, and, even if the register is read, only the default value is read

Note that this does not apply to the following registers.

- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers 0, 3, 5 (PIM0, PIM3, PIM5)**
- **Port output mode registers 0, 3, 5 (POM0, POM3, POM5)**
- **Port mode control register 0 (PMC0)**
- **Port mode registers 0, 3, 5, 6 (PM0, PM3, PM5, PM6)**
- **Port registers 0, 3, 5, 6 (P0, P3, P5, P6)**

Caution 2. Be sure to clear the following bits to 0.

Bits 1, 3, 4, 6, 7

Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

14.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 14 - 26 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm)... This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm)... This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

1: Serial clock output value is “1”

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

(e) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

0: Disables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

14.5 Operation of Simplified SPI (CSI00) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]
- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate $^{\text{Note}}$
  During master communication: Max. $f_{CLK}/2$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only following simplified SPI (CSI) can be specified for asynchronous reception.
- CSI00

In addition, CSI00 supports the slave select input function. For details, refer to **14.6 Clock Synchronous Serial Communication with Slave Select Input Function**.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**.

The channels supporting simplified SPI (CSI00) are channel 0 of SAU0.

- 30, 32, 44-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

Simplified SPI (CSI00) performs the following seven types of communication operations.

• Master transmission	(See **14.5.1**.)
• Master reception	(See **14.5.2**.)
• Master transmission/reception	(See **14.5.3**.)
• Slave transmission	(See **14.5.4**.)
• Slave reception	(See **14.5.5**.)
• Slave transmission/reception	(See **14.5.6**.)
• SNOOZE mode function (CSI00 only)	(See **14.5.7**.)

14.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits
Transfer rate Note	Max. $f_{MCK}/2$ [Hz] (CSI00) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 27 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							
									SIOp							

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOm2 ×	0	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

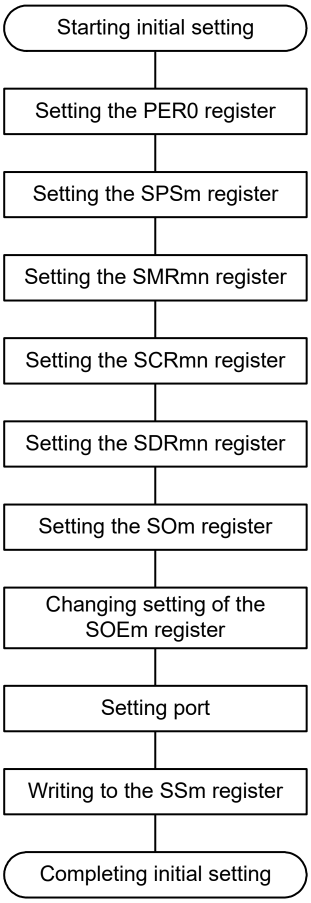
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. ☐: Setting is fixed in the simplified SPI (CSI) master transmission mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 28 Initial Setting Procedure for Master Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

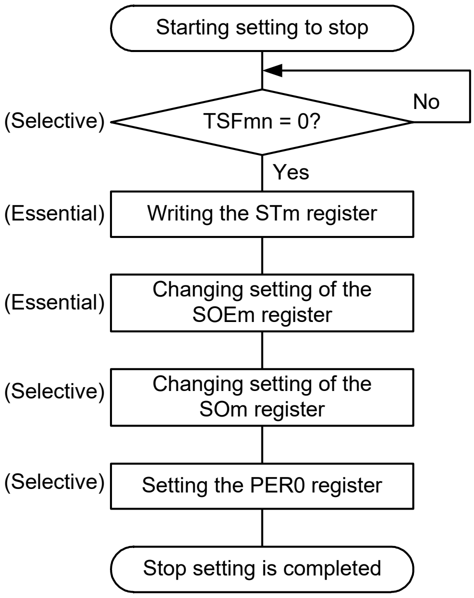
Set the SOEmn bit to 1 and enable data output of the target channel.

Setting a port register and a port mode register (Enable data output and clock output of the target channel by).

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Setting of SAU is completed.
Write transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 14 - 29 Procedure for Stopping Master Transmission



If there is any data being transferred, wait for their completion (If there is an urgent must stop, do not wait).

Write 1 to the STmn bit of the target channel. (SEmn = 0: to operation stop status)

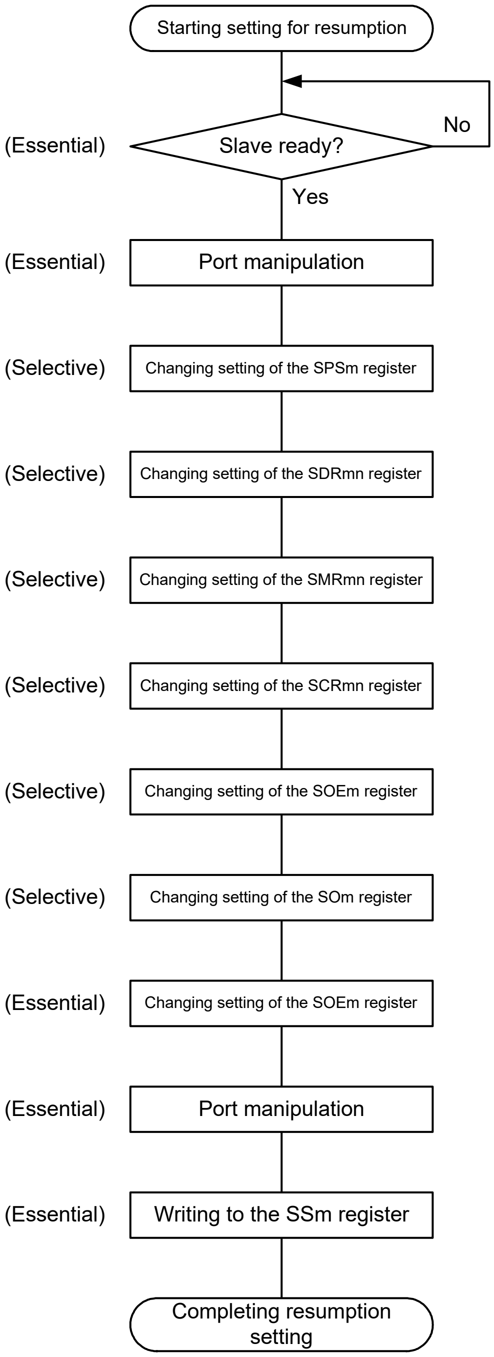
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

To use the STOP mode, reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 30 Procedure for Resuming Master Transmission



Wait until stop the communication target (slave) or communication operation completed.

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

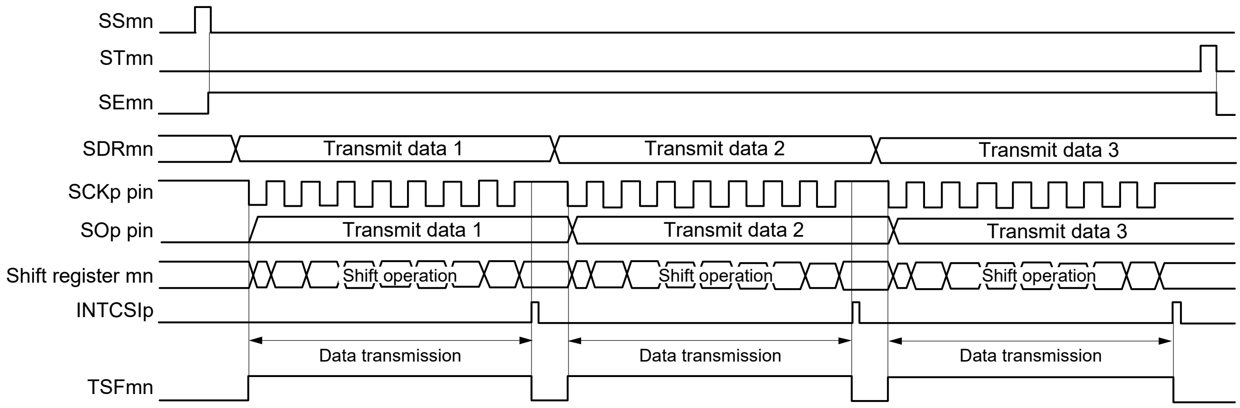
Set the SSmn bit of the target channel to 1 (SEmn = 1 : to enable operation).

Setting is completed.
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

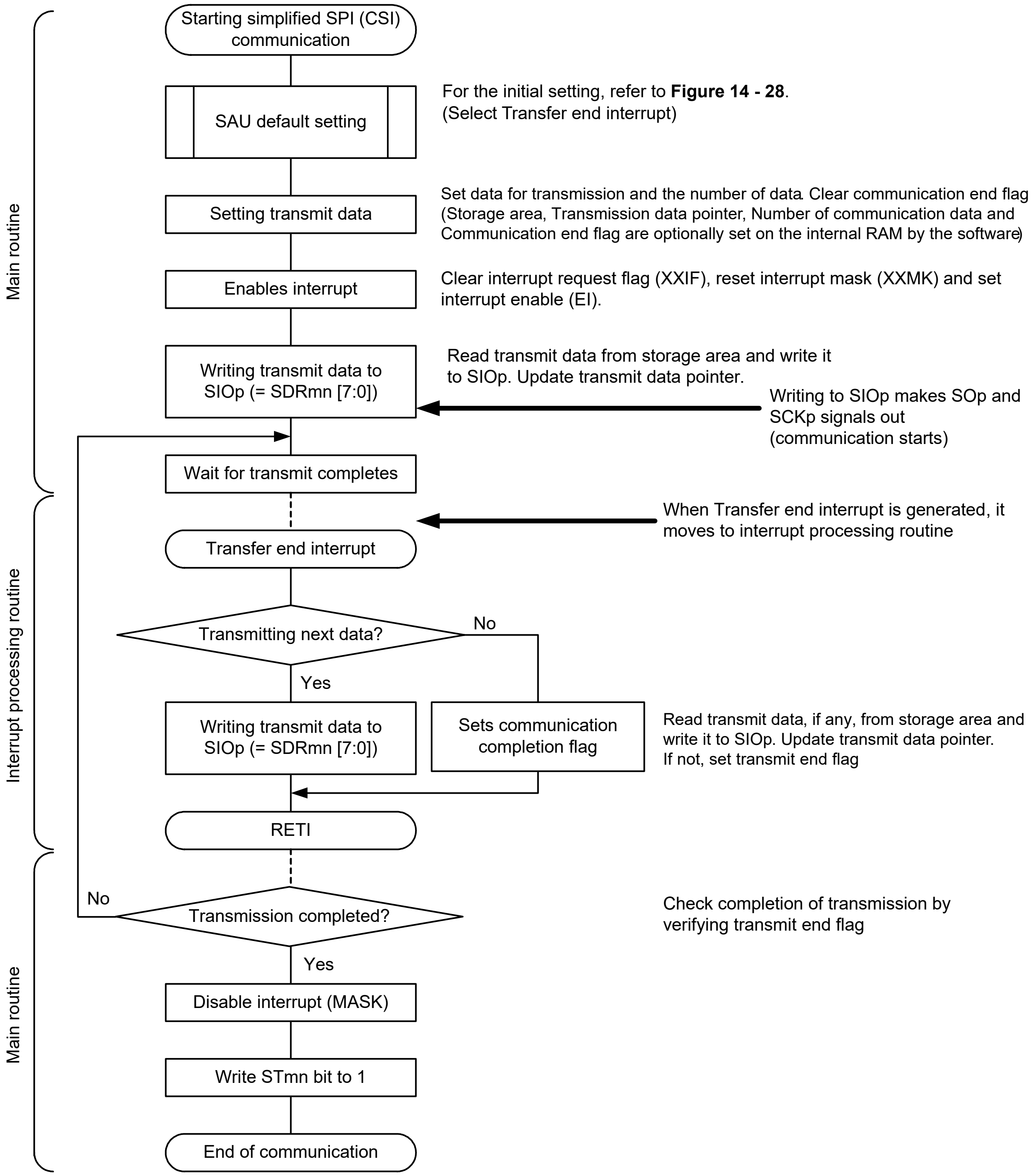
(3) Processing flow (in single-transmission mode)

Figure 14 - 31 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



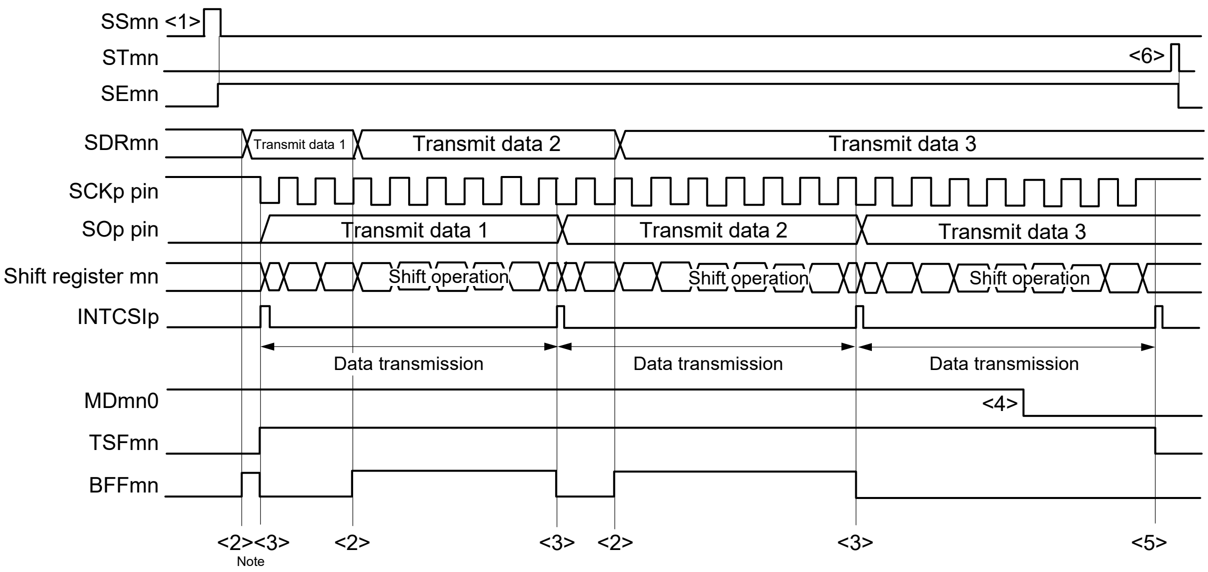
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 32 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

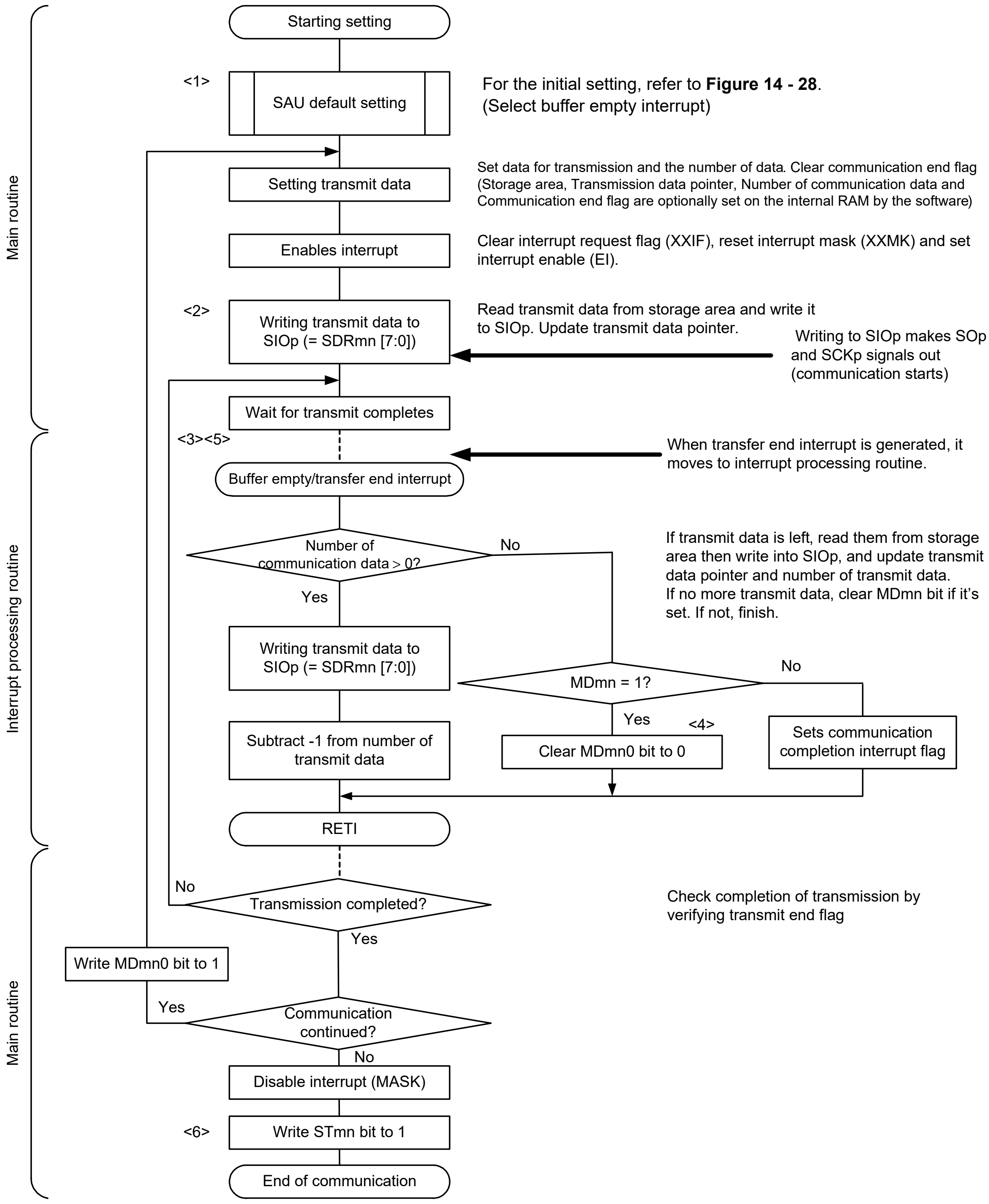


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 34 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate Note	Max. $f_{MCK}/2$ [Hz] (CSI00) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 35 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 ×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

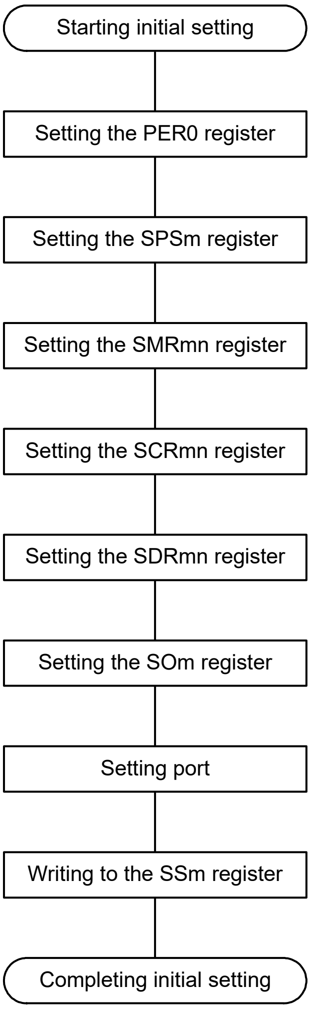
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) master reception mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 36 Initial Setting Procedure for Master Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (fMCK)).

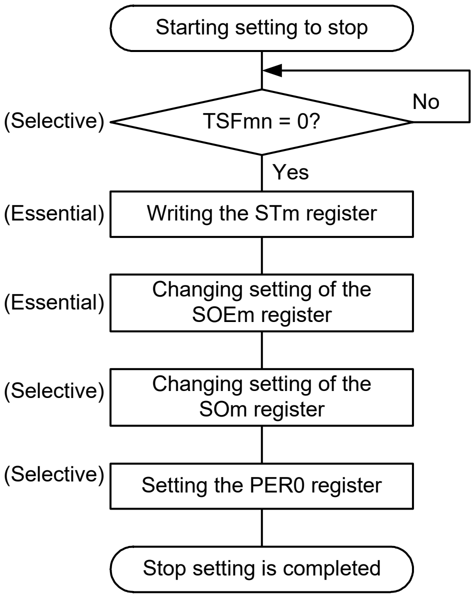
Set the initial output level of the serial clock (CKOmn).

Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 14 - 37 Procedure for Stopping Master Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

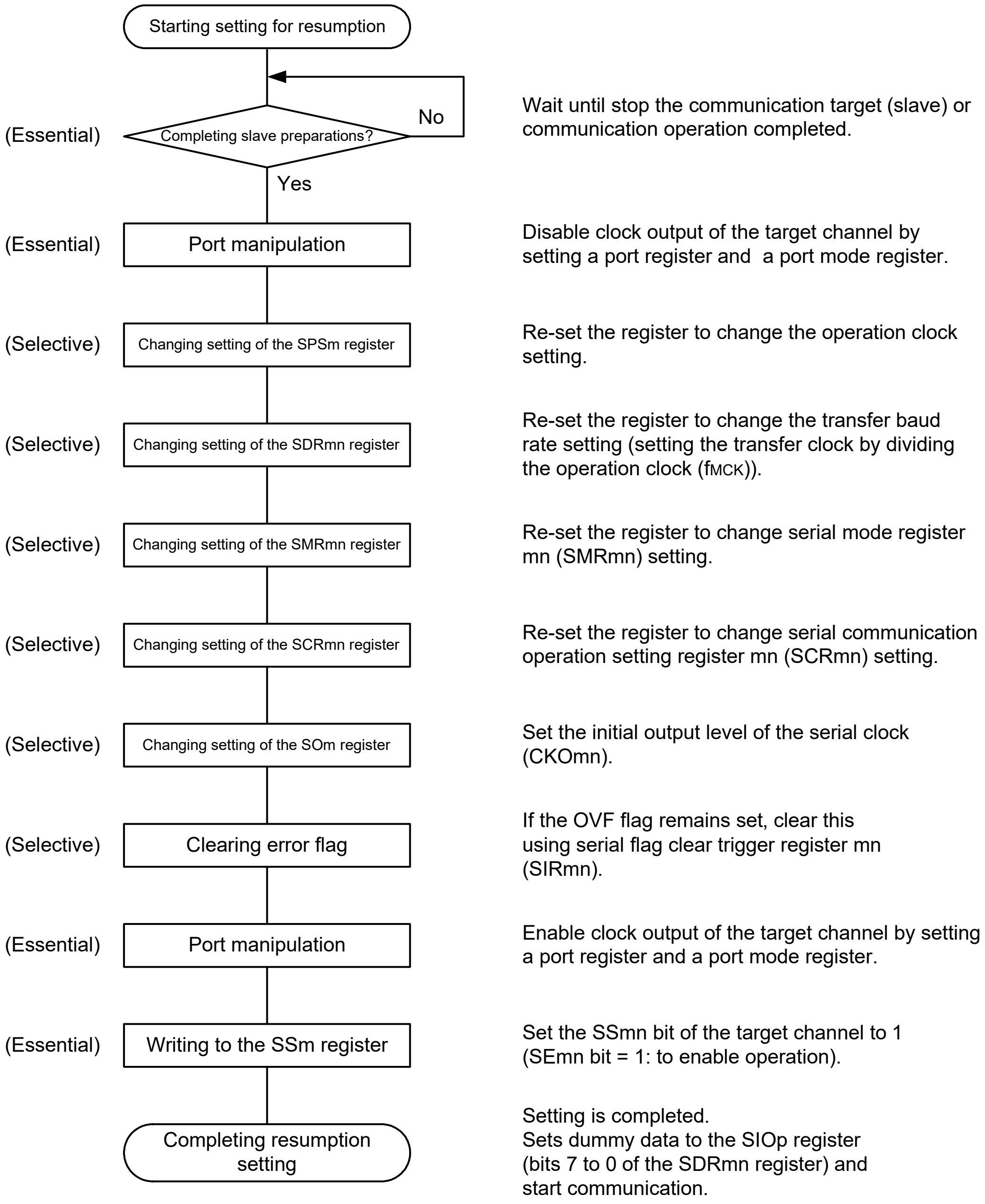
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

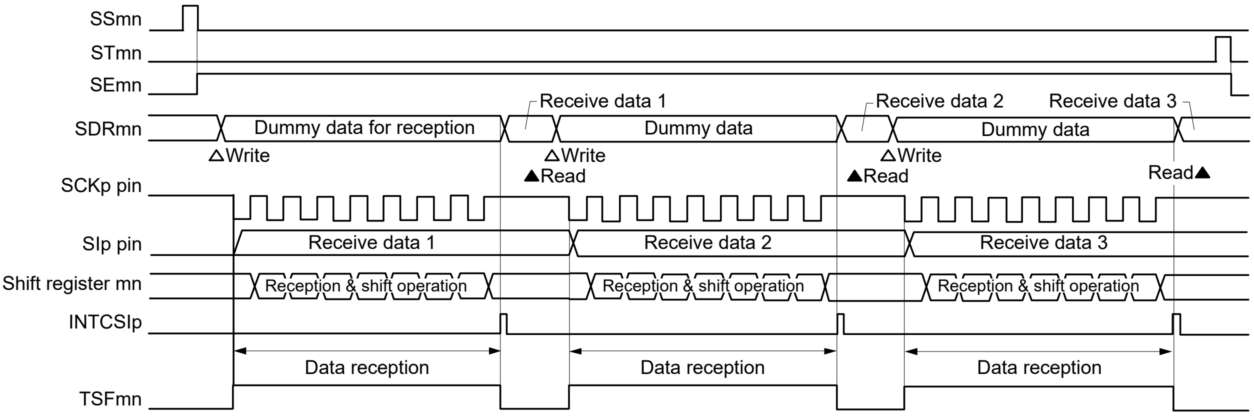
Figure 14 - 38 Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

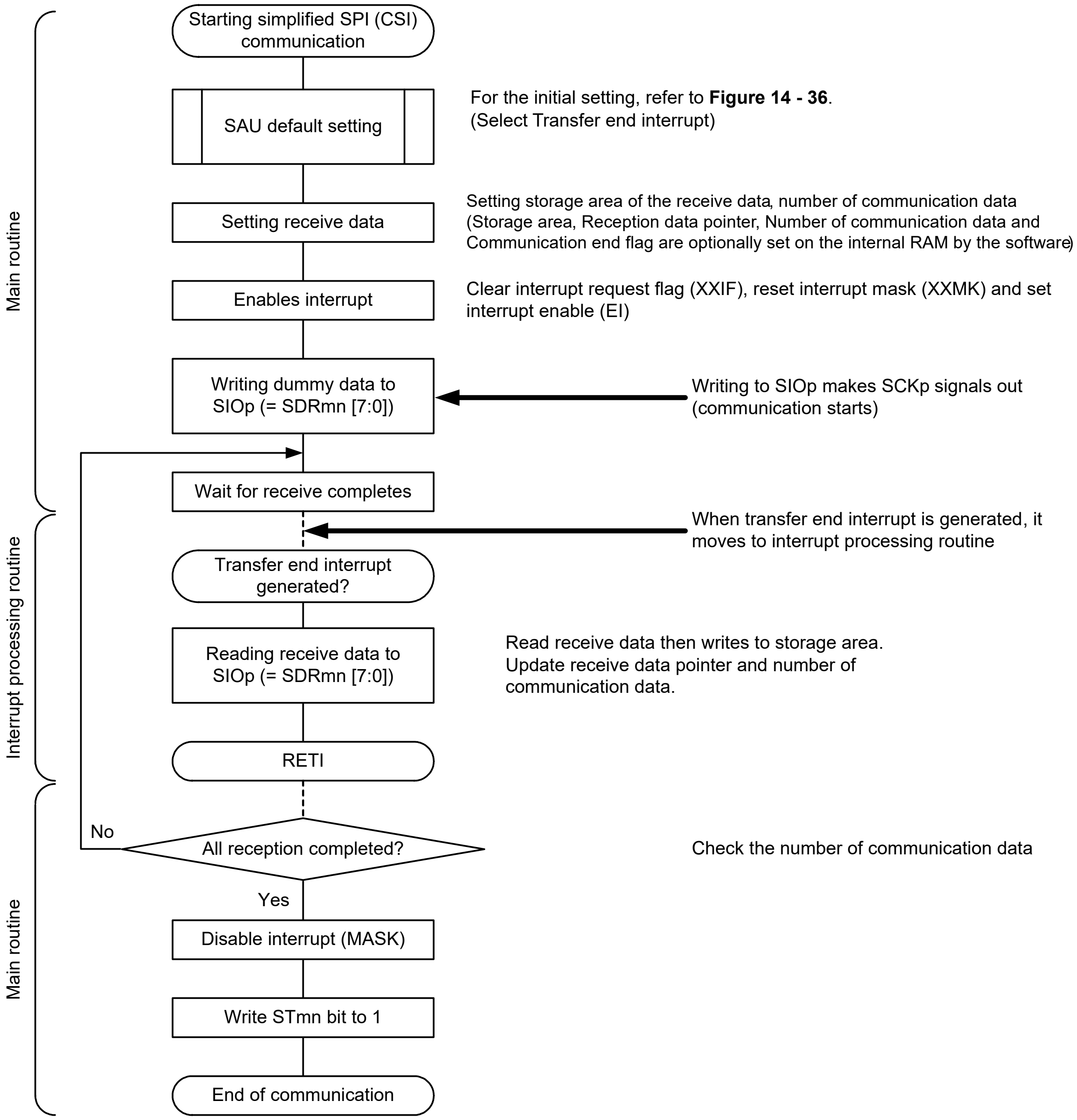
(3) Processing flow (in single-reception mode)

Figure 14 - 39 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



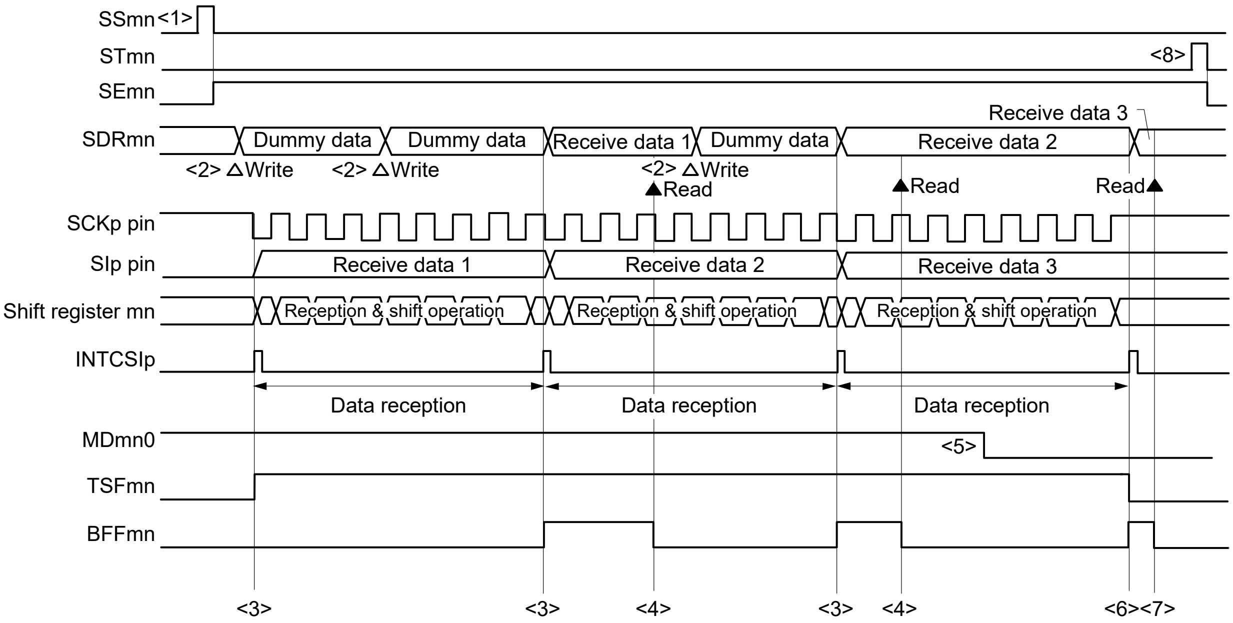
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 40 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 14 - 41 Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

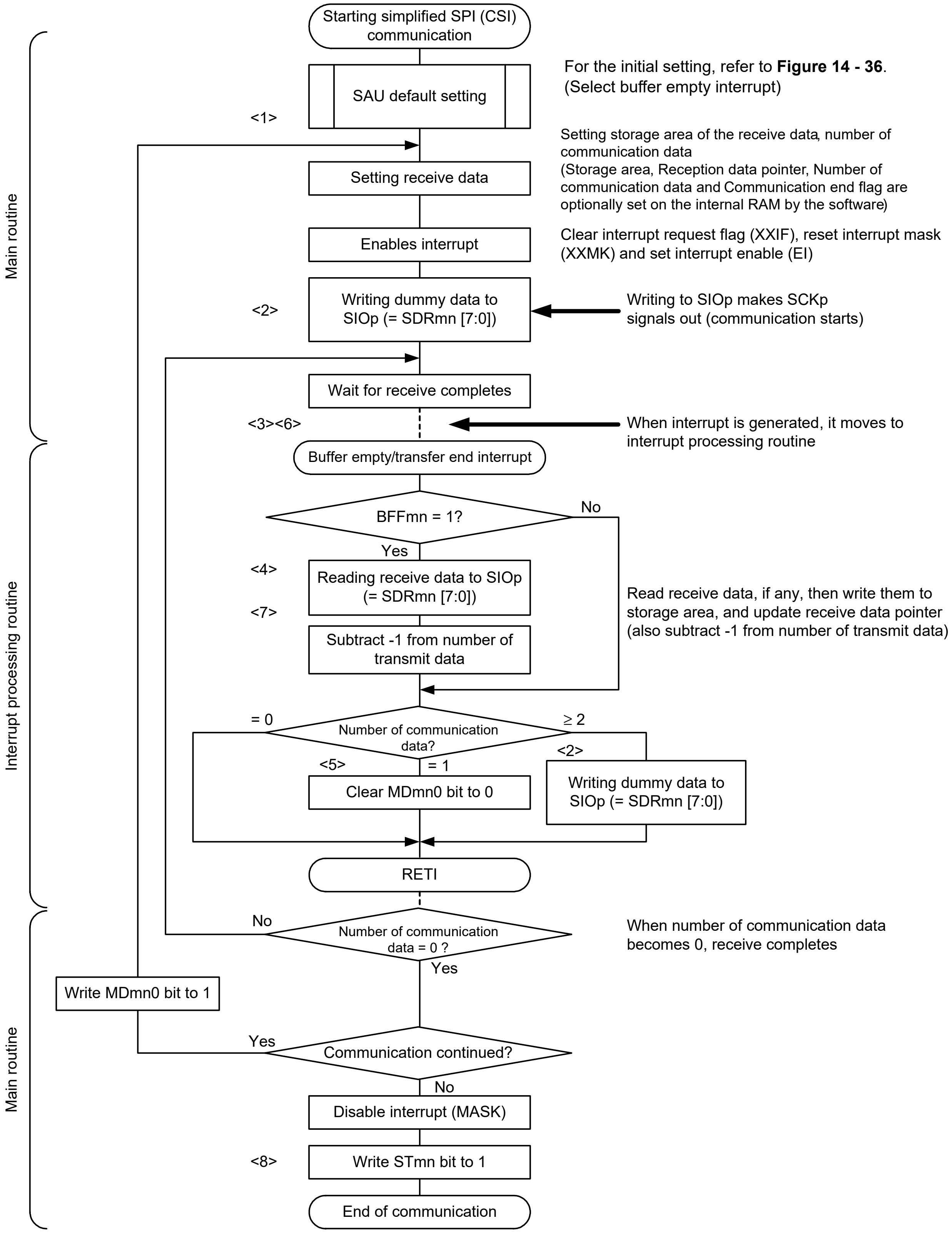


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 42 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 42 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 41 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate Note	Max. $f_{MCK}/2$ [Hz] (CSI00) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

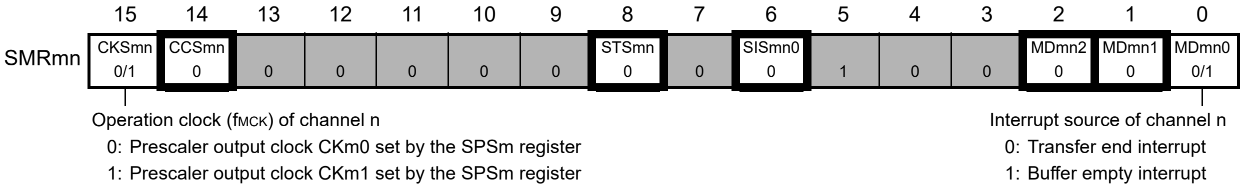
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 43 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00)

(a) Serial mode register mn (SMRmn)

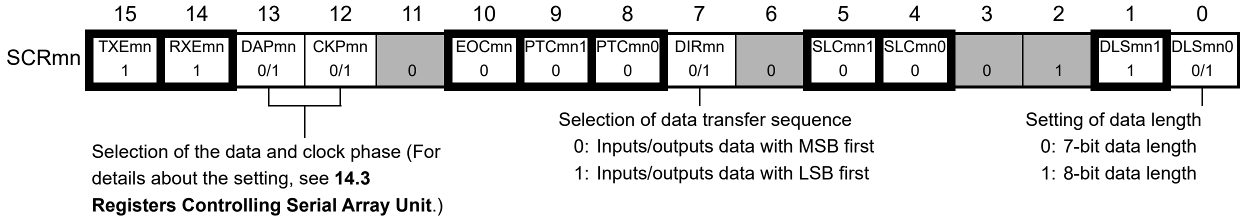


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)



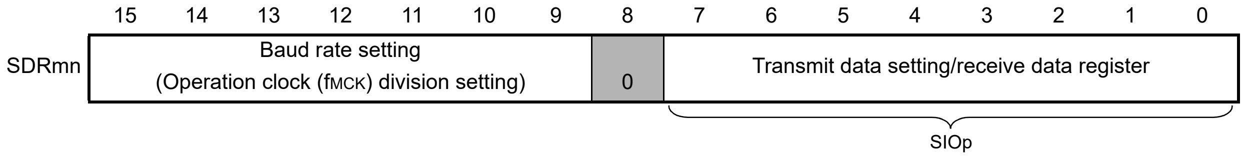
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

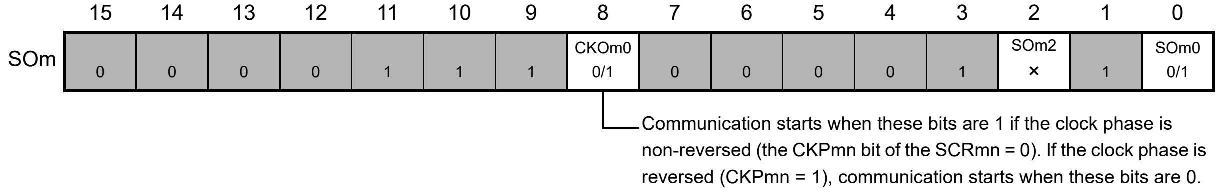
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							

SIOp

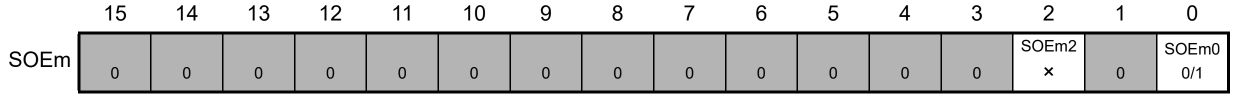
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

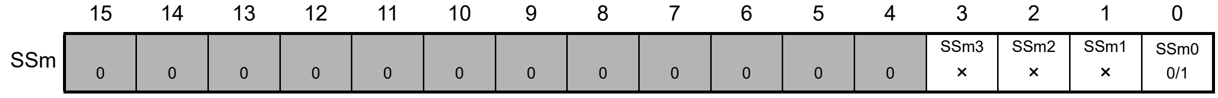
Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



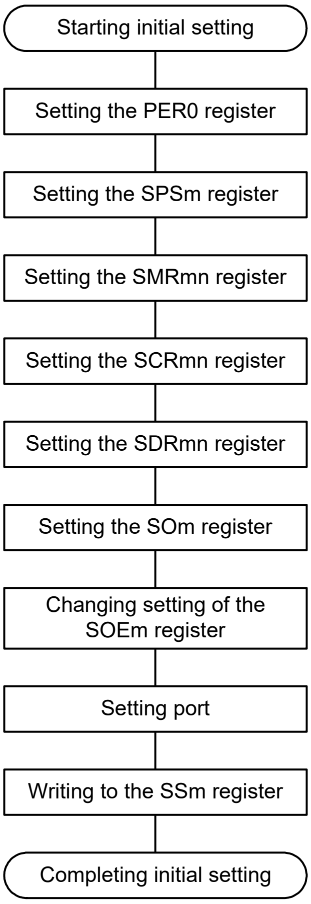
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. □: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 44 Initial Setting Procedure for Master Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

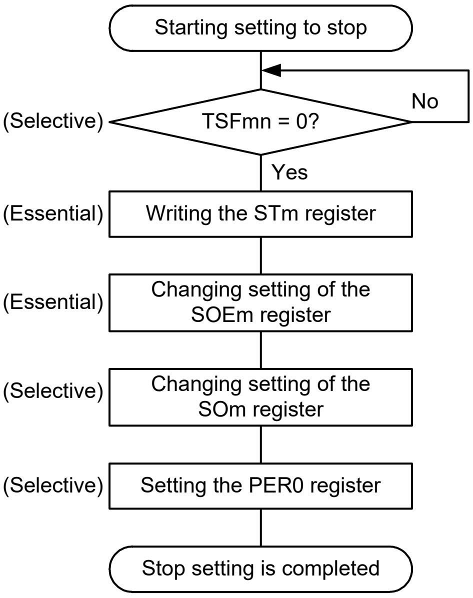
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start.

Figure 14 - 45 Procedure for Stopping Master Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

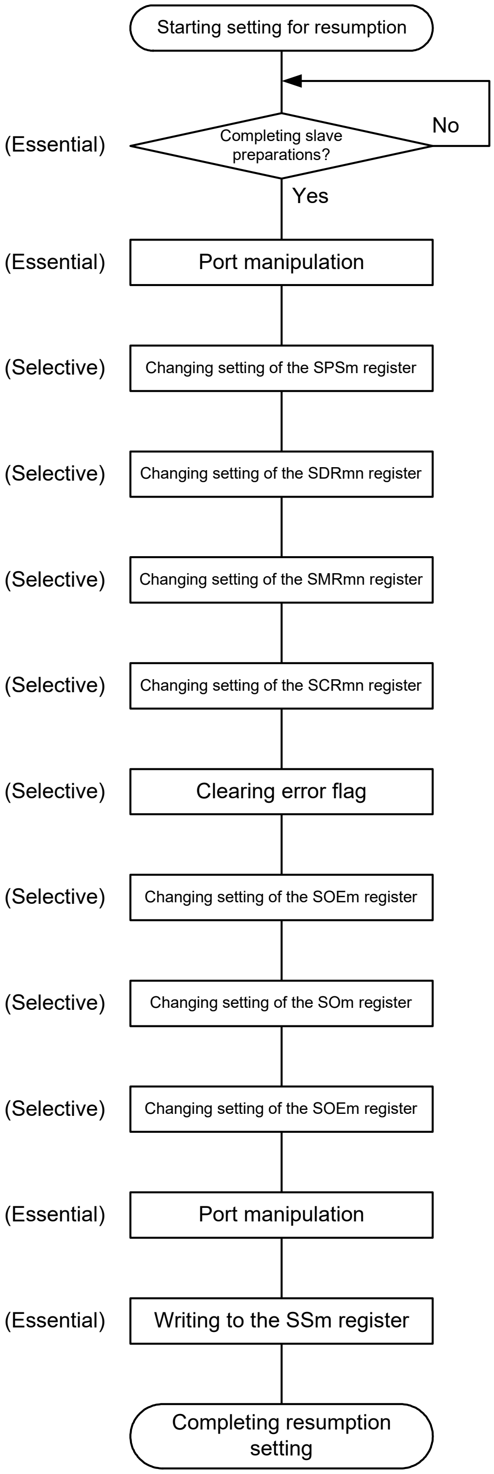
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 46 Procedure for Resuming Master Transmission/Reception



Wait until stop the communication target (slave) or communication operation completed.

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

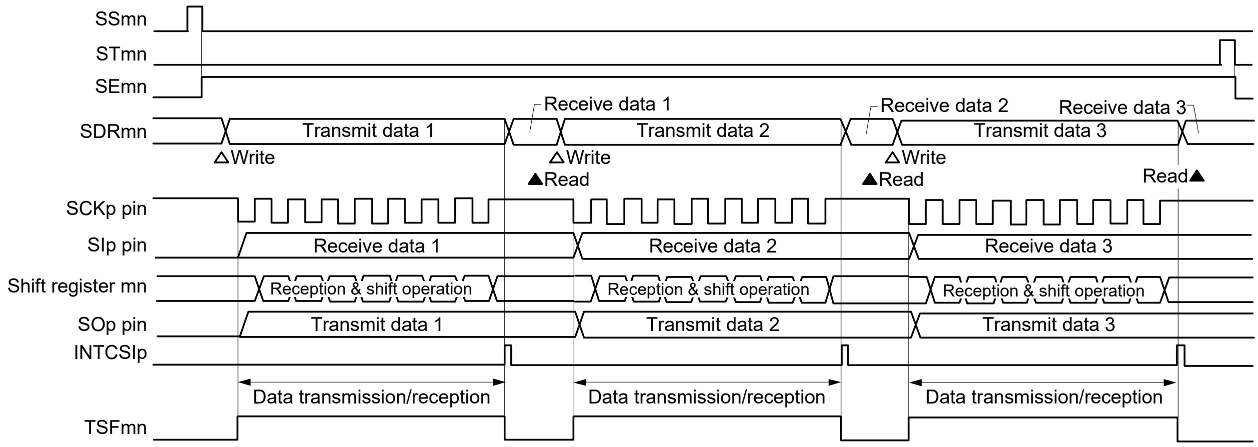
Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

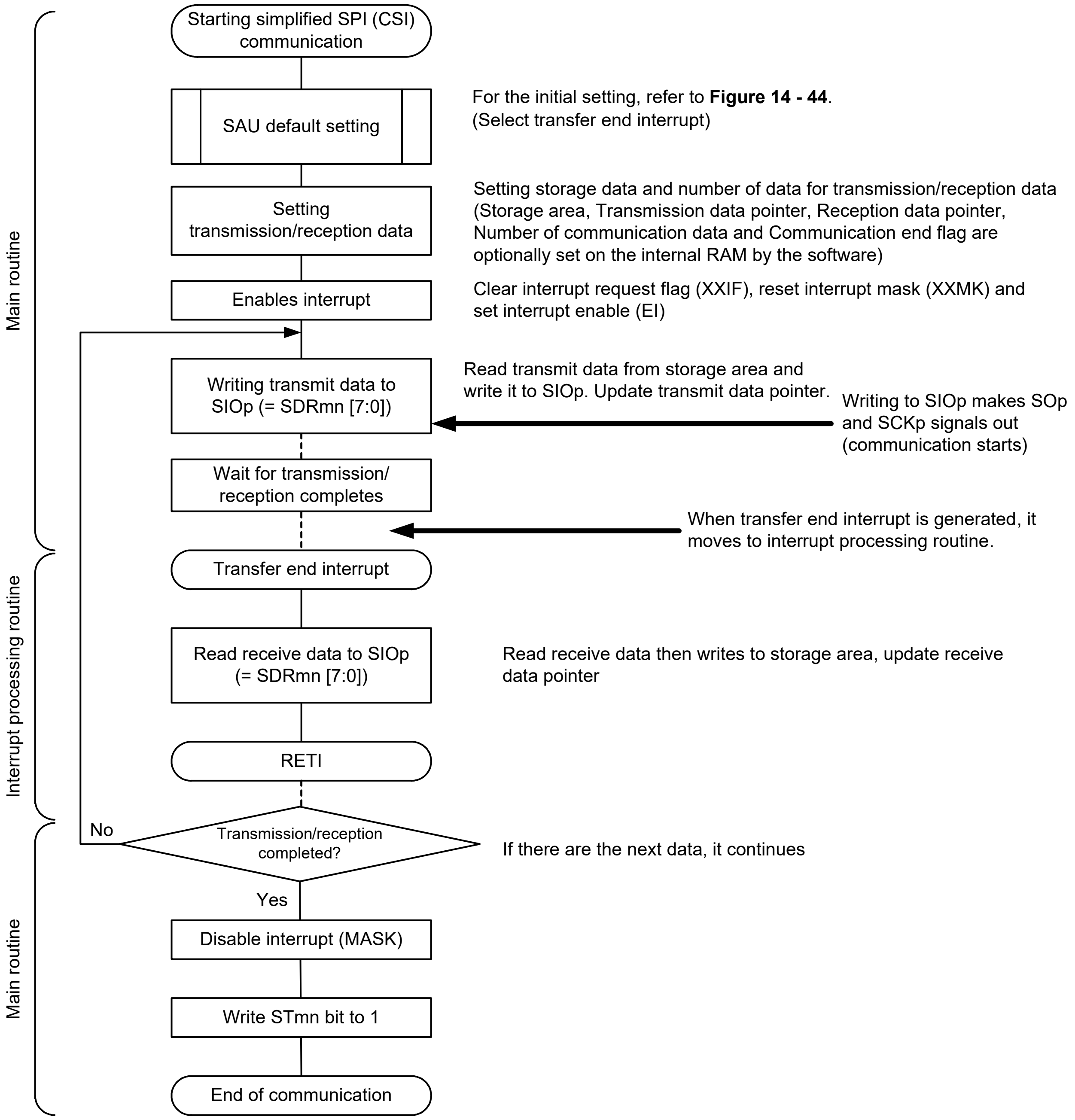
(3) Processing flow (in single-transmission/reception mode)

Figure 14 - 47 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



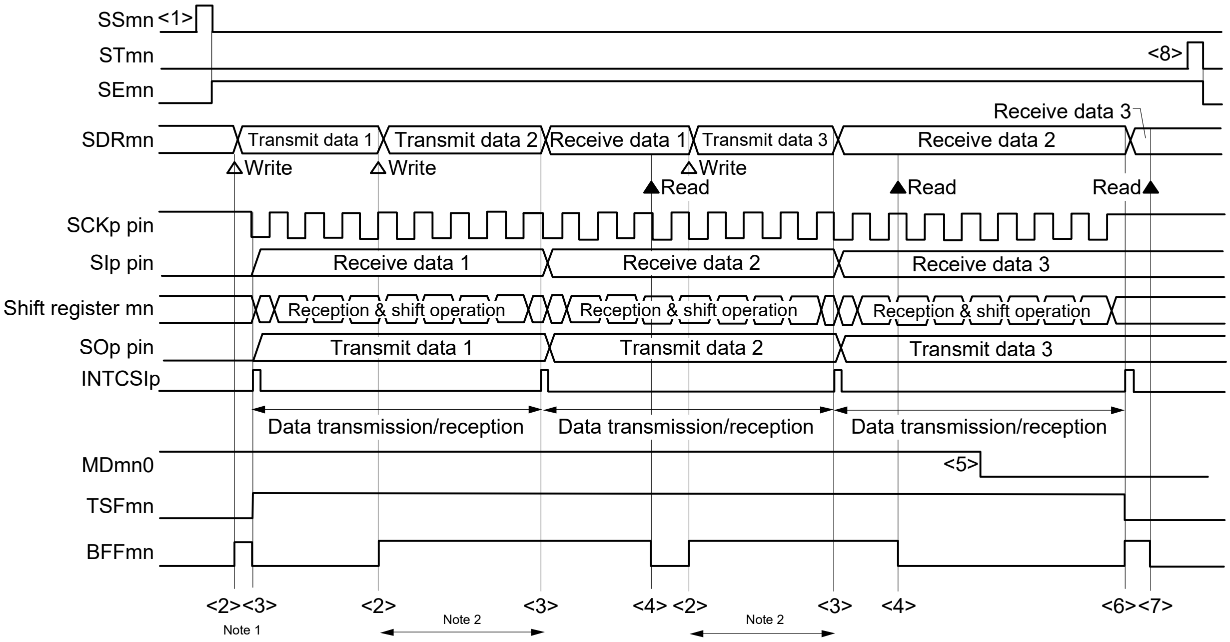
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 48 Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 14 - 49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

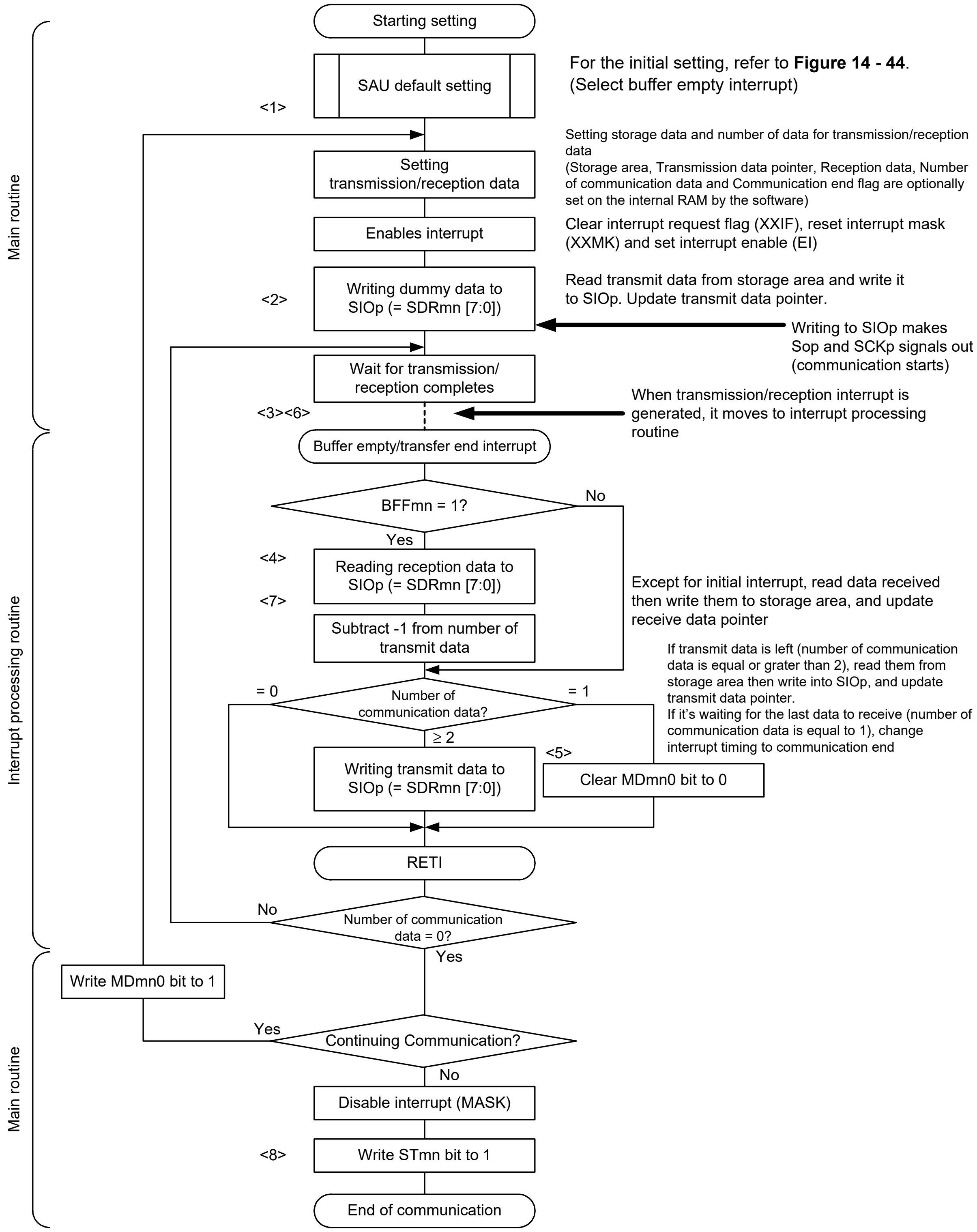
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



For the initial setting, refer to **Figure 14 - 44**.
(Select buffer empty interrupt)

Setting storage data and number of data for transmission/reception data
(Storage area, Transmission data pointer, Reception data, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Writing to SIOp makes Sop and SCKp signals out (communication starts)

When transmission/reception interrupt is generated, it moves to interrupt processing routine

Except for initial interrupt, read data received then write them to storage area, and update receive data pointer

If transmit data is left (number of communication data is equal or grater than 2), read them from storage area then write into SIOp, and update transmit data pointer.
If it's waiting for the last data to receive (number of communication data is equal to 1), change interrupt timing to communication end

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2.
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

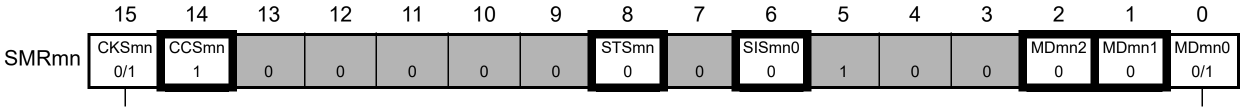
Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 51 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00)

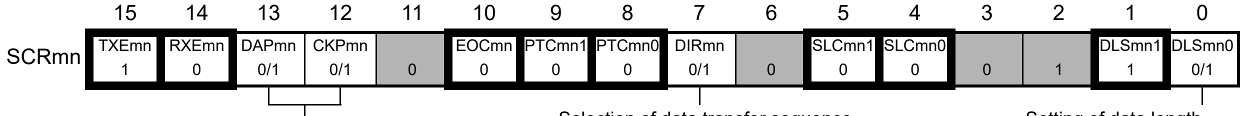
(a) Serial mode register mn (SMRmn)



Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

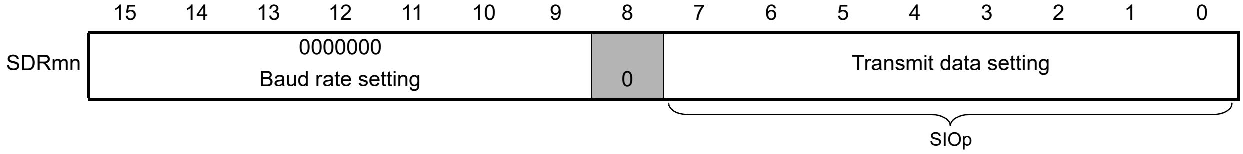


Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

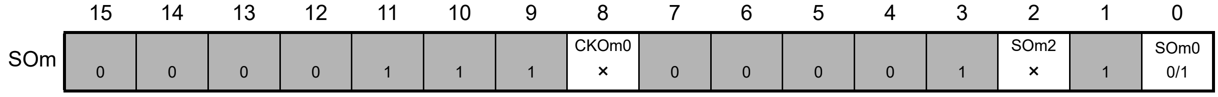
Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

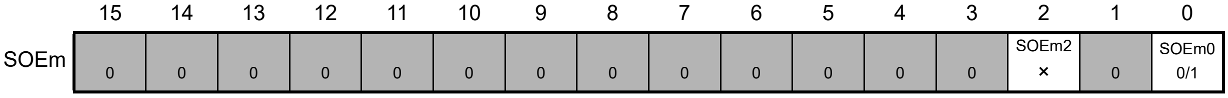
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



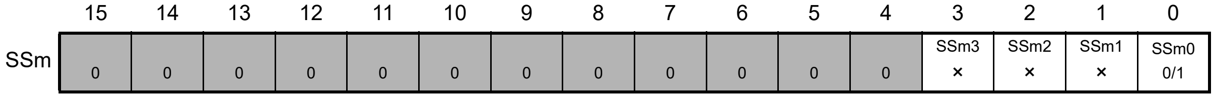
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

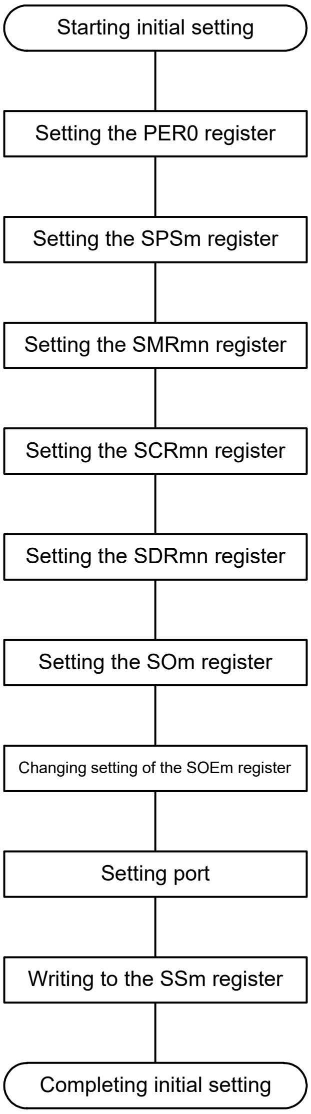


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) slave transmission mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 52 Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

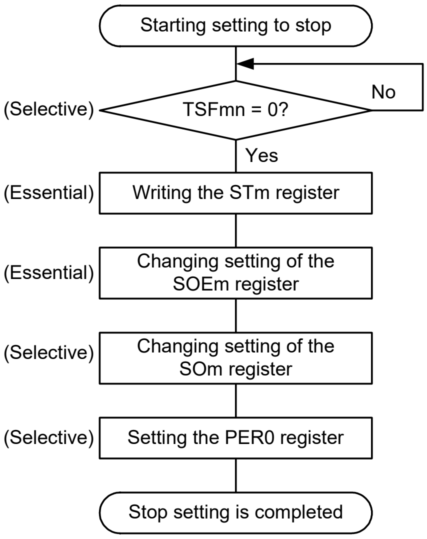
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Figure 14 - 53 Procedure for Stopping Slave Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel (SEmn = 0: to operation stop status).

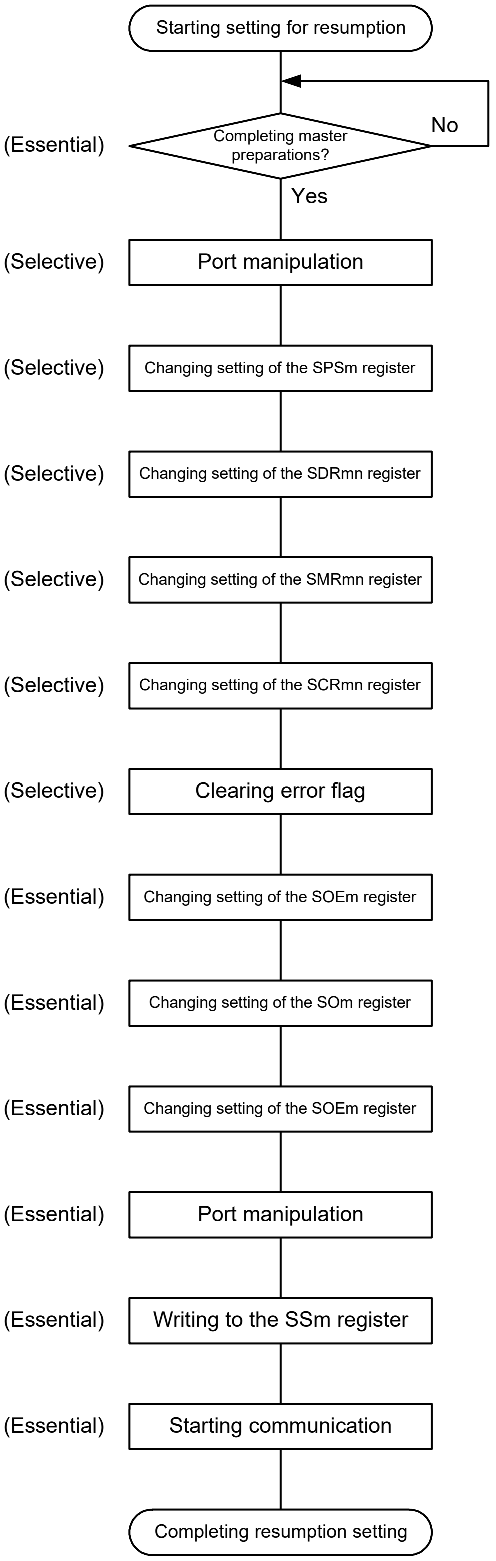
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 54 Procedure for Resuming Slave Transmission



Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

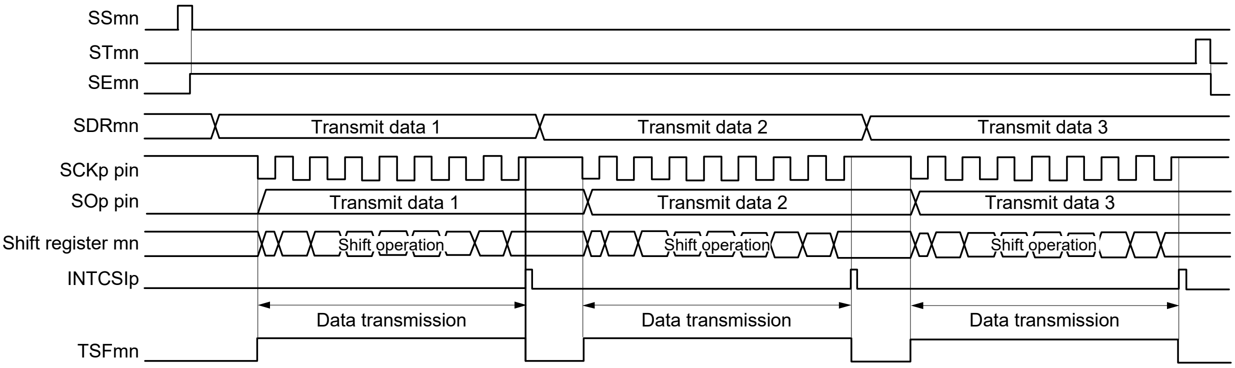
Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

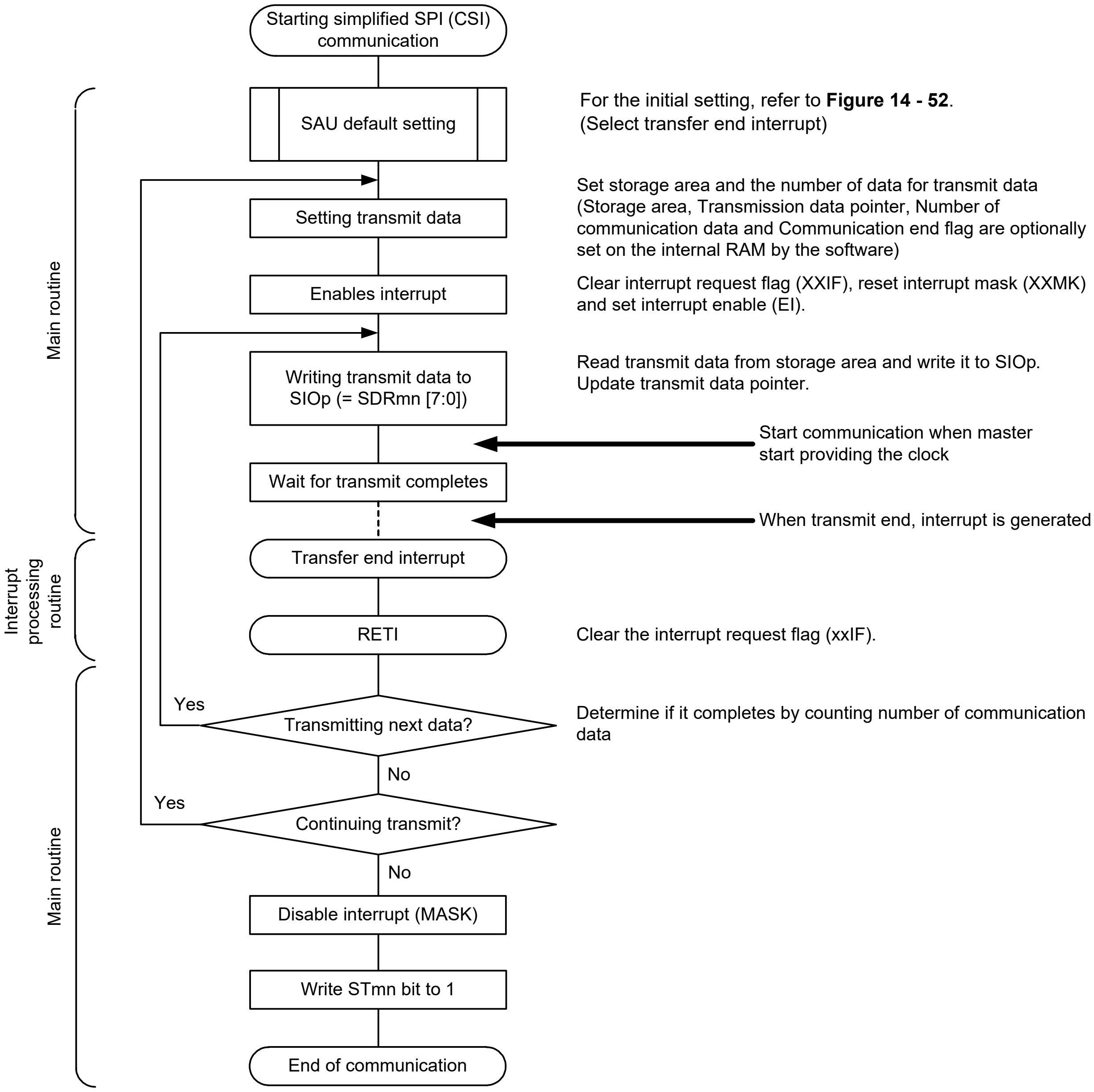
(3) Processing flow (in single-transmission mode)

Figure 14 - 55 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



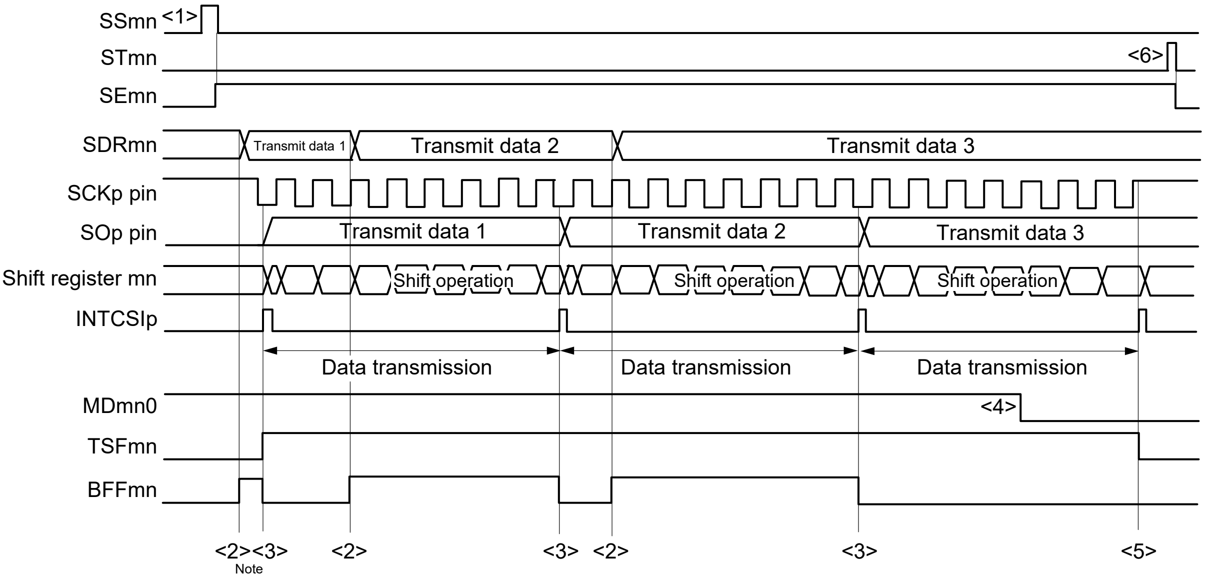
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 56 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

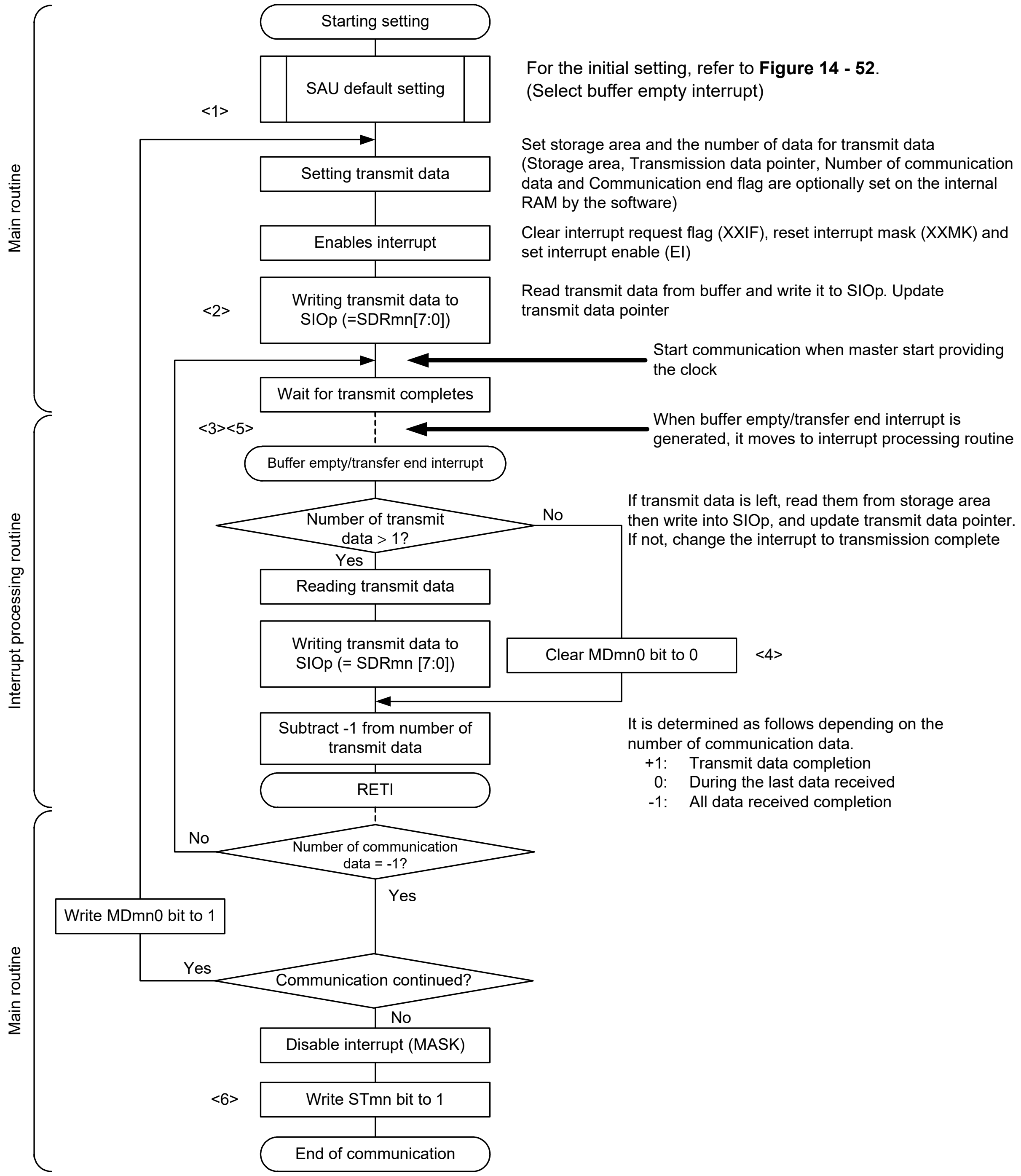


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 58 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00
Interrupt	INTCSI00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 59 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							

SIOp

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

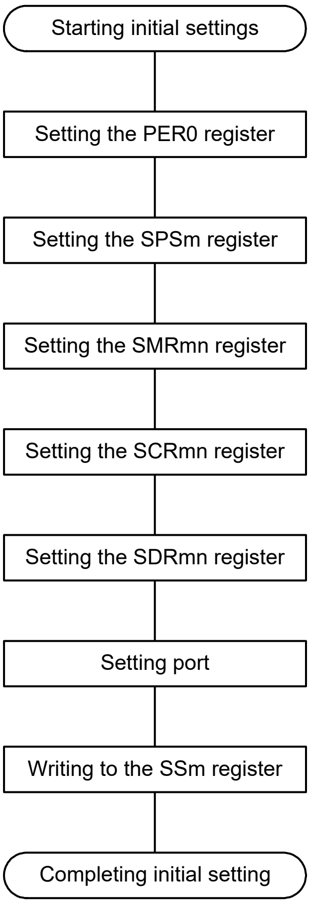
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. ☐: Setting is fixed in the simplified SPI (CSI) slave reception mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 60 Initial Setting Procedure for Slave Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

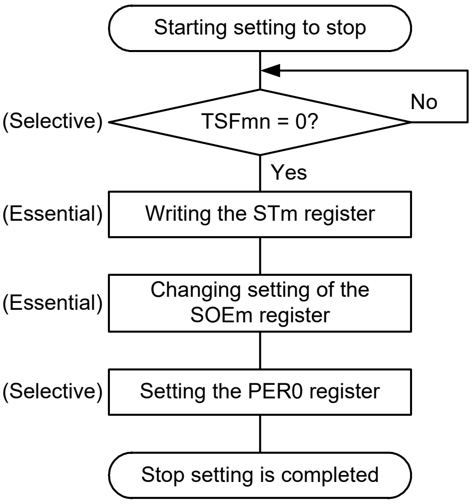
Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 14 - 61 Procedure for Stopping Slave Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

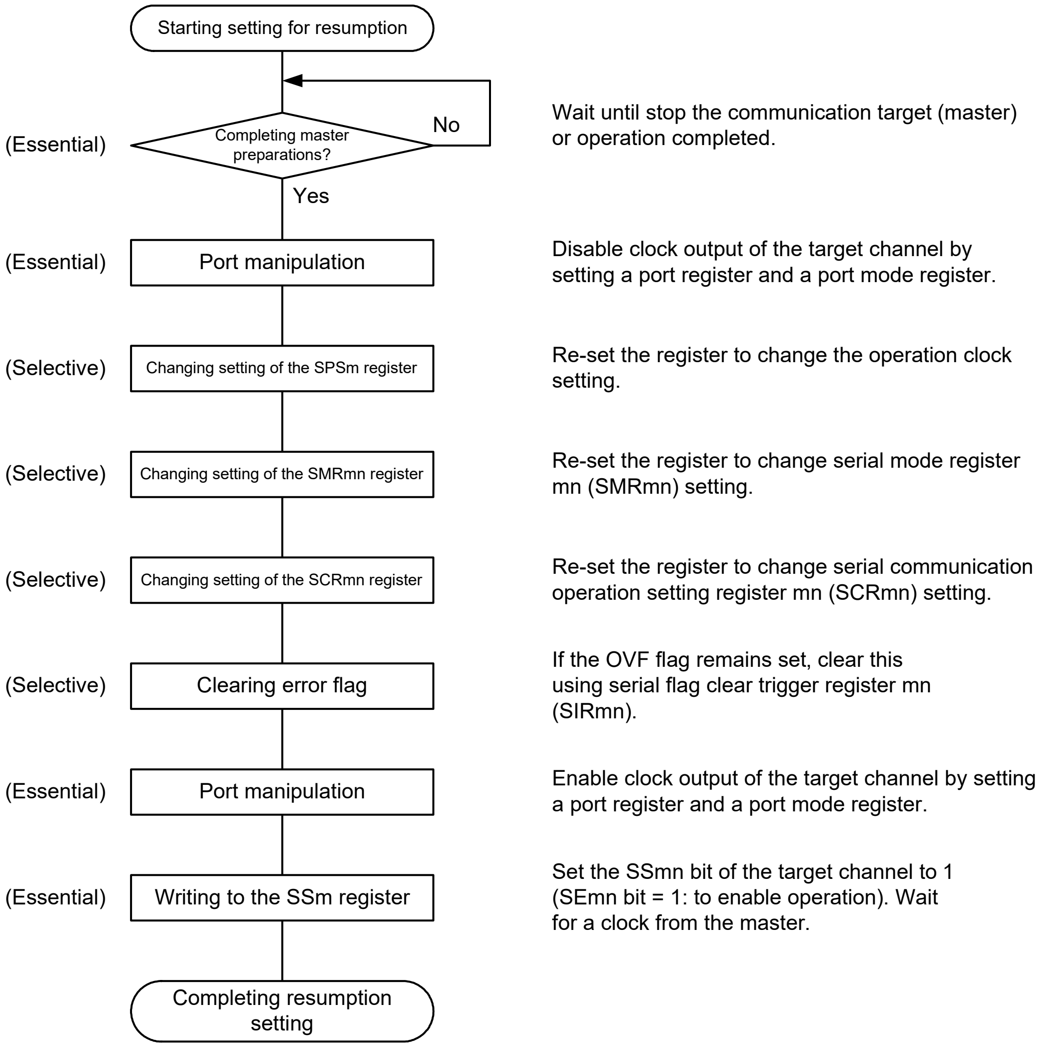
Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

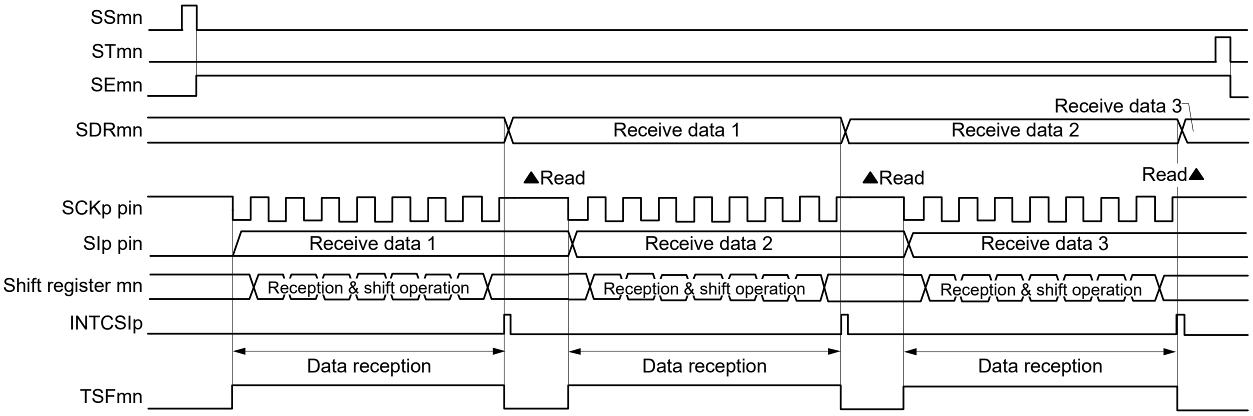
Figure 14 - 62 Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

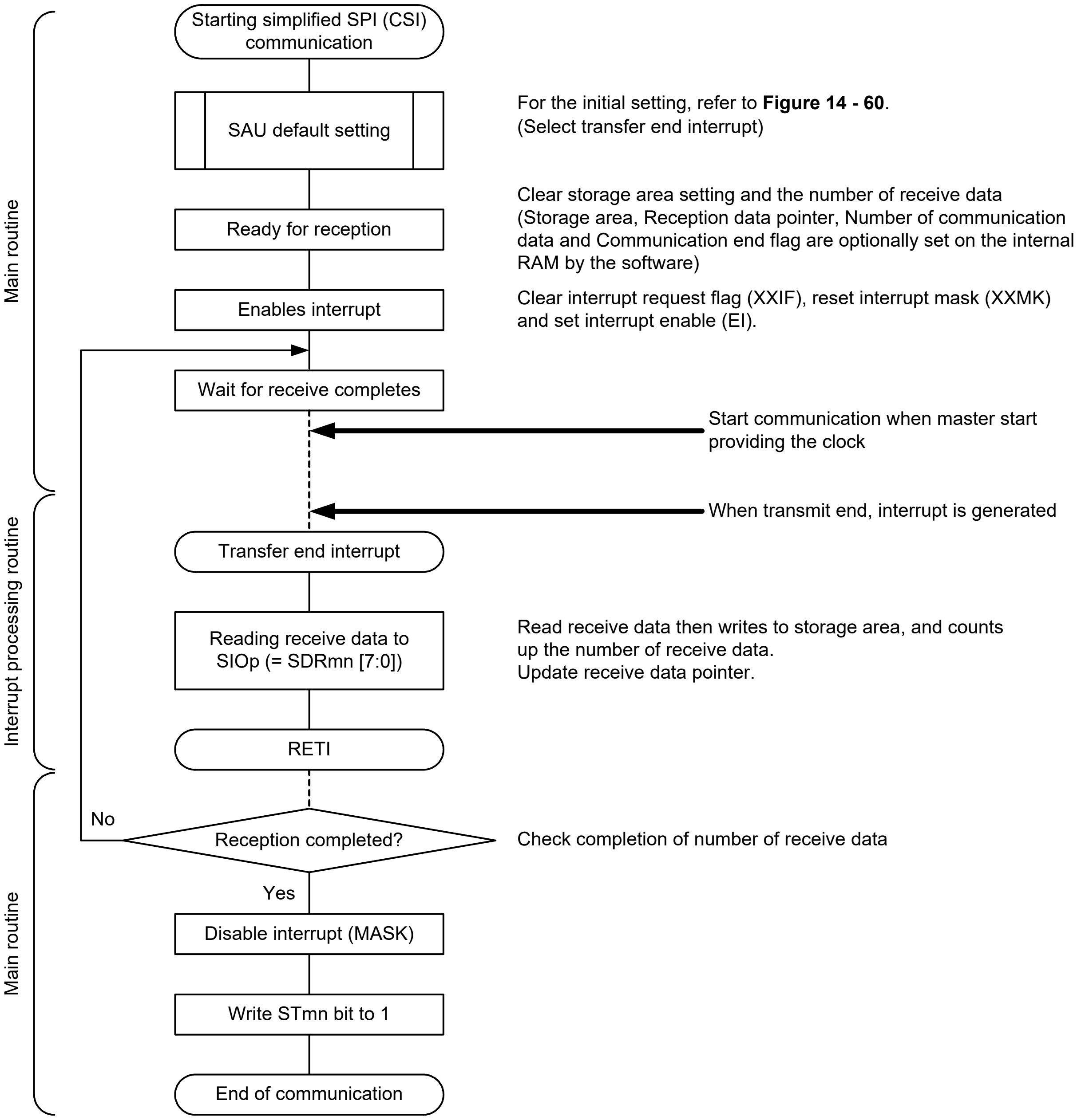
(3) Processing flow (in single-reception mode)

Figure 14 - 63 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 64 Flowchart of Slave Reception (in Single-Reception Mode)



For the initial setting, refer to **Figure 14 - 60**.
(Select transfer end interrupt)

Clear storage area setting and the number of receive data (Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read receive data then writes to storage area, and counts up the number of receive data.
Update receive data pointer.

Check completion of number of receive data

14.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2.
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 65 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

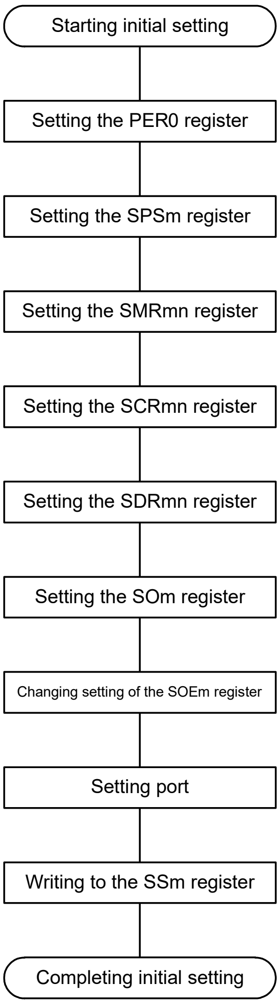
Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Remark 2. □: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 66 Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

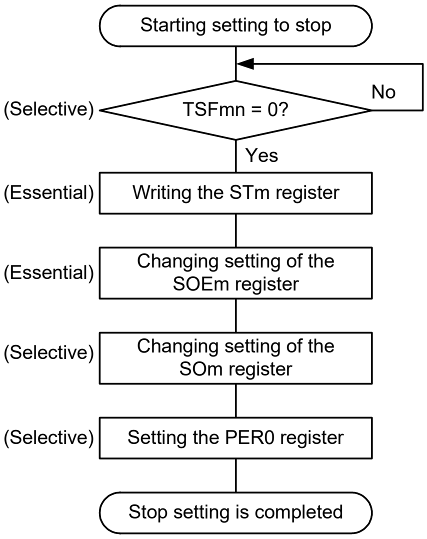
Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Figure 14 - 67 Procedure for Stopping Slave Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

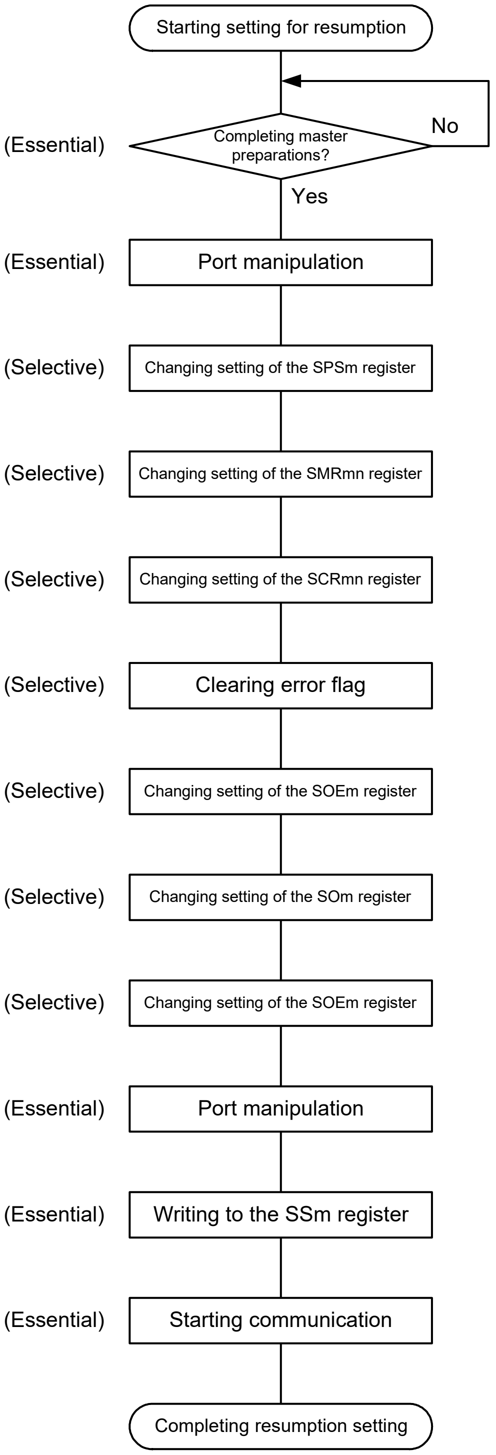
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 68 Procedure for Resuming Slave Transmission/Reception



Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

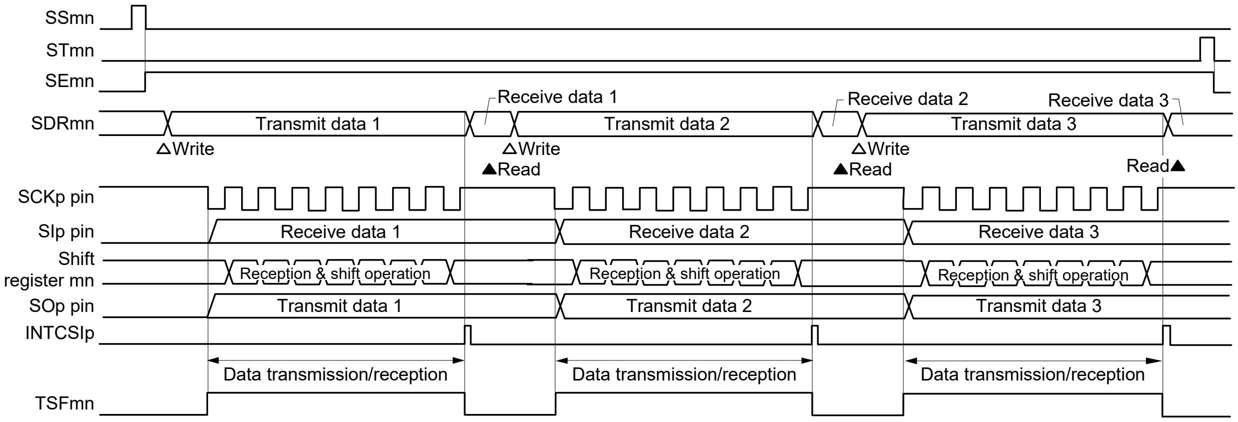
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

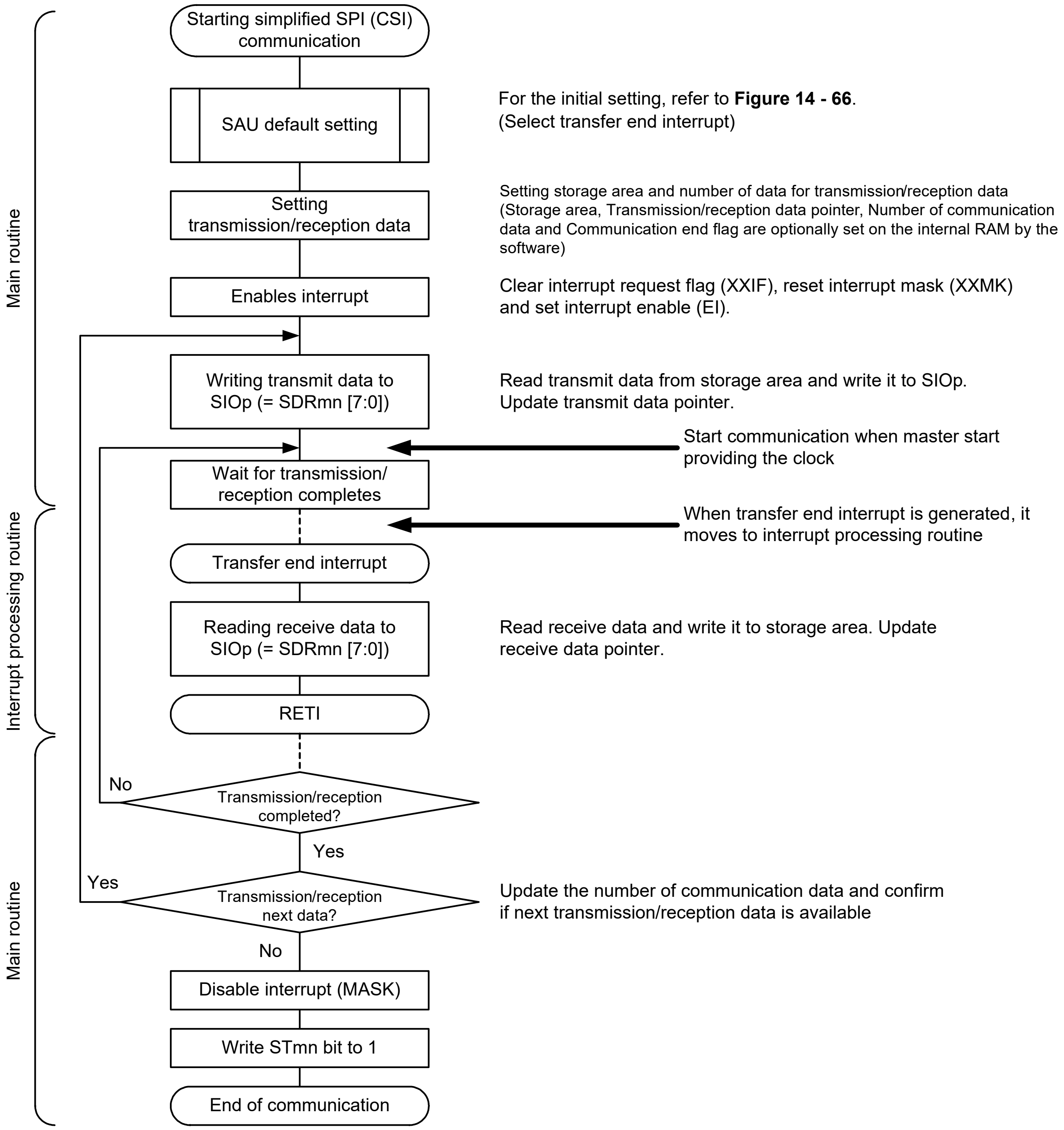
(3) Processing flow (in single-transmission/reception mode)

Figure 14 - 69 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 70 Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



For the initial setting, refer to **Figure 14 - 66**.
(Select transfer end interrupt)

Setting storage area and number of data for transmission/reception data (Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

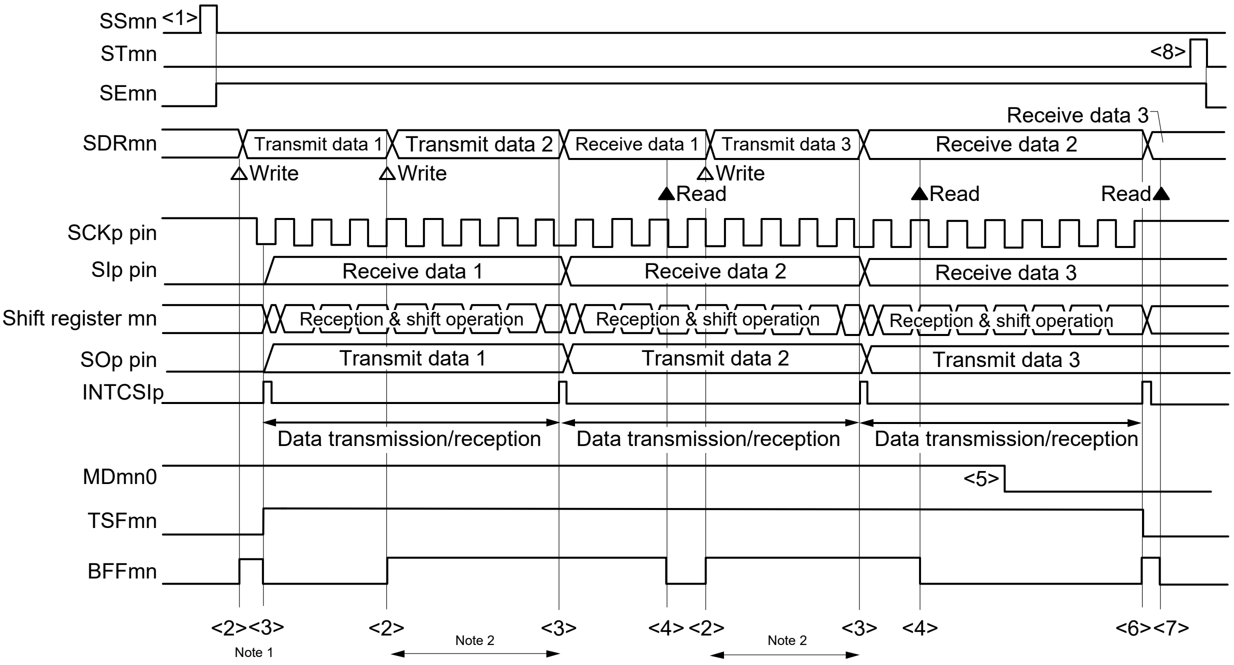
Read receive data and write it to storage area. Update receive data pointer.

Update the number of communication data and confirm if next transmission/reception data is available

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 14 - 71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

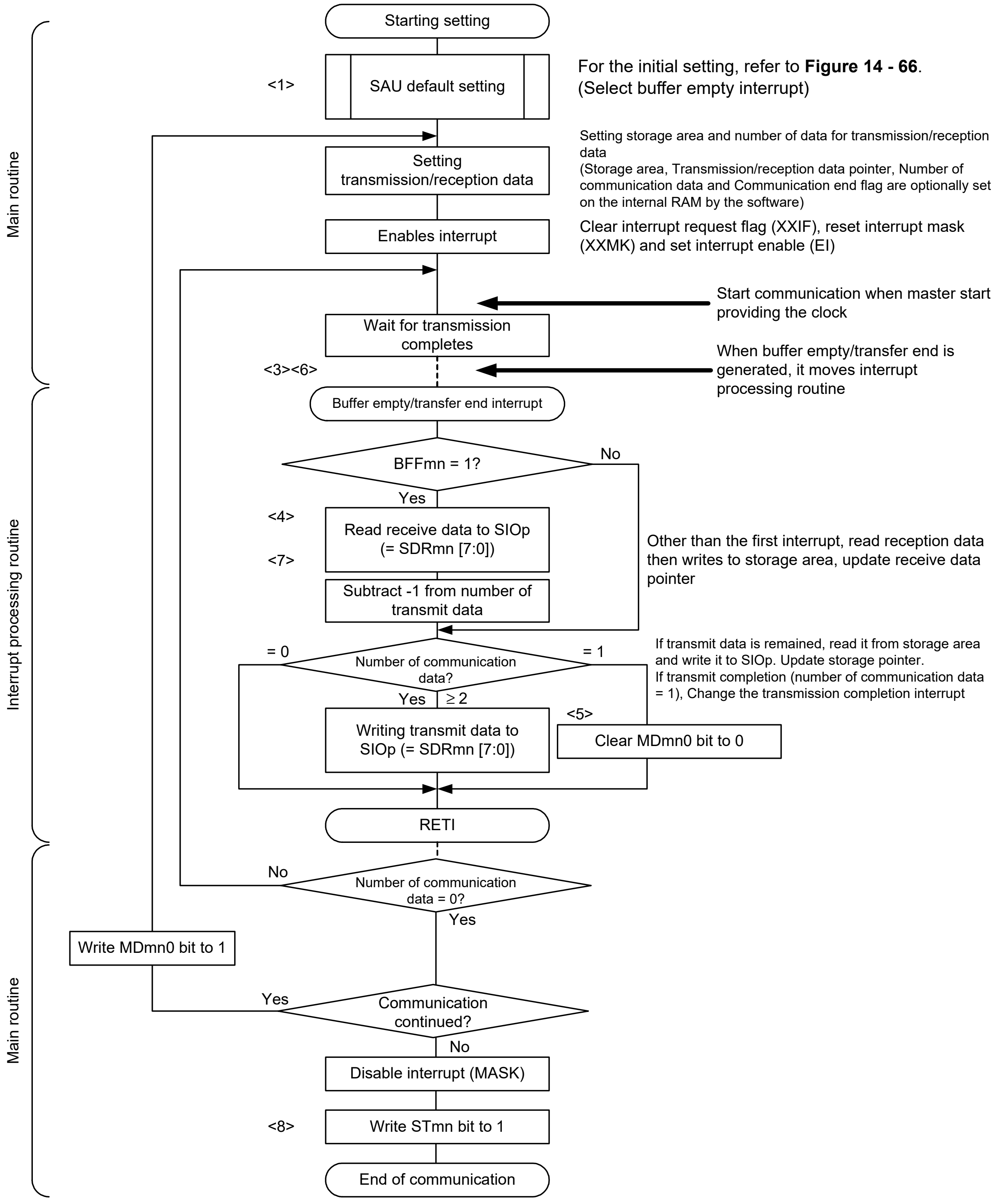
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)
mn = 00

Figure 14 - 72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.7 SNOOZE mode function

In the SNOOZE mode, the simplified SPI (CSI) proceeds with reception upon detection of input on the SCKp pin while in the STOP mode. Transfer through the simplified SPI (CSI) will normally stop in the STOP mode. However, using the SNOOZE mode enables reception by the simplified SPI (CSI) without CPU operations upon detection of input on the SCKp pin. Only the following channel can be set to the SNOOZE mode.

- CSI00

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 14 - 74** and **Figure 14 - 76** Flowchart of SNOOZE Mode Operation).

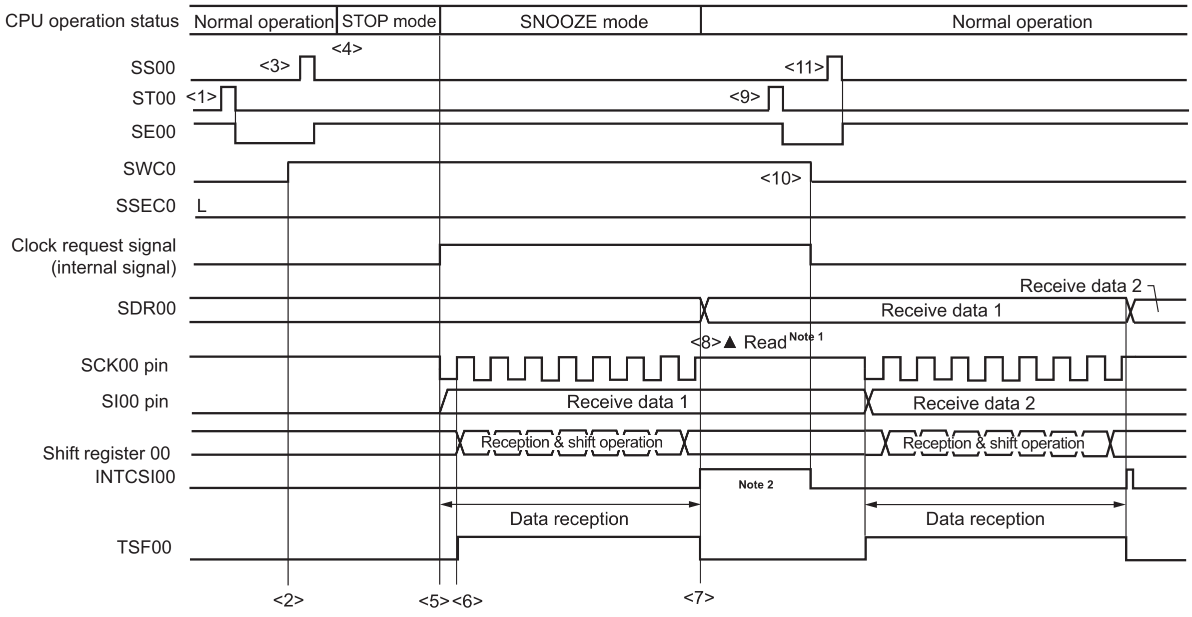
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode.
  A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for fCLK.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 14 - 73 Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

Note 2. The transfer end interrupt (INTCSIp) is cleared either when SWCm is cleared to 0 or when the next edge of the SCKp pin input is detected.

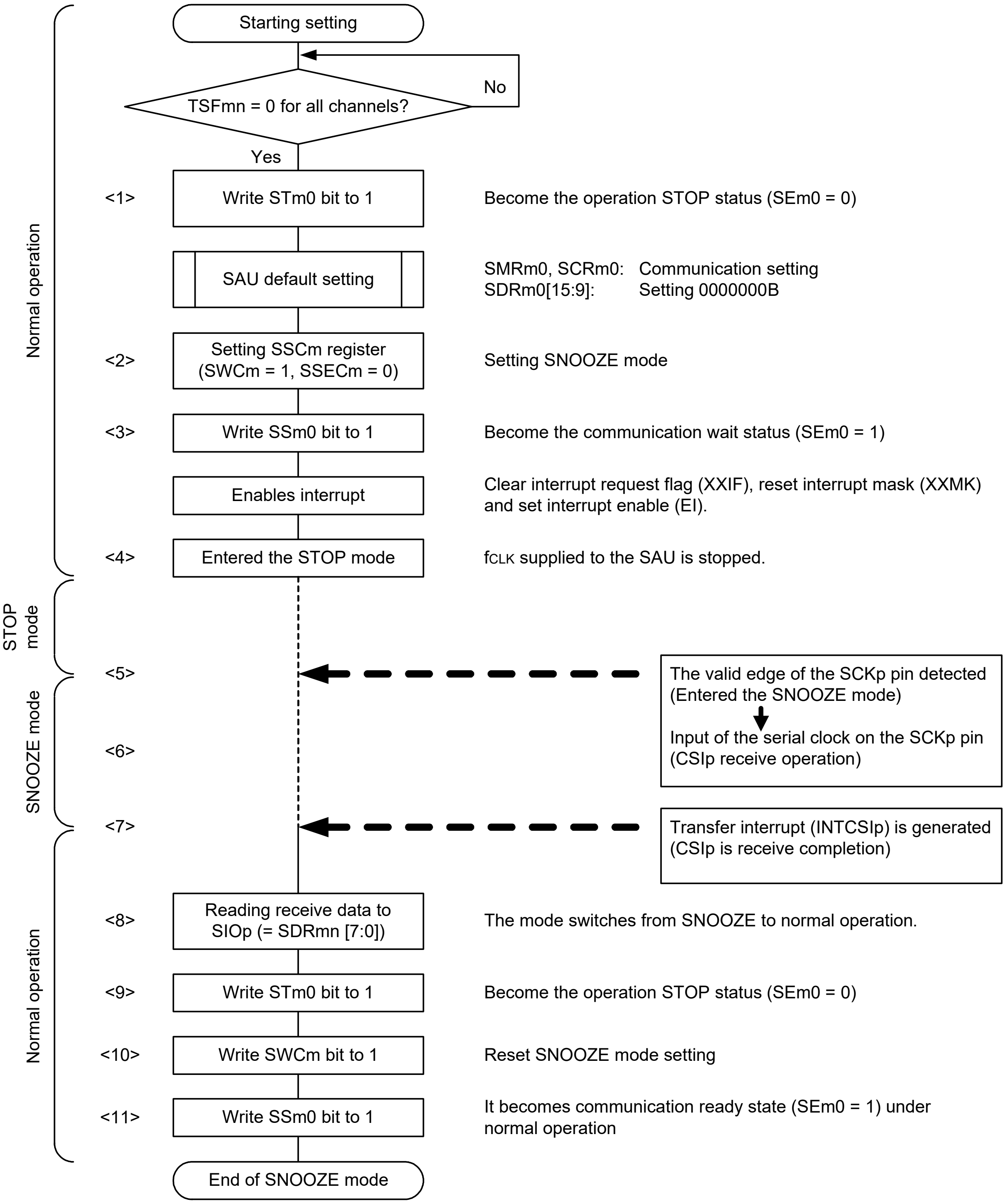
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).**
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 14 - 74 Flowchart of SNOOZE Mode Operation (Once

Startup).

Remark 2. m = 0; p = 00

Figure 14 - 74 Flowchart of SNOOZE Mode Operation (Once Startup)

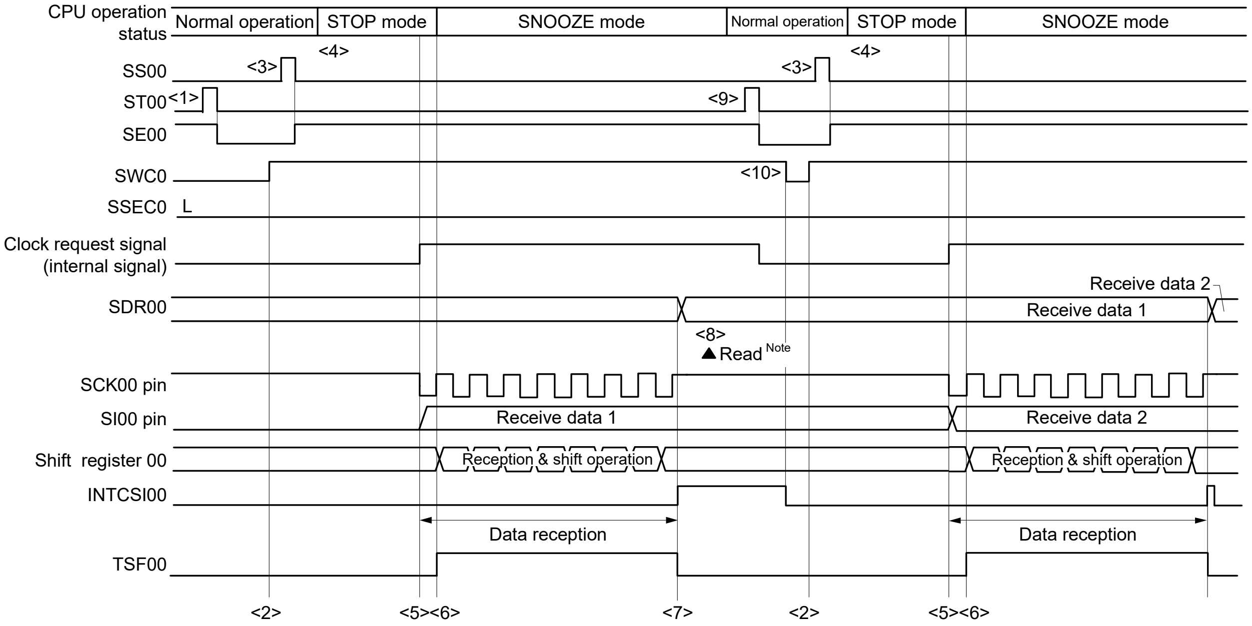


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 14 - 73 Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 14 - 75 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

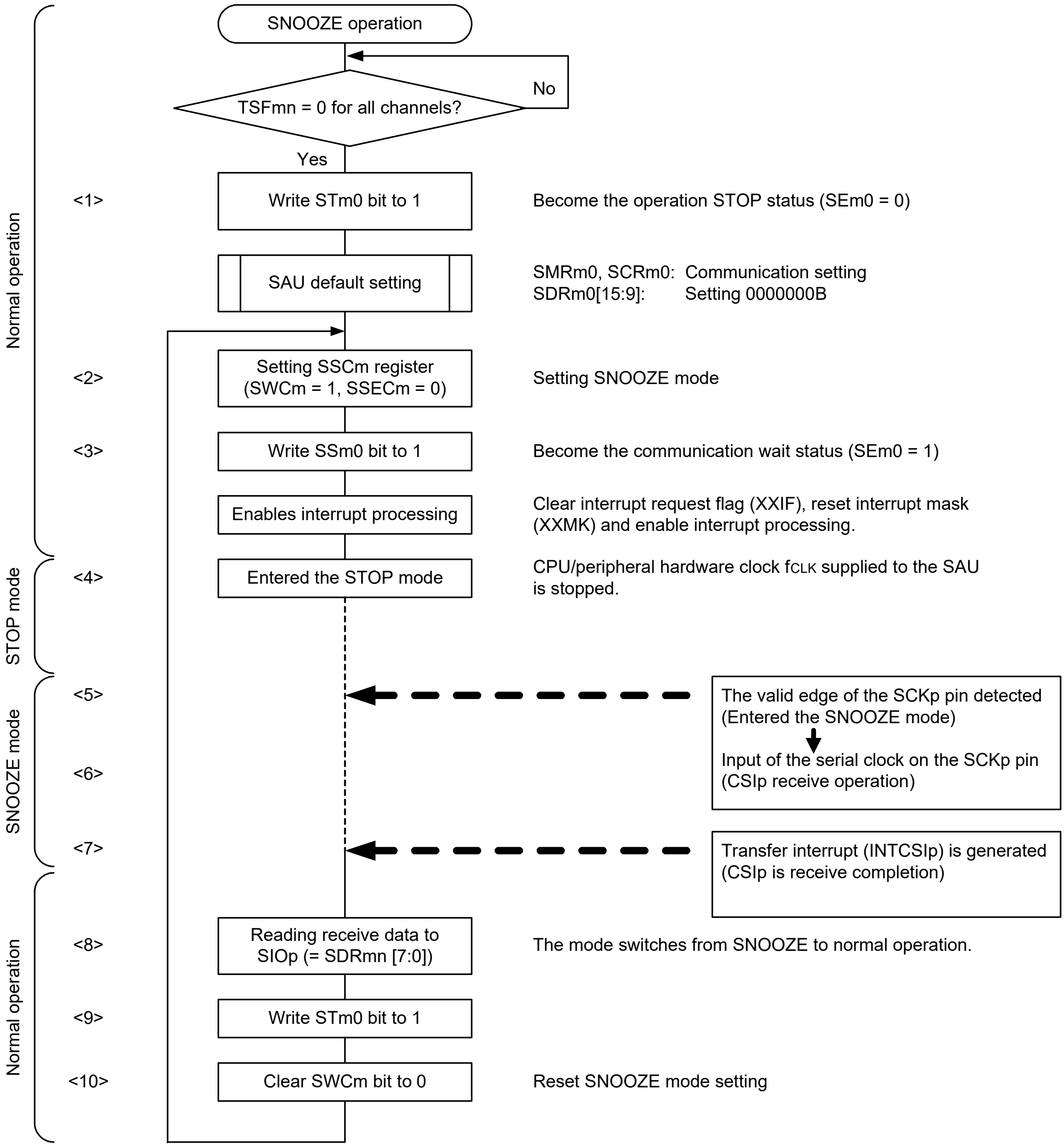
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 14 - 76 Flowchart of SNOOZE Mode Operation (Continuous Startup).

Remark 2. m = 0; p = 00

Figure 14 - 76 Flowchart of SNOOZE Mode Operation (Continuous Startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 14 - 75 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

14.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14 - 2 Selection of Operation Clock For Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.88 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.75 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.88 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

14.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00) communication

The procedure for processing errors that occurred during simplified SPI (CSI00) communication is described in Figure 14 - 77.

Figure 14 - 77 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

14.6 Clock Synchronous Serial Communication with Slave Select Input Function

Channel 0 of SAU0 correspond to the clock synchronous serial communication with slave select input function.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate Note

  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

[Expansion function]
- Slave select function

Note Use the clocks within a range satisfying the SCK cycle time (tKCY) characteristics. For details, see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**.

- 30, 32, 44-pin products

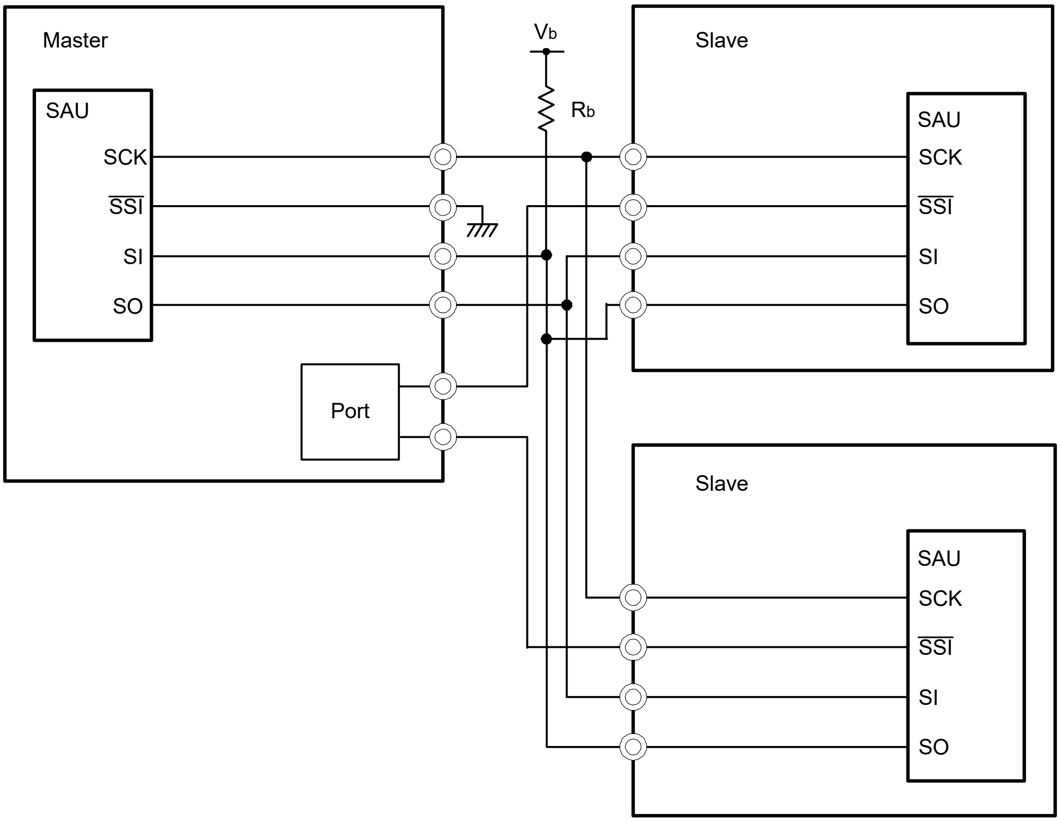
Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

Slave select input function performs the following three types of communication operations.
- Slave transmission (See **14.6.1**.)
- Slave reception (See **14.6.2**.)
- Slave transmission/reception (See **14.6.3**.)

Multiple slaves can be connected to a master and communication can be performed by using the slave select input function. The master outputs a slave select signal to the slave (one) that is the other party of communication, and each slave judges whether it has been selected as the other party of communication and controls the SO pin output. When a slave is selected, transmit data can be communicated from the SO pin to the master. When a slave is not selected, the SO pin is set to high-level output. Therefore, in an environment where multiple slaves are connected, it is necessary set the SO pin to N-ch open-drain and pull up the node. Furthermore, when a slave is not selected, no transmission/reception operation is performed even if a serial clock is input from the master.

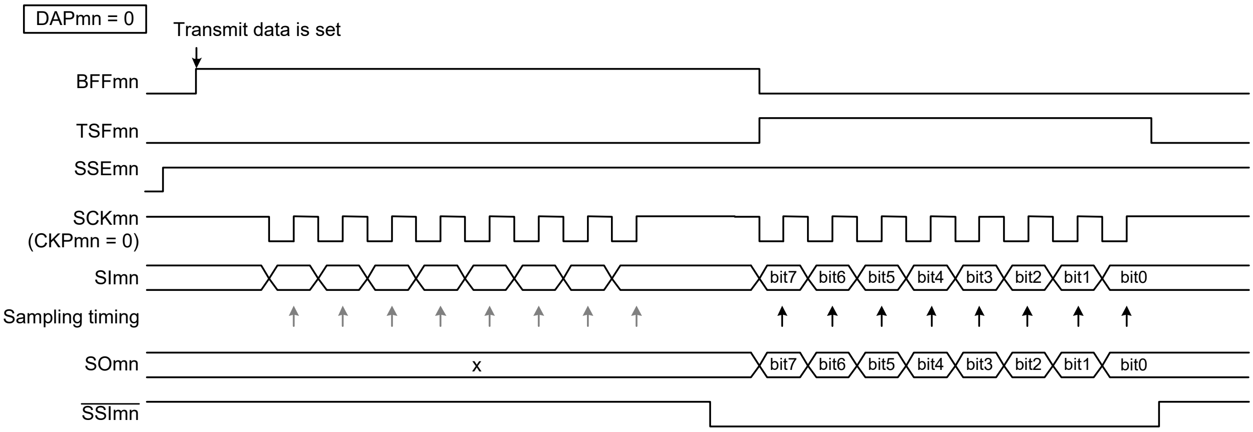
Caution Output the slave select signal by port manipulation.

Figure 14 - 78 Example of Slave Select Input Function Configuration

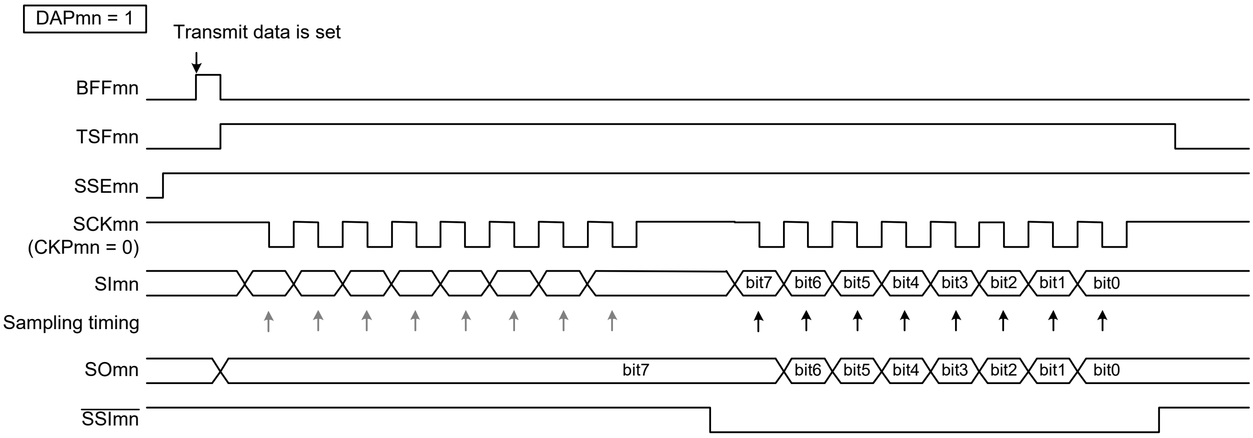


Caution Make sure VDD ≥ Vb.
Select the N-ch open-drain output (VDD tolerance) mode for the SO00 pin.

Figure 14 - 79 Slave Select Input Function Timing Diagram



While $\overline{\text{SSImn}}$ is at high level, transmission is not performed even if the falling edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the rising edge.
When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.



If DAPmn = 1, when transmit data is set while $\overline{\text{SSImn}}$ is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

14.6.1 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Slave select Input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select Input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

Figure 14 - 80 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting							

Bits 7 to 0: SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) slave transmission mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14 - 81 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

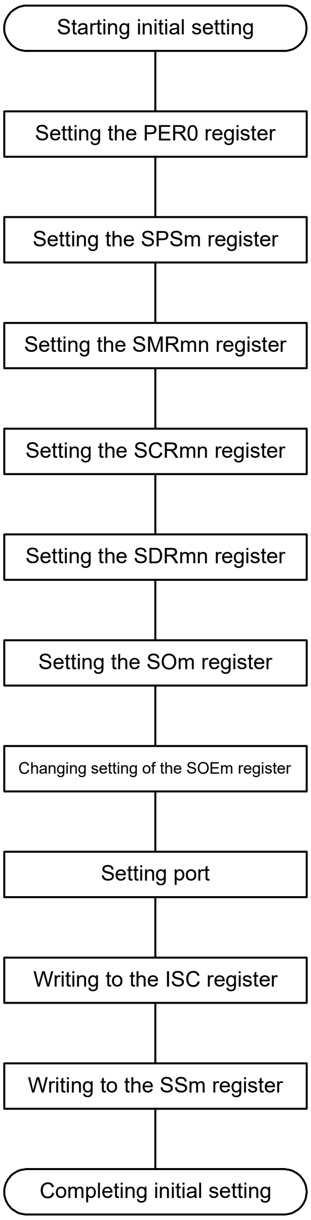
0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 82 Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register. When connecting multiple slaves, set N-ch open-drain before setting data output.

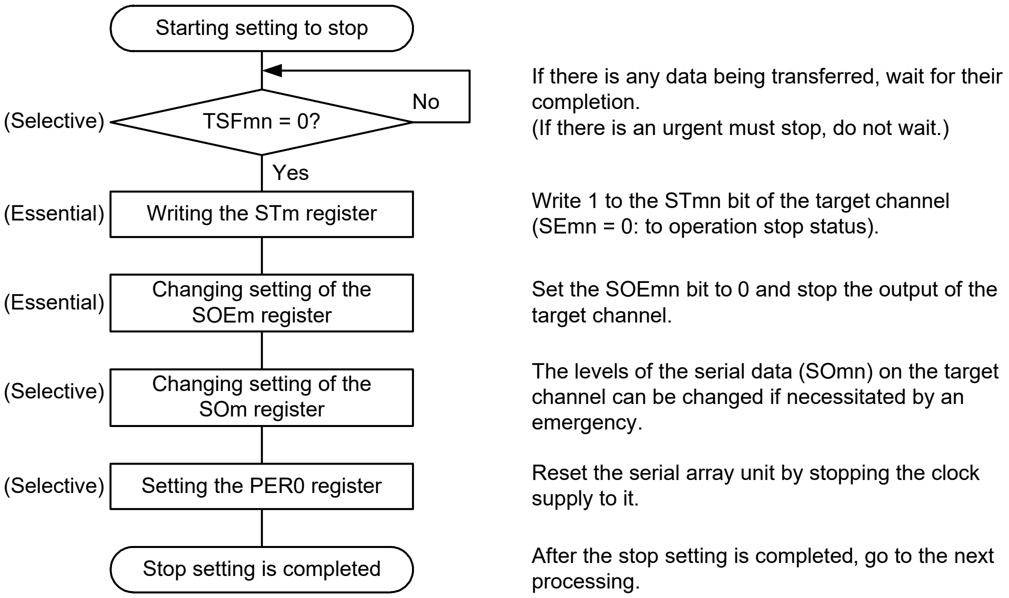
Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 83 Procedure for Stopping Slave Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel (SEmn = 0: to operation stop status).

Set the SOEmn bit to 0 and stop the output of the target channel.

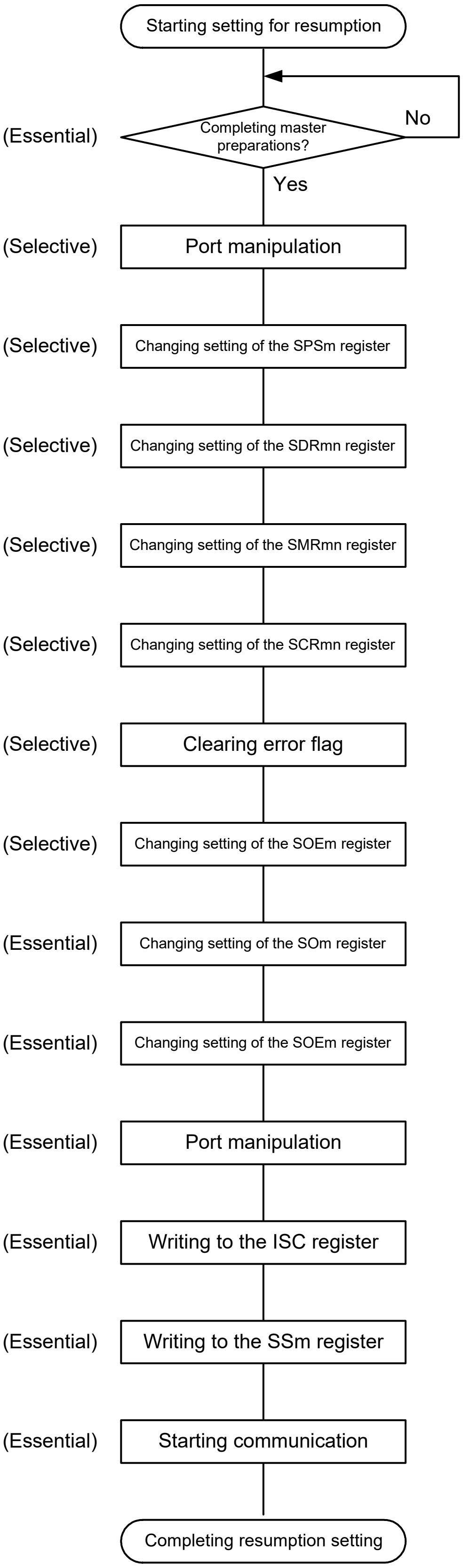
The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 84 Procedure for Resuming Slave Transmission



Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register. When connecting multiple slaves, set N-ch open-drain before setting data output.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

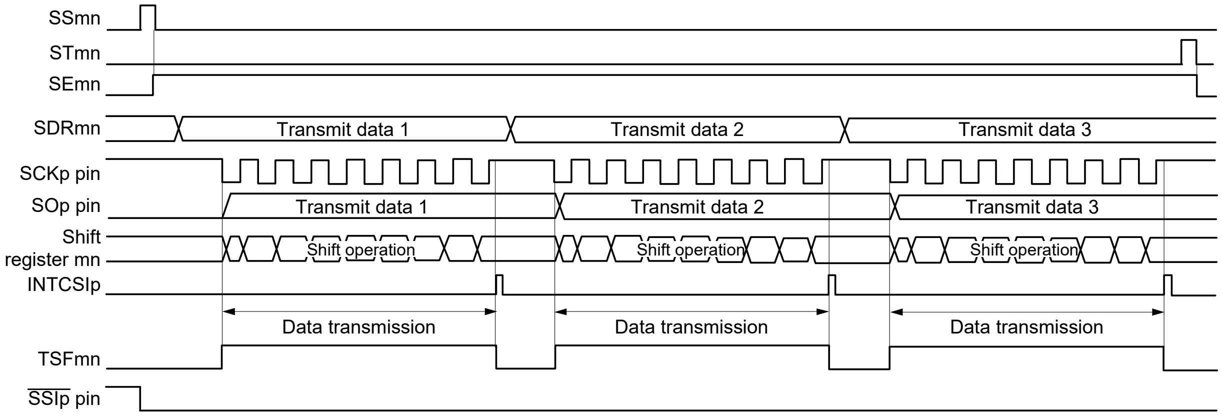
Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

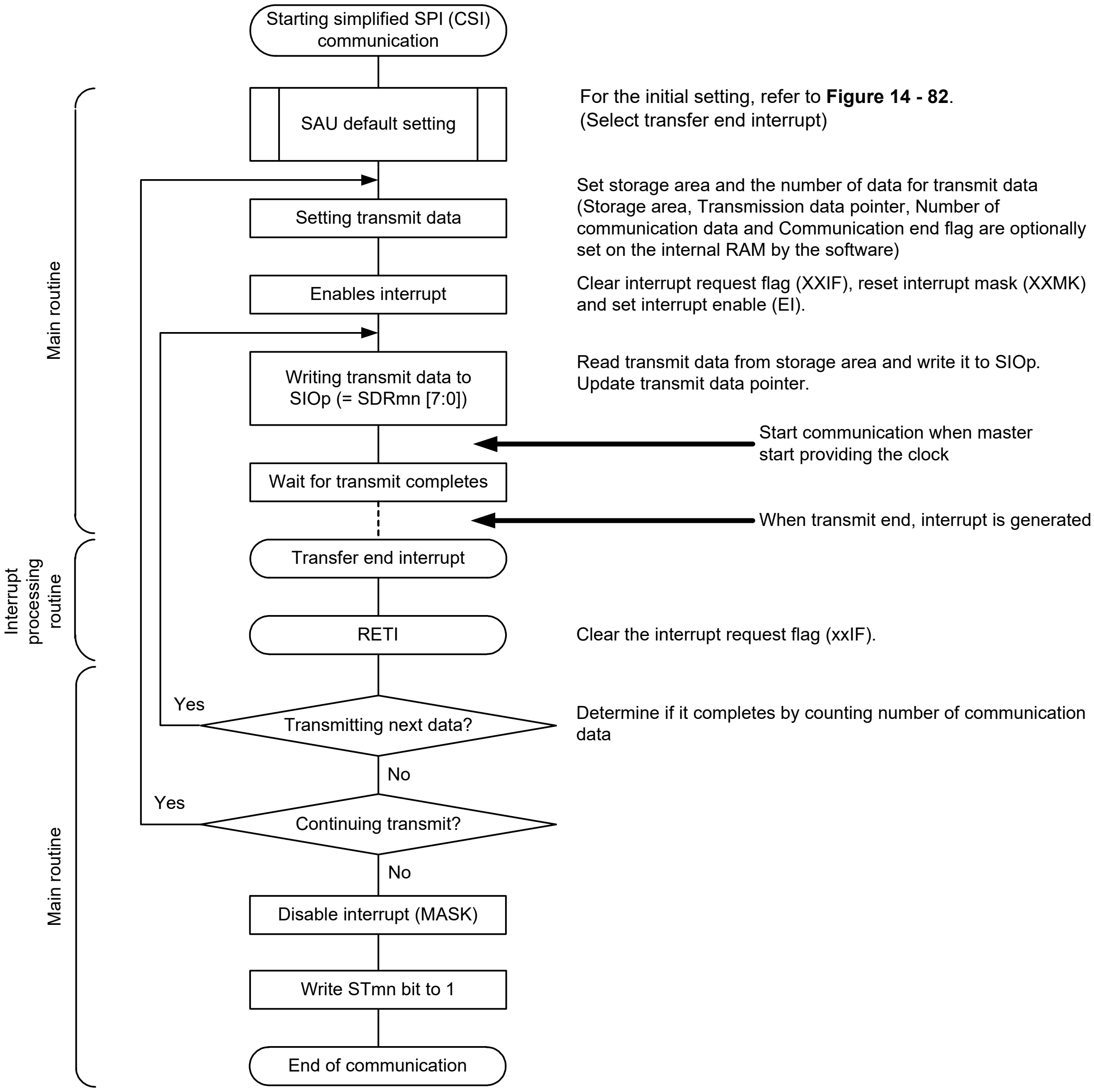
(3) Processing flow (in single-transmission mode)

Figure 14 - 85 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

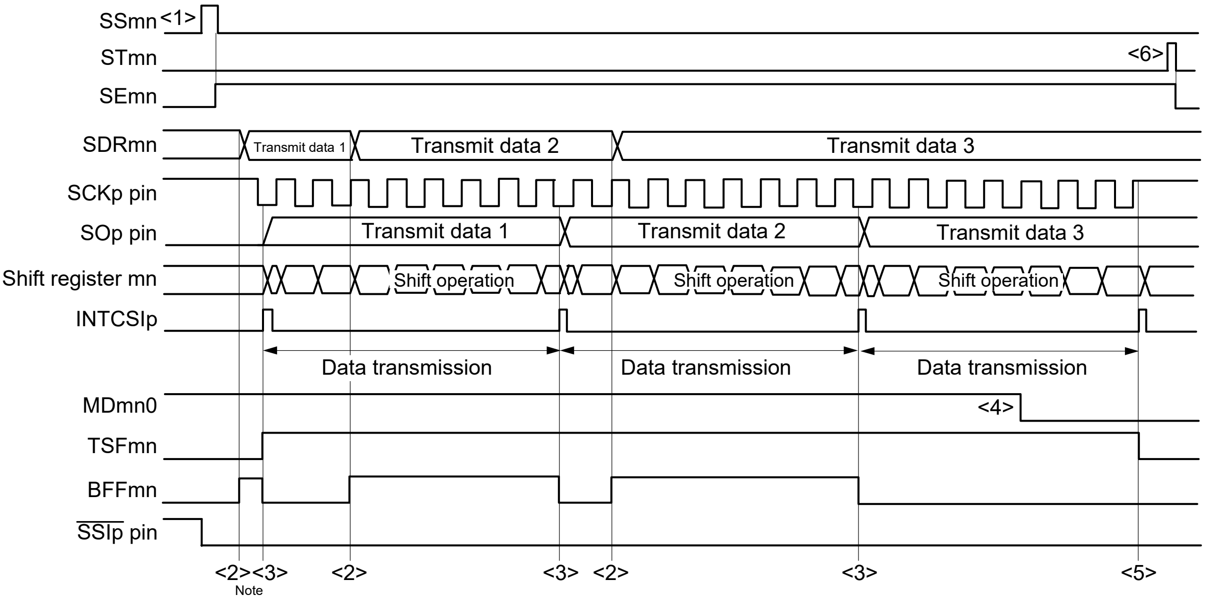
Figure 14 - 86 Flowchart of Slave Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing flow (in continuous transmission mode)

Figure 14 - 87 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

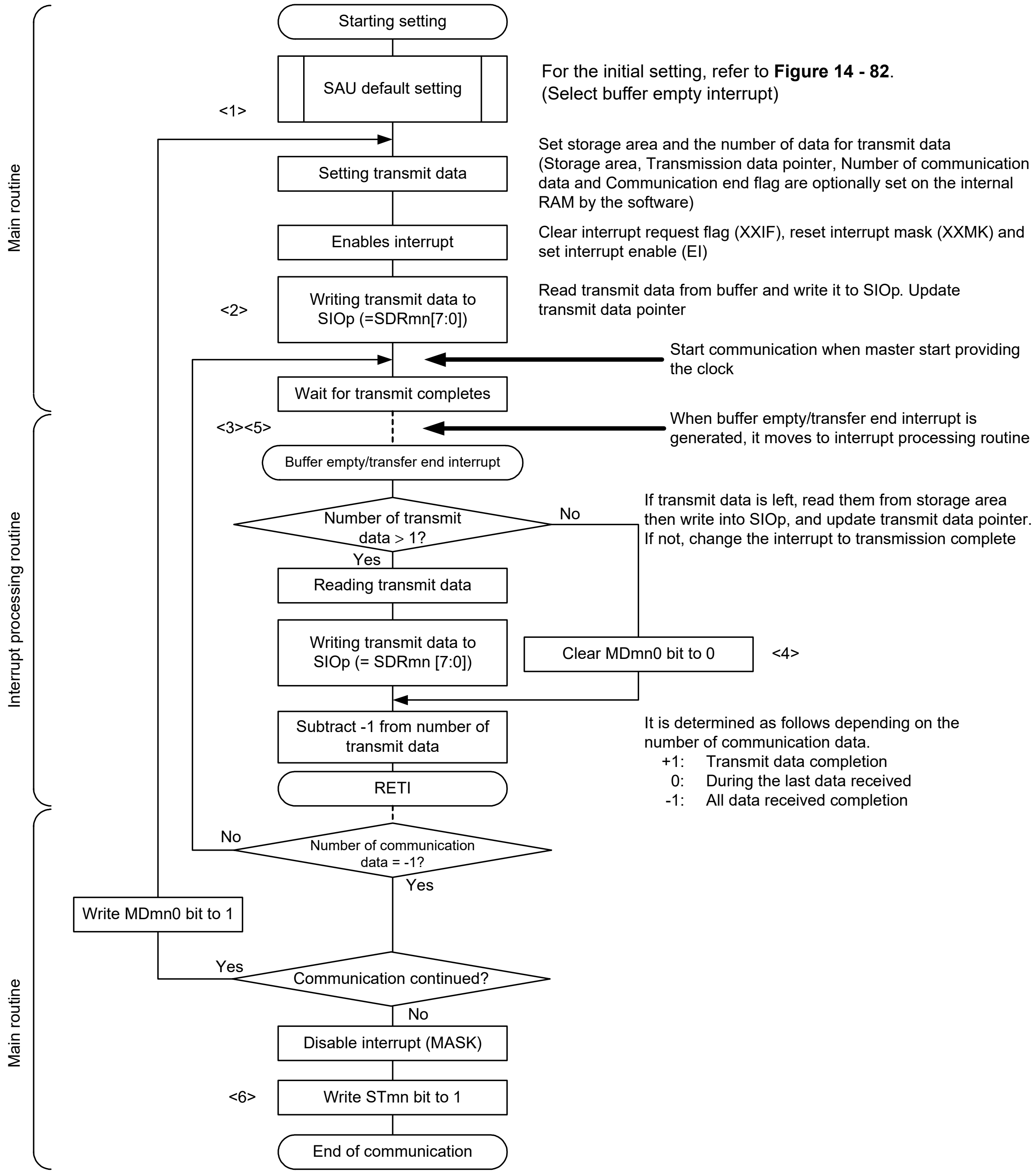


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 88 Flowchart of Slave Transmission (in Continuous Transmission Mode)



For the initial setting, refer to **Figure 14 - 82**.
(Select buffer empty interrupt)

Set storage area and the number of data for transmit data (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Read transmit data from buffer and write it to SIOp. Update transmit data pointer

Start communication when master start providing the clock

When buffer empty/transfer end interrupt is generated, it moves to interrupt processing routine

If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer. If not, change the interrupt to transmission complete

It is determined as follows depending on the number of communication data.
+1: Transmit data completion
0: During the last data received
-1: All data received completion

Remark 1. <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 87 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

14.6.2 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

Figure 14 - 89 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							

SIOp

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. □: Setting is fixed in the simplified SPI (CSI) slave reception mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14 - 90 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

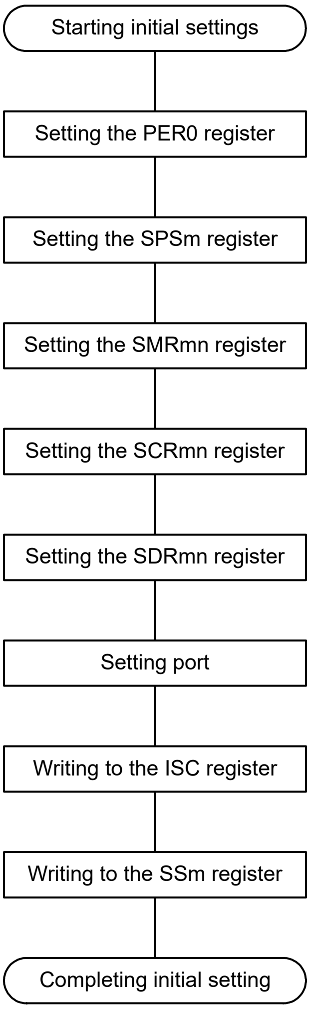
	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

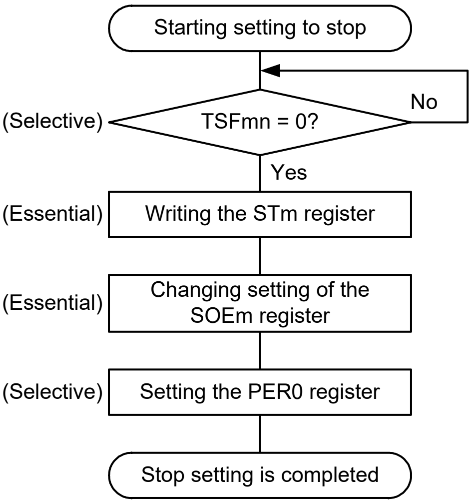
Set bits 15 to 9 to 0000000B for baud rate setting.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 14 - 91 Initial Setting Procedure for Slave Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

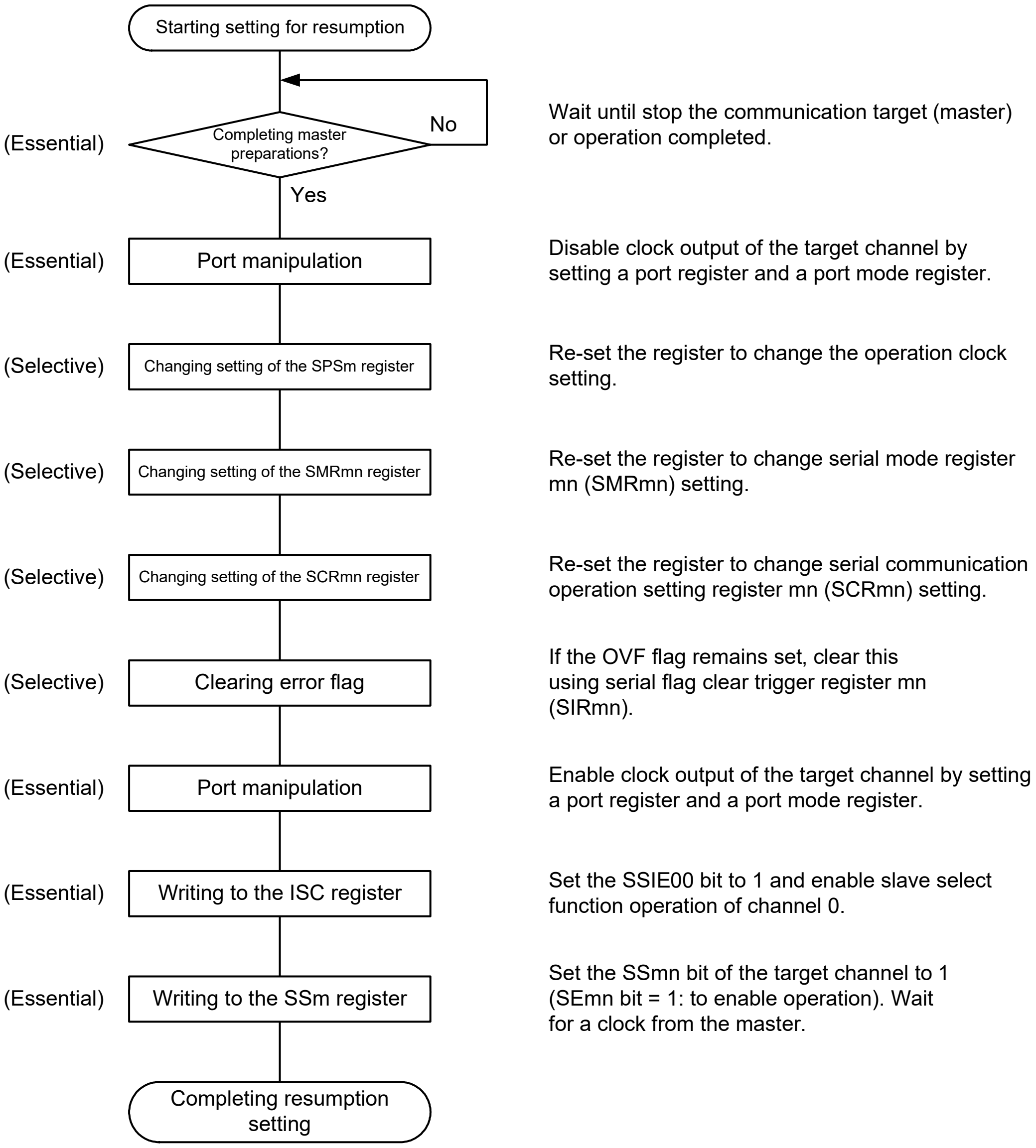
Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 92 Procedure for Stopping Slave Reception

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

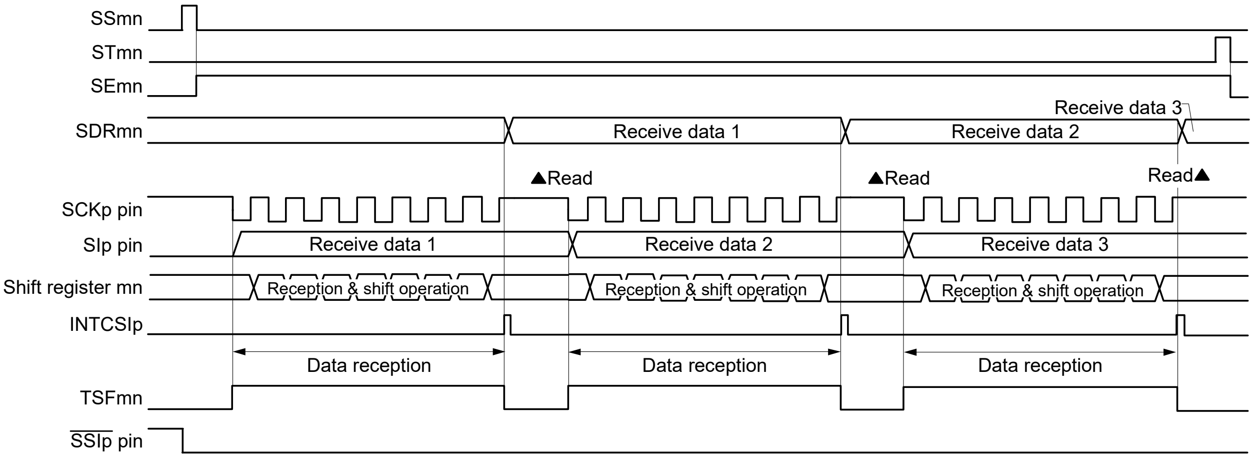
Figure 14 - 93 Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

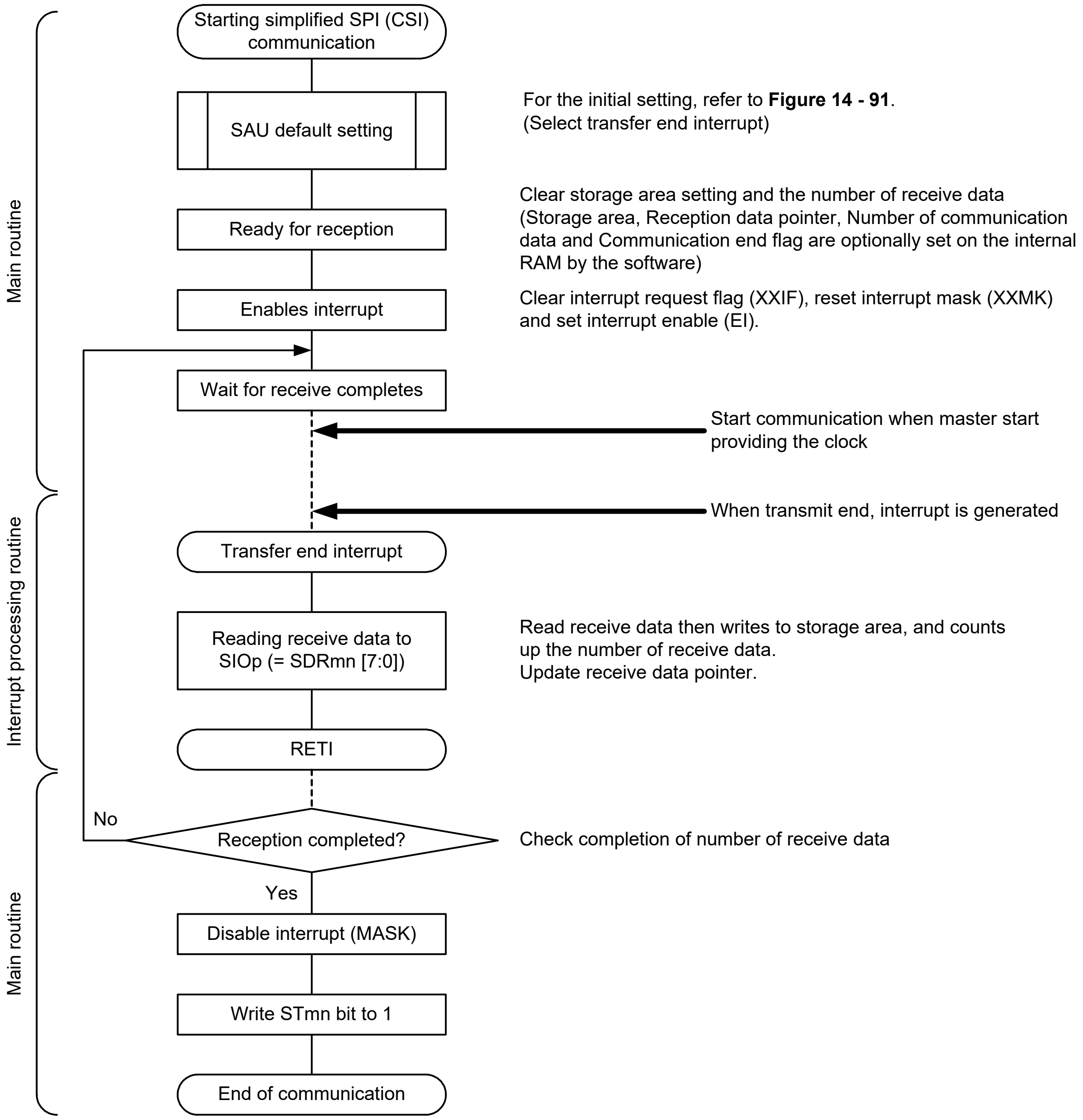
(3) Processing flow (in single-reception mode)

Figure 14 - 94 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 95 Flowchart of Slave Reception (in Single-Reception Mode)



14.6.3 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

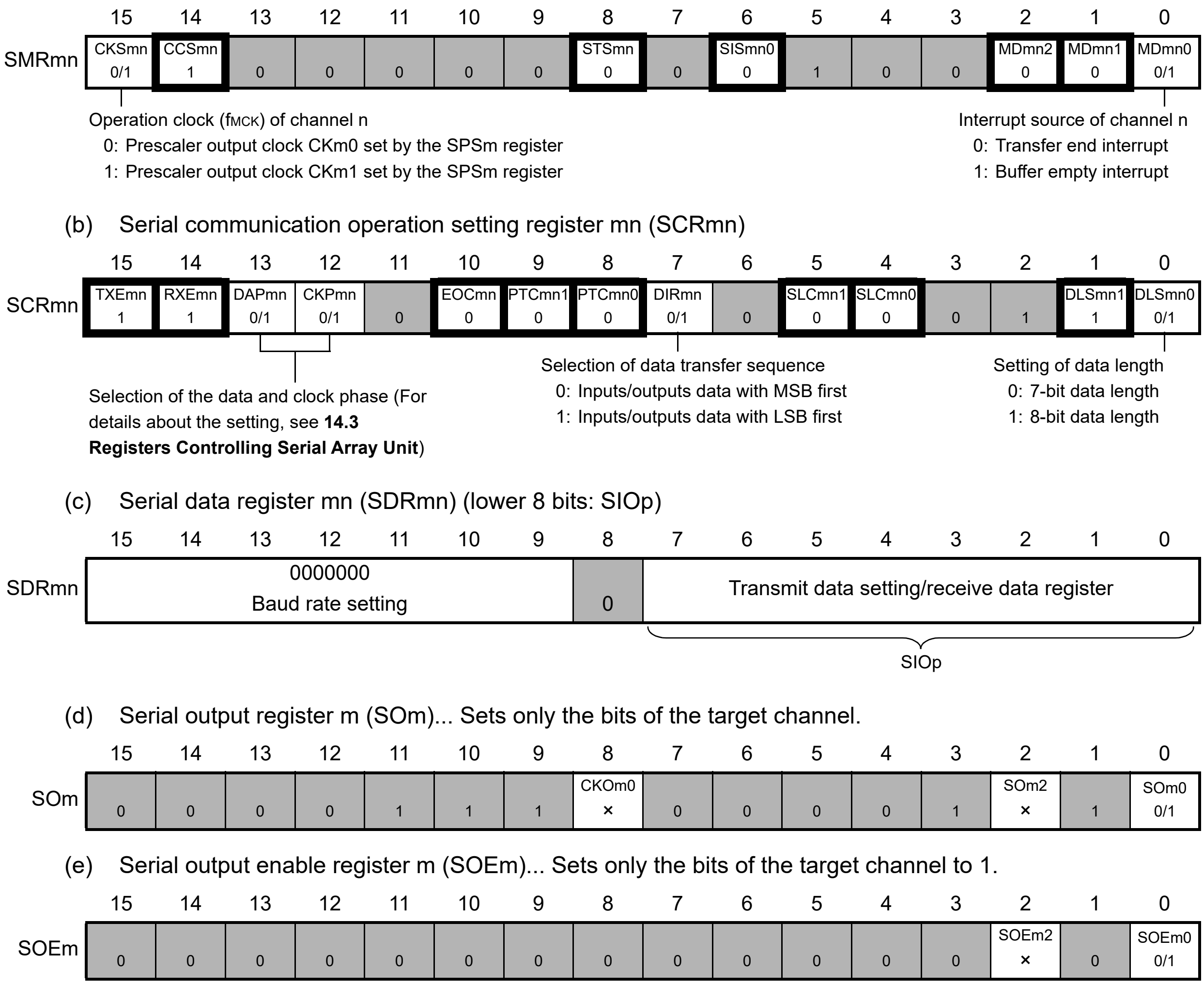


Figure 14 - 96 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **14.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 Baud rate setting	0	Transmit data setting/receive data register (SIOp)

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14 - 97 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

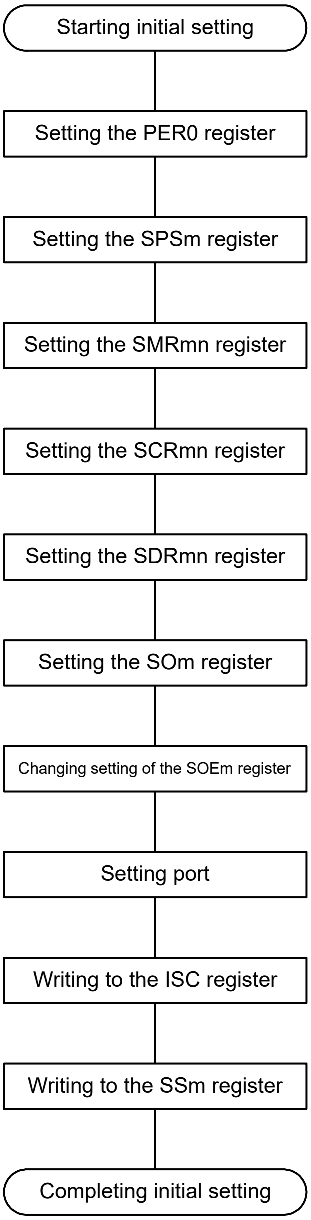
Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 98 Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register. When connecting multiple slaves, set N-ch open-drain before setting data output.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

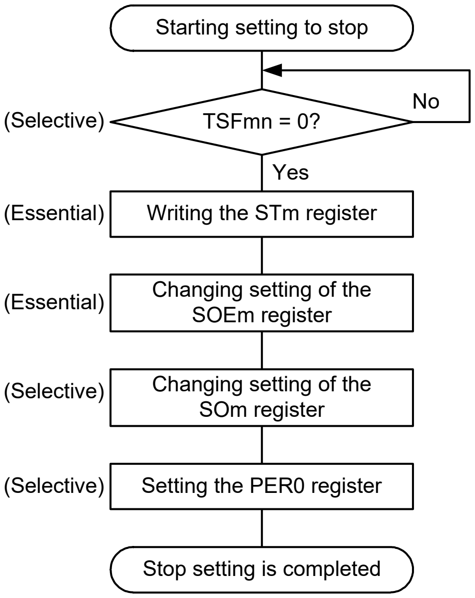
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 99 Procedure for Stopping Slave Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

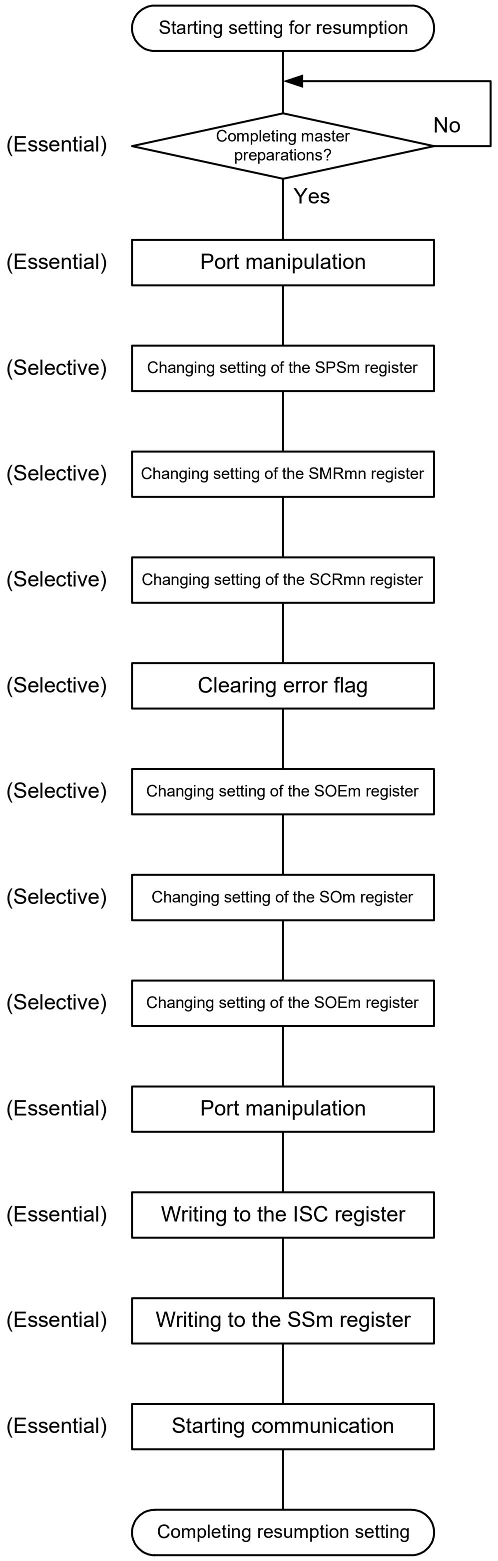
The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 100 Procedure for Resuming Slave Transmission/Reception



Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register. When connecting multiple slaves, set N-ch open-drain before setting data output.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

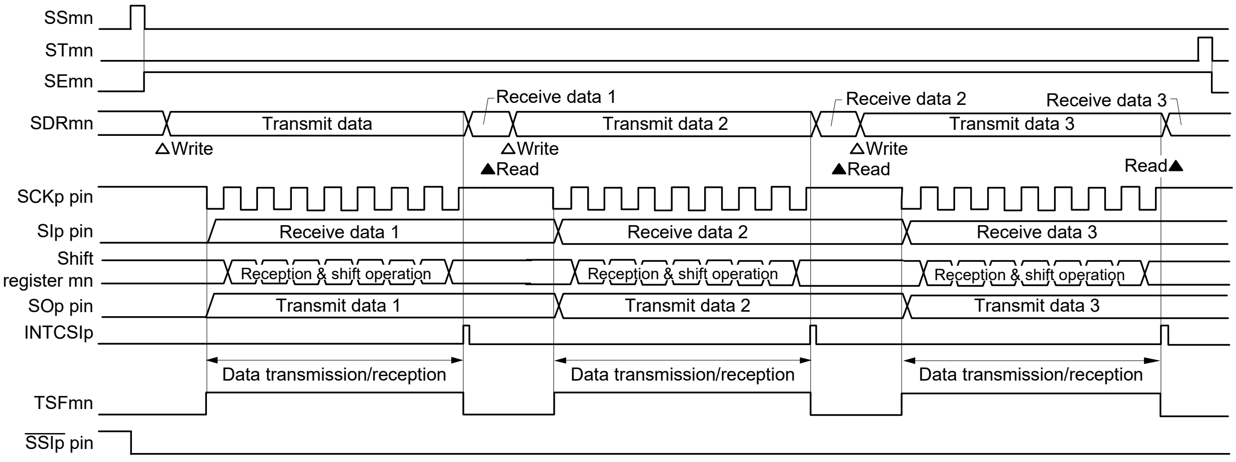
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

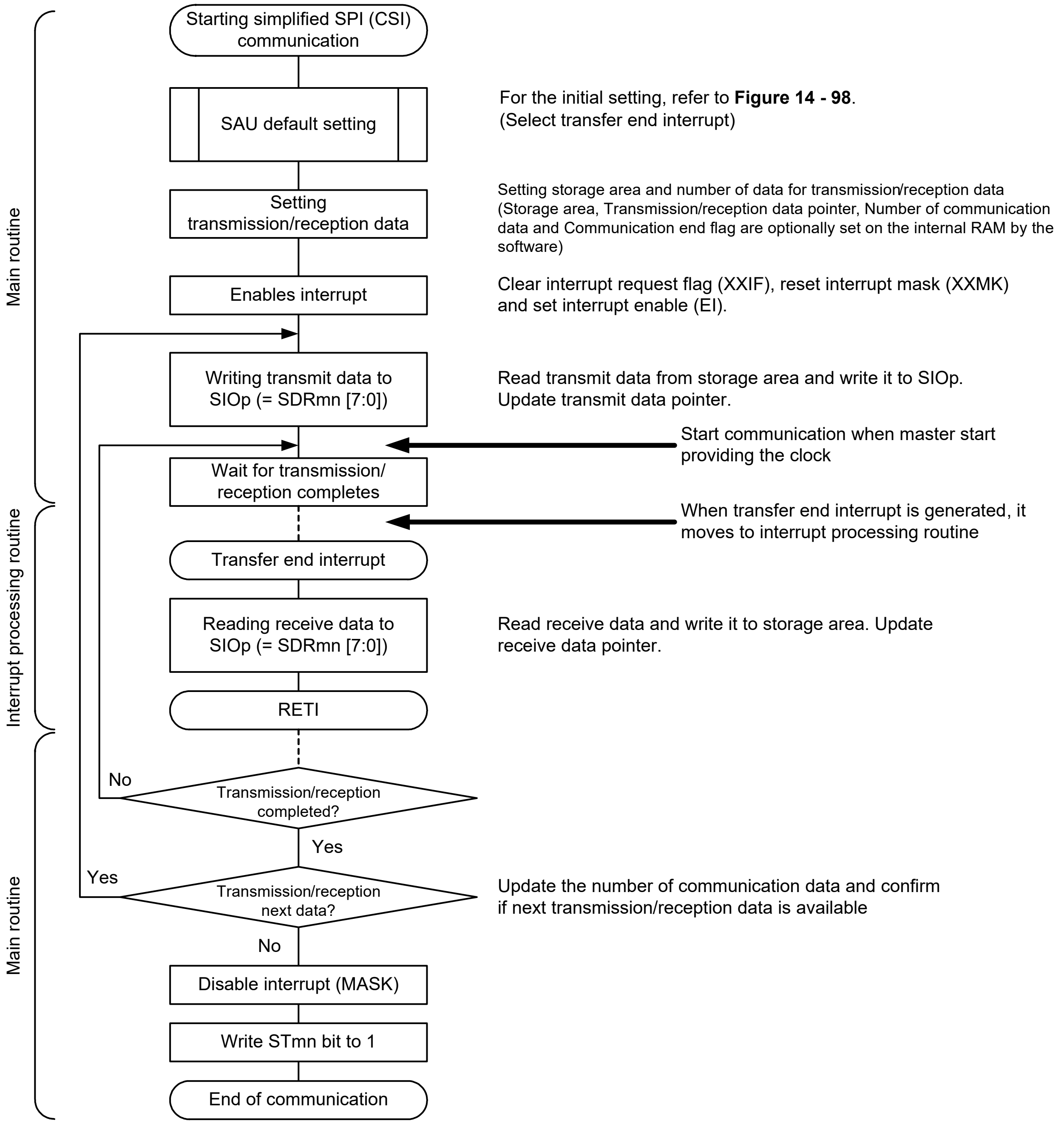
(3) Processing flow (in single-transmission/reception mode)

Figure 14 - 101 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 102 Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



For the initial setting, refer to **Figure 14 - 98**.
(Select transfer end interrupt)

Setting storage area and number of data for transmission/reception data (Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Start communication when master start providing the clock

When transfer end interrupt is generated, it moves to interrupt processing routine

Read receive data and write it to storage area. Update receive data pointer.

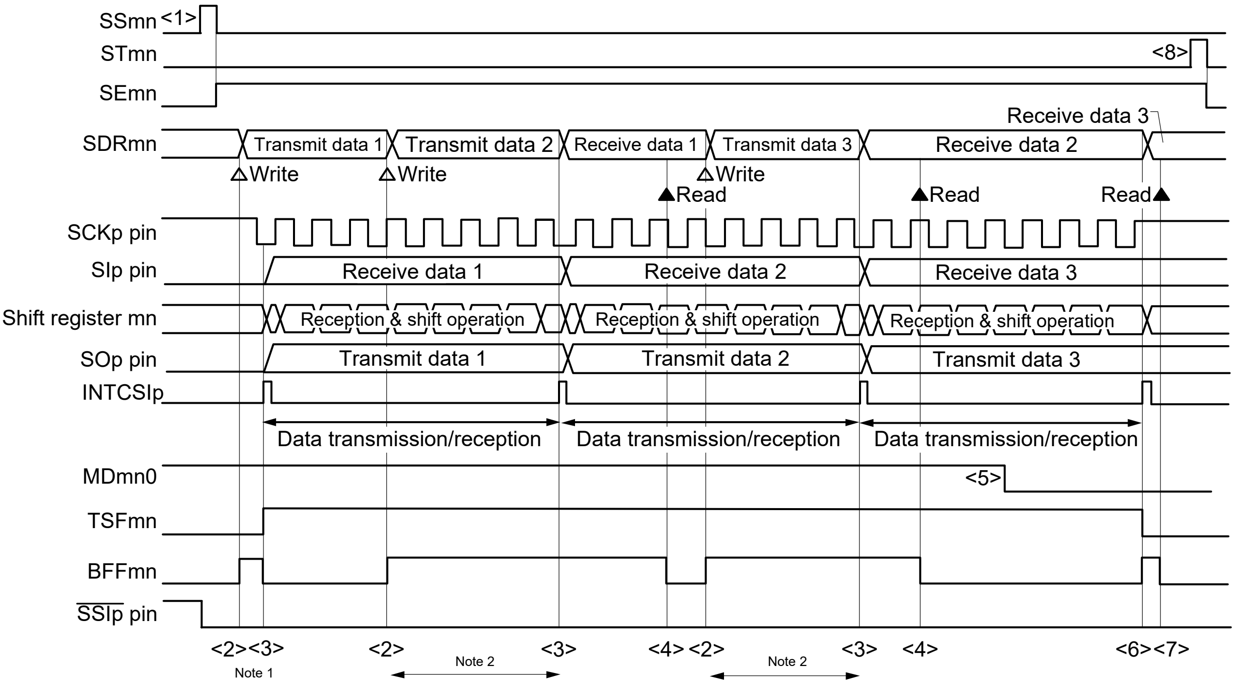
Update the number of communication data and confirm if next transmission/reception data is available

Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing flow (in continuous transmission/reception mode)

Figure 14 - 103 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

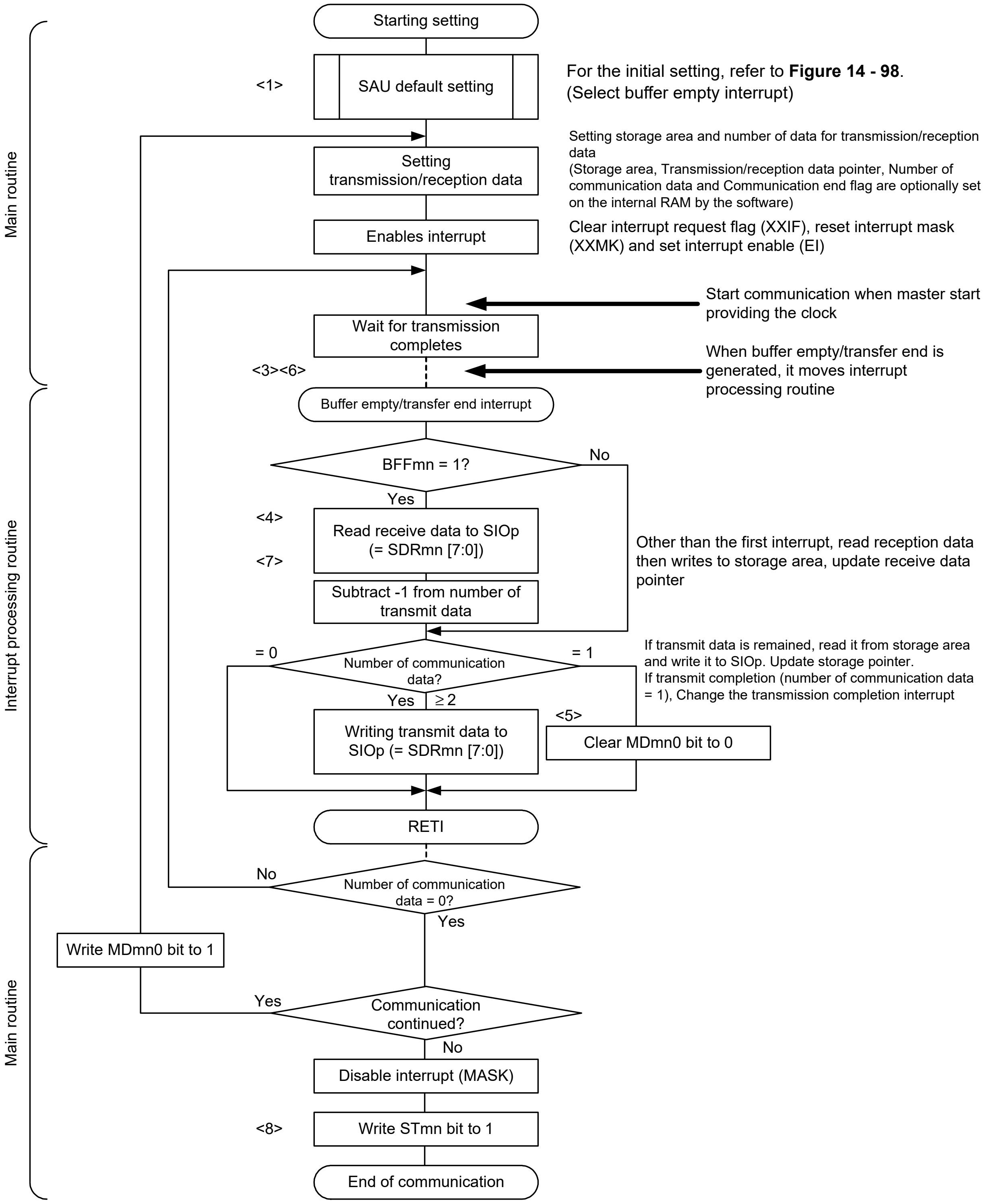
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 104 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 14 - 104 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 103 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

14.6.4 Calculating transfer clock frequency

The transfer clock frequency for slave select input function (CSI00) communication can be calculated by the following expressions.

(1) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Table 14 - 3 Selection of Operation Clock For Slave Select Input Function

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.88 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don’t care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

14.6.5 Procedure for processing errors that occurred during slave select input function communication

The procedure for processing errors that occurred during slave select input function communication is described in Figure 14 - 105.

Figure 14 - 105 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

14.7 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]
- Data length of 7, 8, or 9 bits [Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

In addition, UART0 supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following UART can be specified when FRQSEL4 in the option byte (000C2H) = 0 in the SNOOZE mode.
- UART0

Note Only following UART can be specified for the 9-bit data length.
- UART0

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.

- 30, 32, 44-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00 and IIC00.
At this time, however, channel 0, 1, or other channels of the same unit can be used for a function other than UART1, such as CSI00, UART0, and IIC00.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following two types of communication operations.

• UART transmission	(See **14.7.1**.)
• UART reception	(See **14.7.2**.)

14.7.1 UART transmission

UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).
Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	TxD0	TxD1
Interrupt	INTST0	INTST1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7, 8, or 9 bits Note 1	
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note 2	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity	
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits	
Data direction	MSB or LSB first	

Note 1. Only following UART can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 14 - 106 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	0	0	0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0/1	SLCmn0 0/1	0	1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0 Note	Transmit data setting							
									TXDq							

(d) Serial output level register m (SOLm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm2 0/1	0	SOLm0 0/1

0: Non-reverse (normal) transmission
1: Reverse transmission

Note When performing 9-bit communication, bits 0 to 8 of the SDRm0 register are used to specify the transmission data.
▪ UART0

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1),
mn = 00, 02

Remark 2. ▣: Setting is fixed in the UART transmission mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14 - 107 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1 Note	1	SOm0 0/1 Note

0: Serial data output value is "0"
1: Serial data output value is "1"

(f) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1)
mn = 00, 02

Remark 2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 108 Initial Setting Procedure for UART Transmission

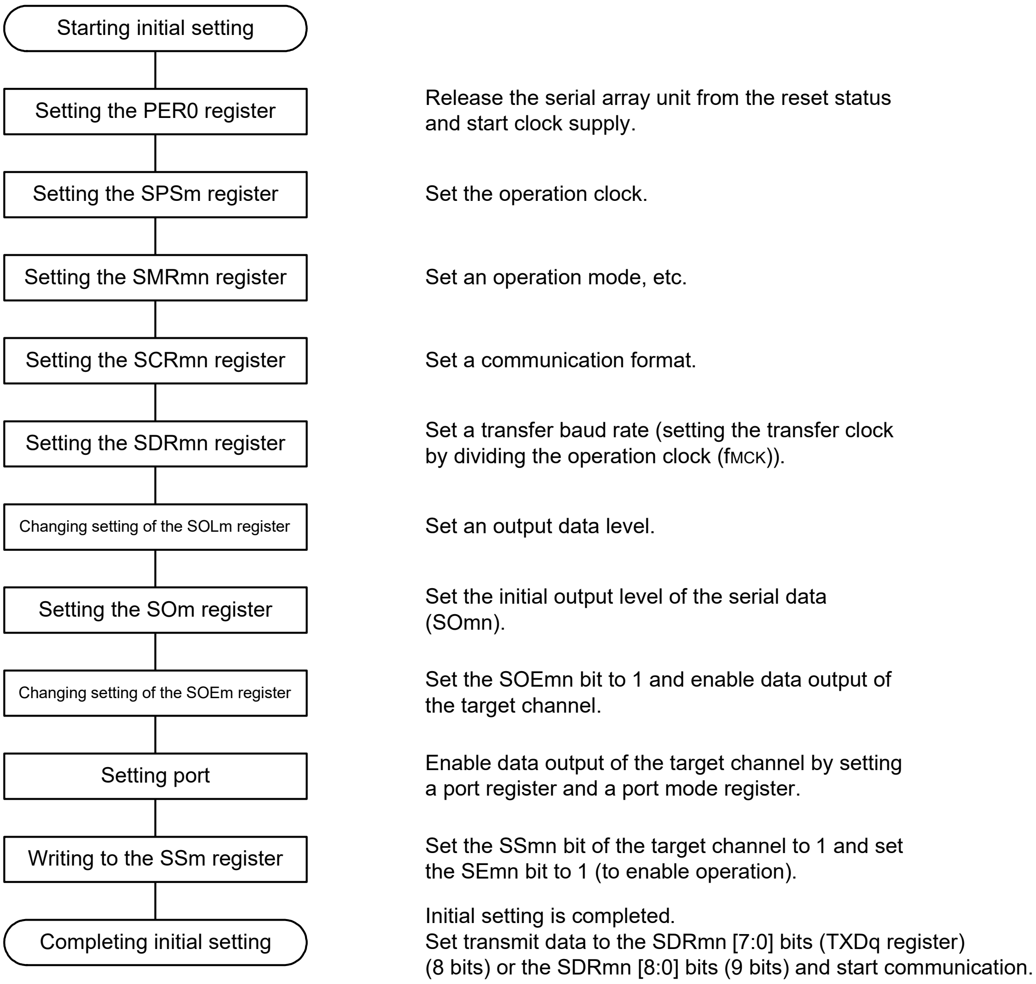


Figure 14 - 109 Procedure for Stopping UART Transmission

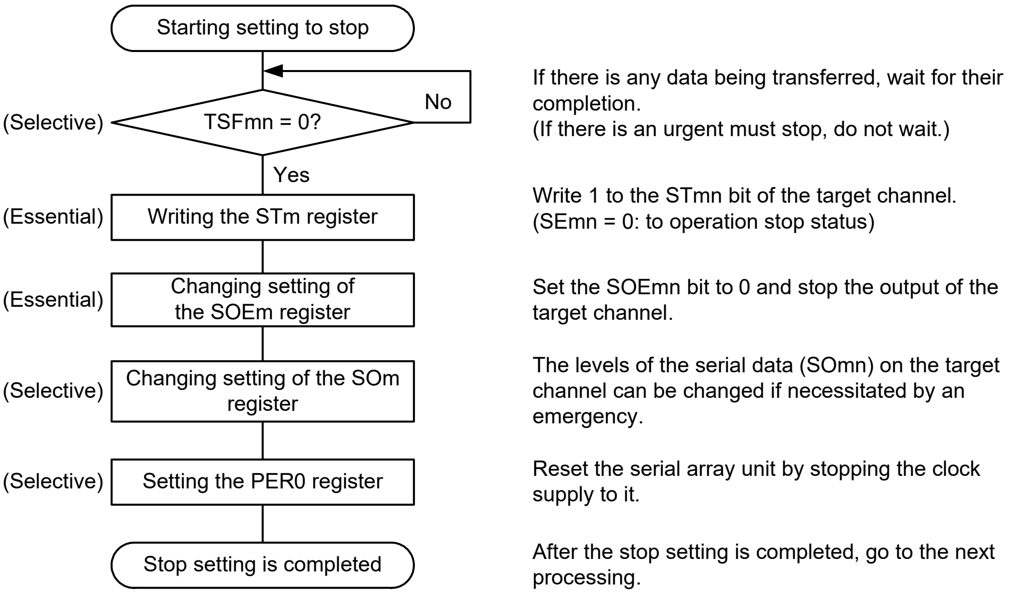
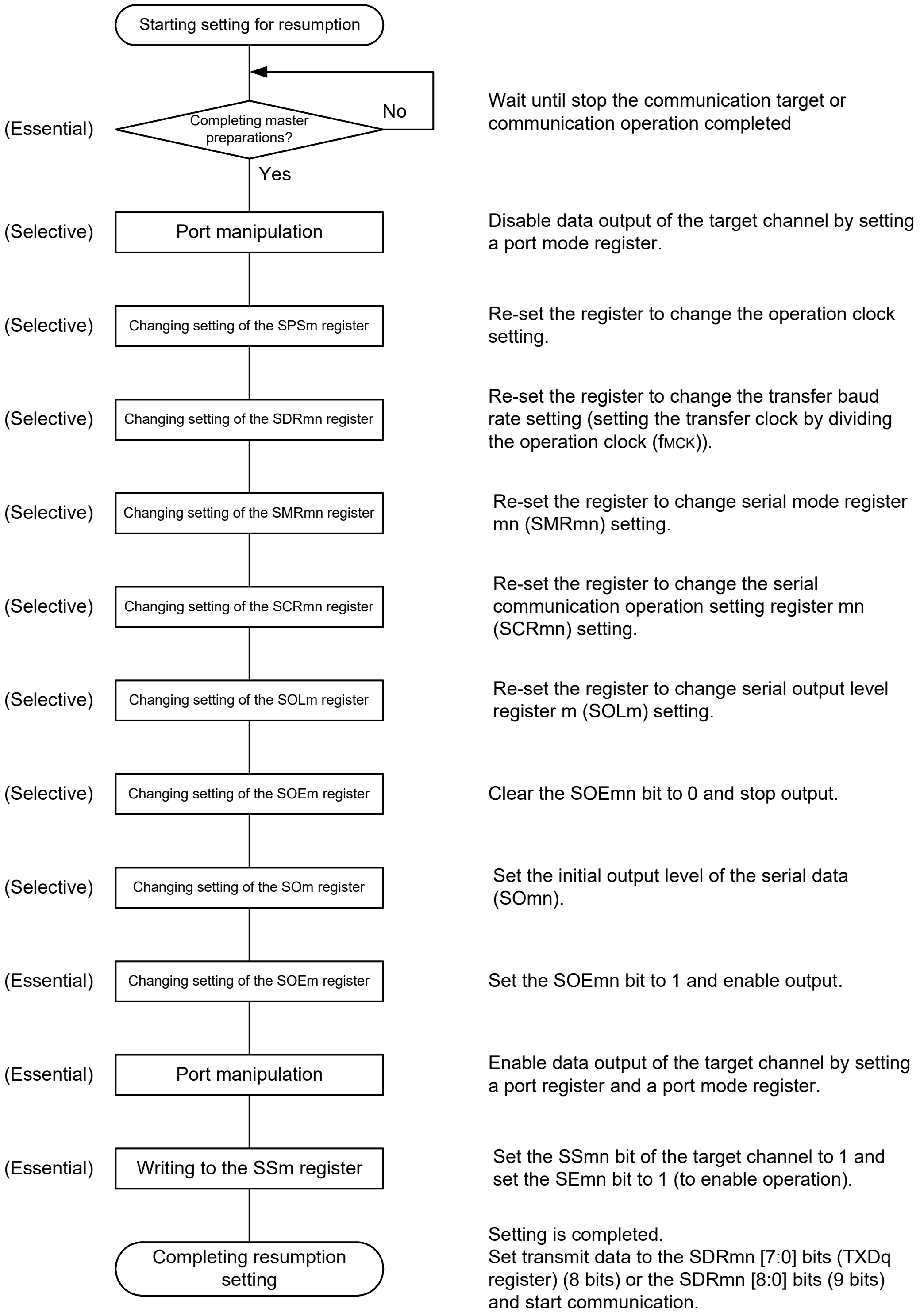


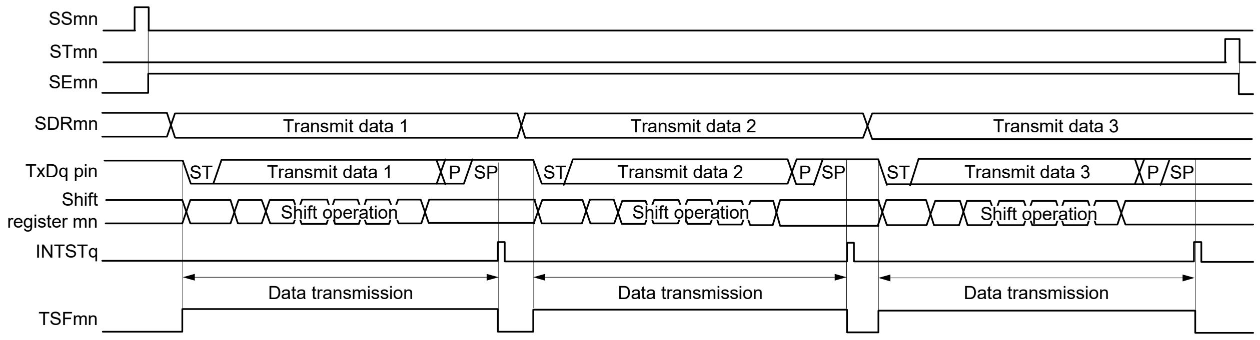
Figure 14 - 110 Procedure for Resuming UART Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

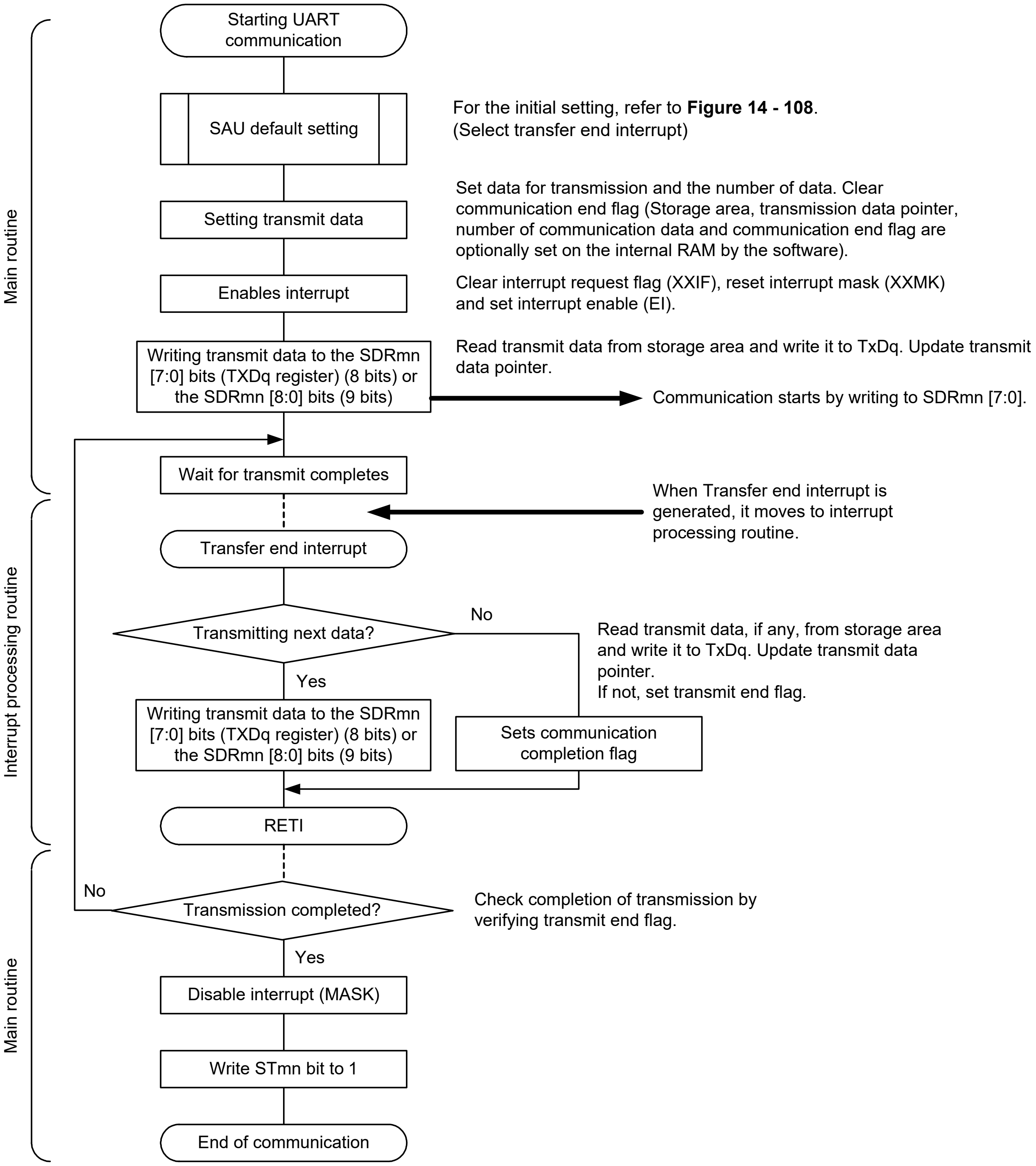
(3) Processing flow (in single-transmission mode)

Figure 14 - 111 Timing Chart of UART Transmission (in Single-Transmission Mode)



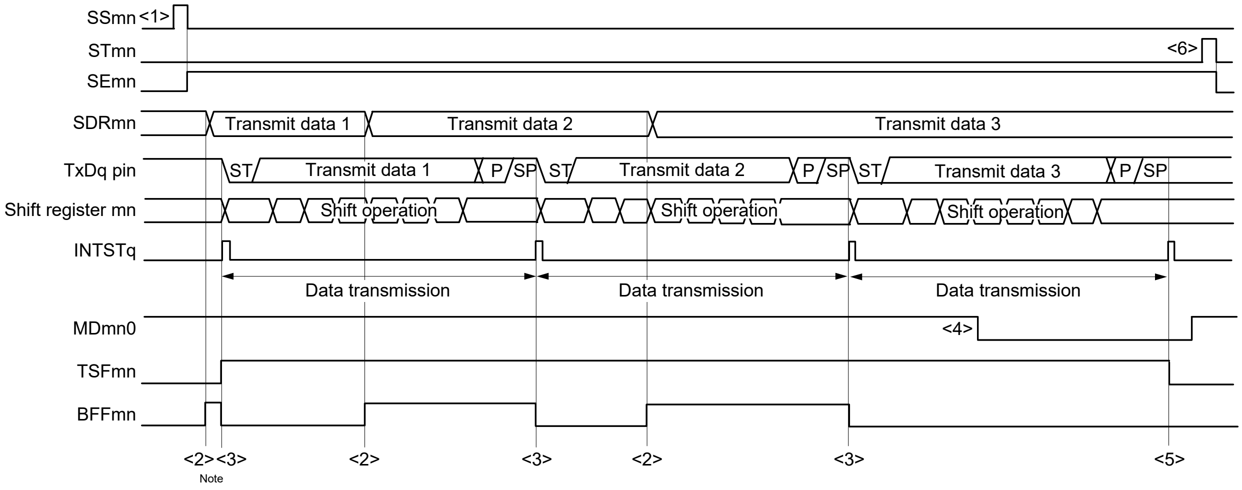
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1)
mn = 00, 02

Figure 14 - 112 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14 - 113 Timing Chart of UART Transmission (in Continuous Transmission Mode)

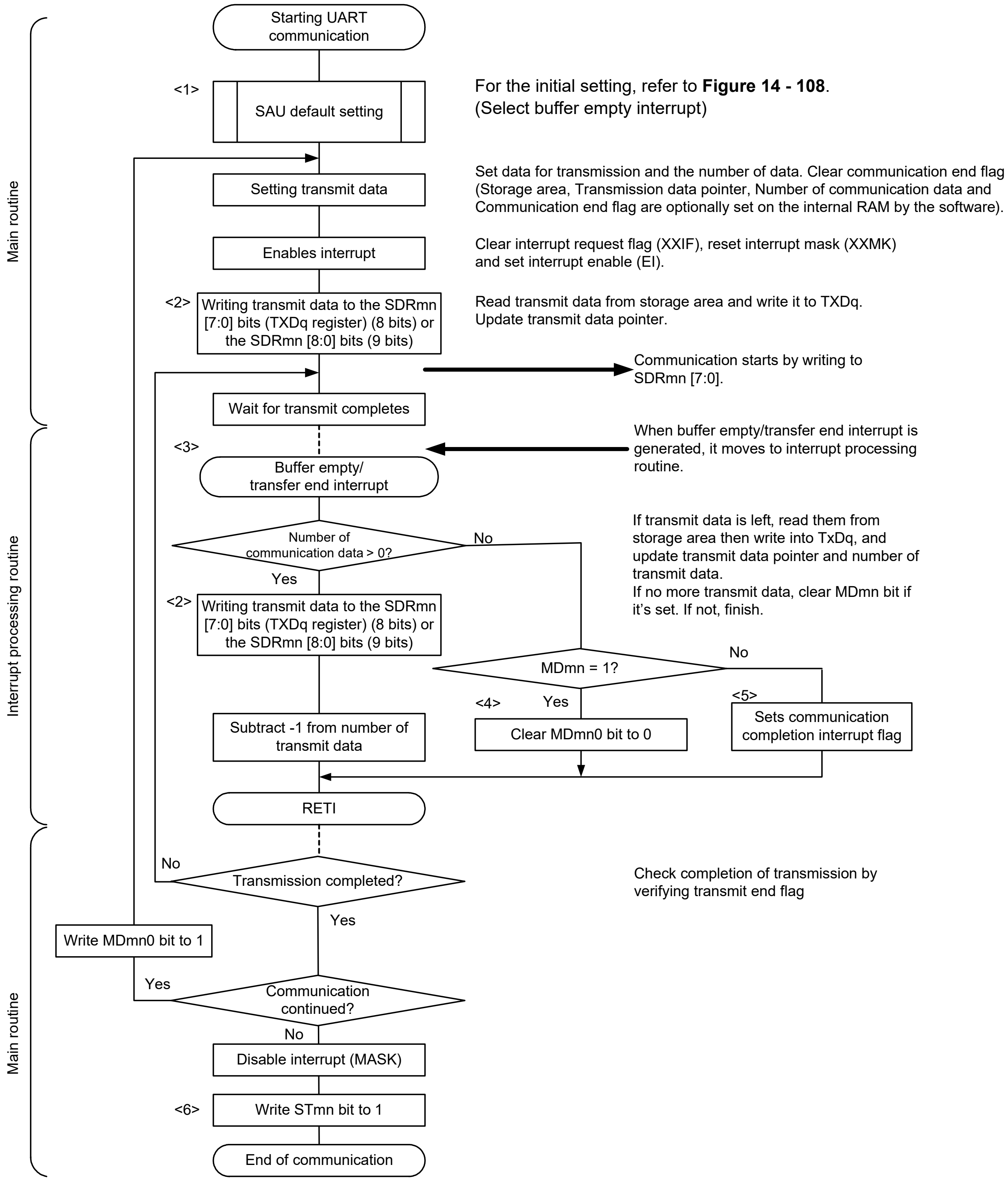


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1)
mn = 00, 02

Figure 14 - 114 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 113 Timing Chart of UART Transmission (in Continuous Transmission Mode).

14.7.2 UART reception

UART reception is an operation wherein the RL78 microcontroller asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1
Target channel	Channel 1 of SAU0	Channel 3 of SAU0
Pins used	RxD0	RxD1
Interrupt	INTSR0	INTSR1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	INTSRE1
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)	
Transfer data length	7, 8 or 9 bits Note 1	
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note 2	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity	
Stop bit	Appending 1 bit	
Data direction	MSB or LSB first	

Note 1. Only following UART can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

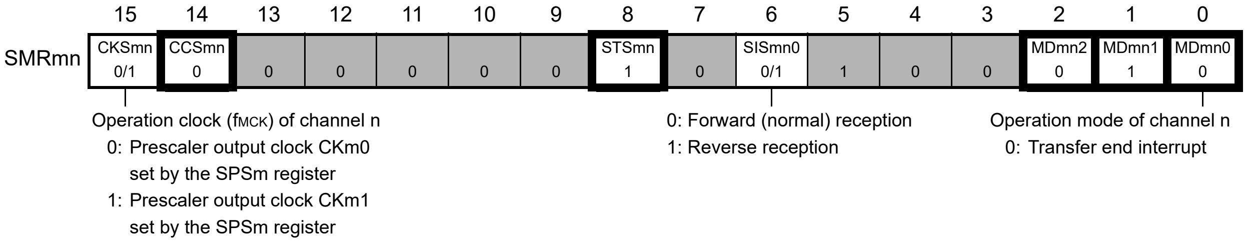
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

(1) Register setting

Figure 14 - 115 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)

(a) Serial mode register mn (SMRmn)



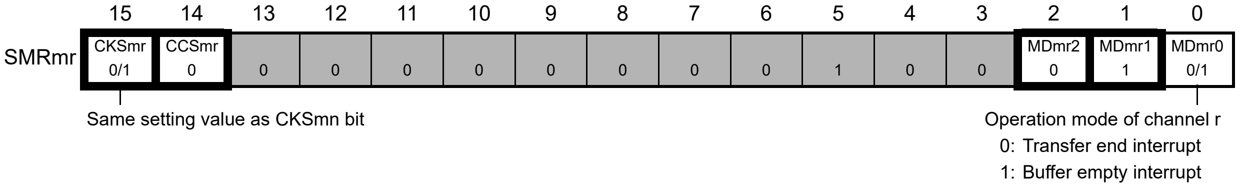
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(b) Serial mode register mr (SMRmr)

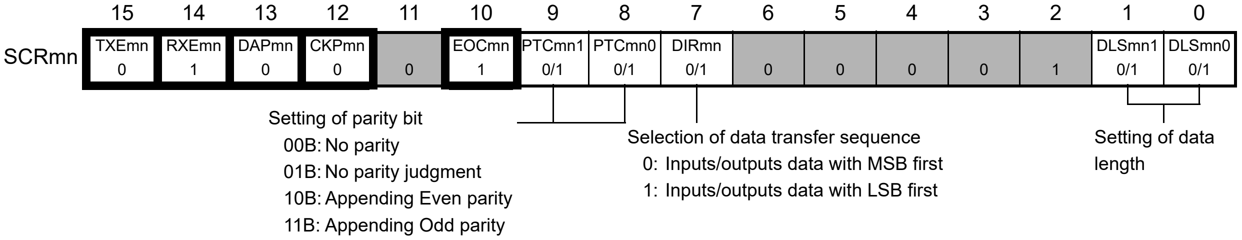


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	0	0	0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn bit

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(c) Serial communication operation setting register mn (SCRmn)



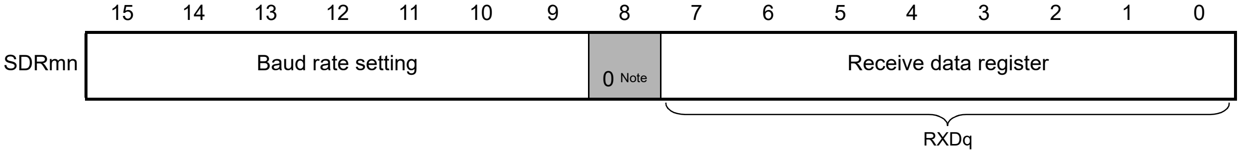
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	0	0	0	1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: No parity judgment
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length

(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0 Note	Receive data register							

Note When performing 9-bit communication, bits 0 to 8 of the SDRm1 register are used to specify the reception data.
▪ UART0

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n - 1), q: UART number (q = 0, 1)

Remark 2. ▣: Setting is fixed in the UART reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14 - 116 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(f) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.

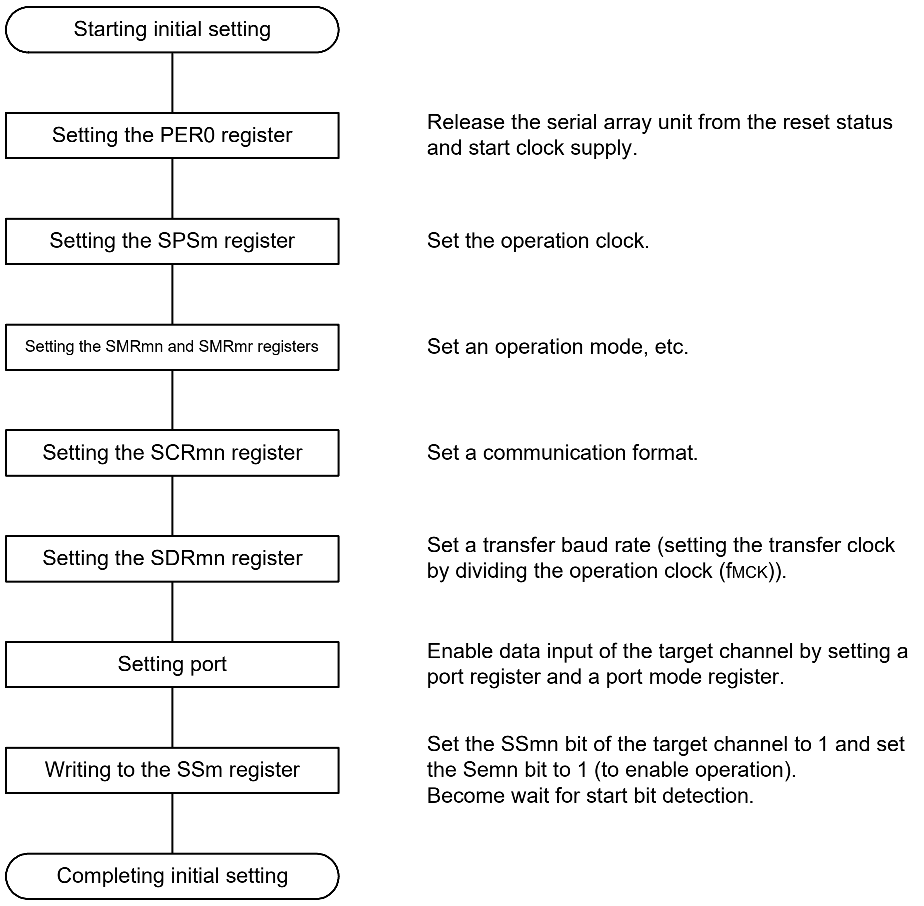
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n - 1), q: UART number (q = 0, 1)

Remark 2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 117 Initial Setting Procedure for UART Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

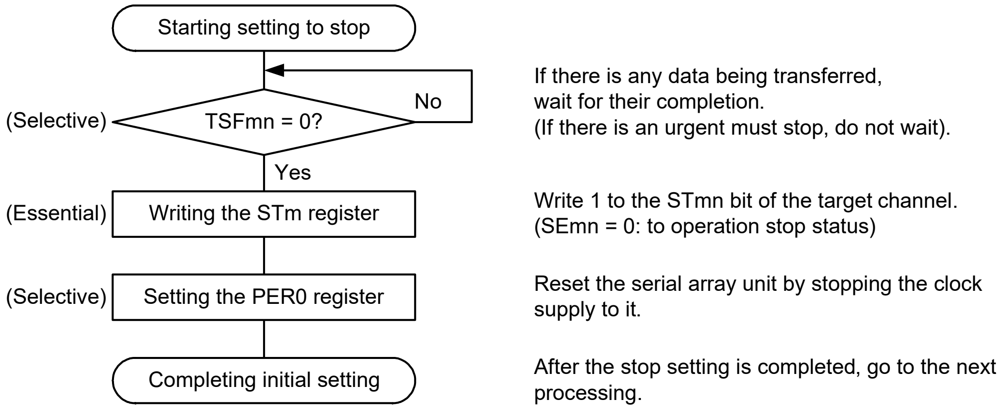
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the Semn bit to 1 (to enable operation).
Become wait for start bit detection.

Caution **Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.**

Figure 14 - 118 Procedure for Stopping UART Reception



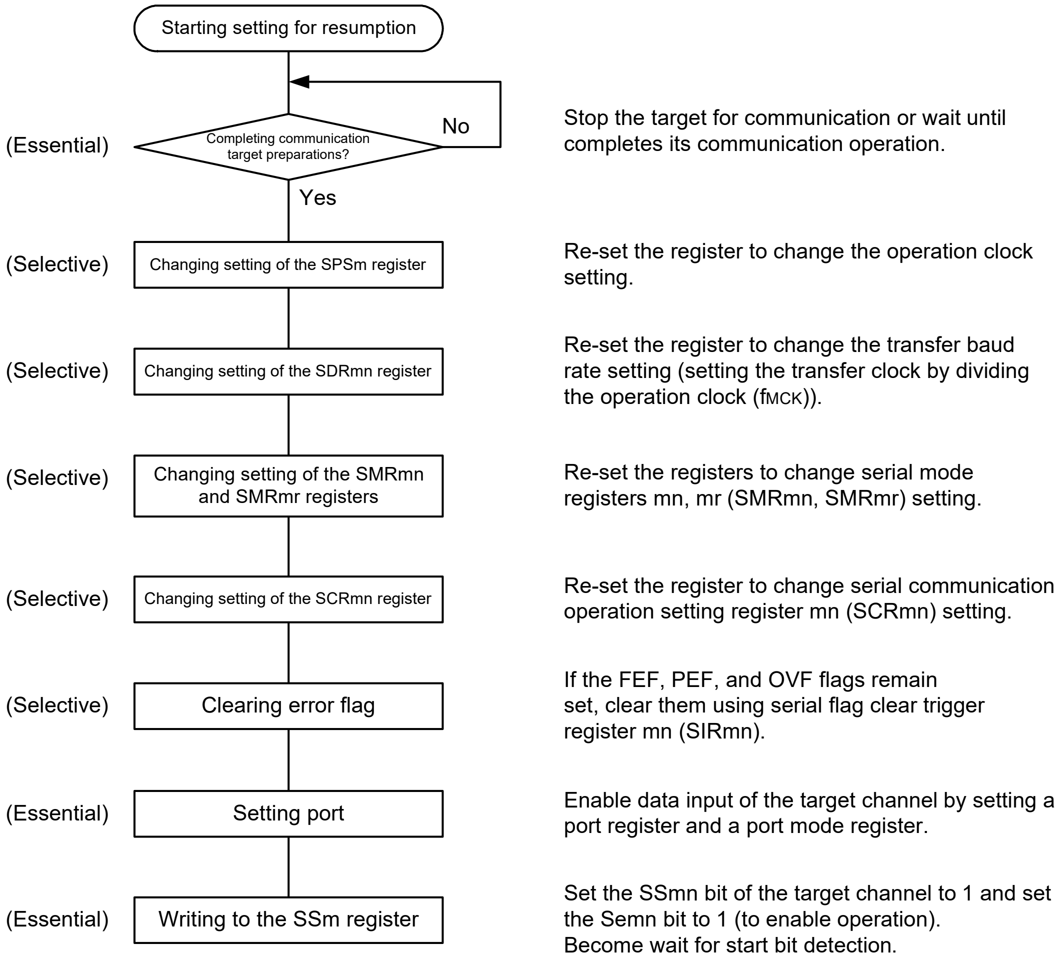
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait).

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 14 - 119 Procedure for Resuming UART Reception

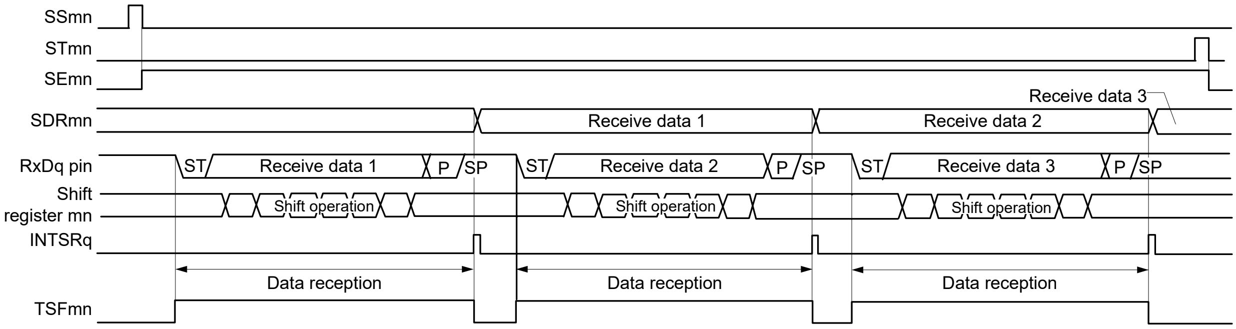


Caution **After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.**

Remark If PER0 is rewritten while stopping the communication target and the clock supply is stopped, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

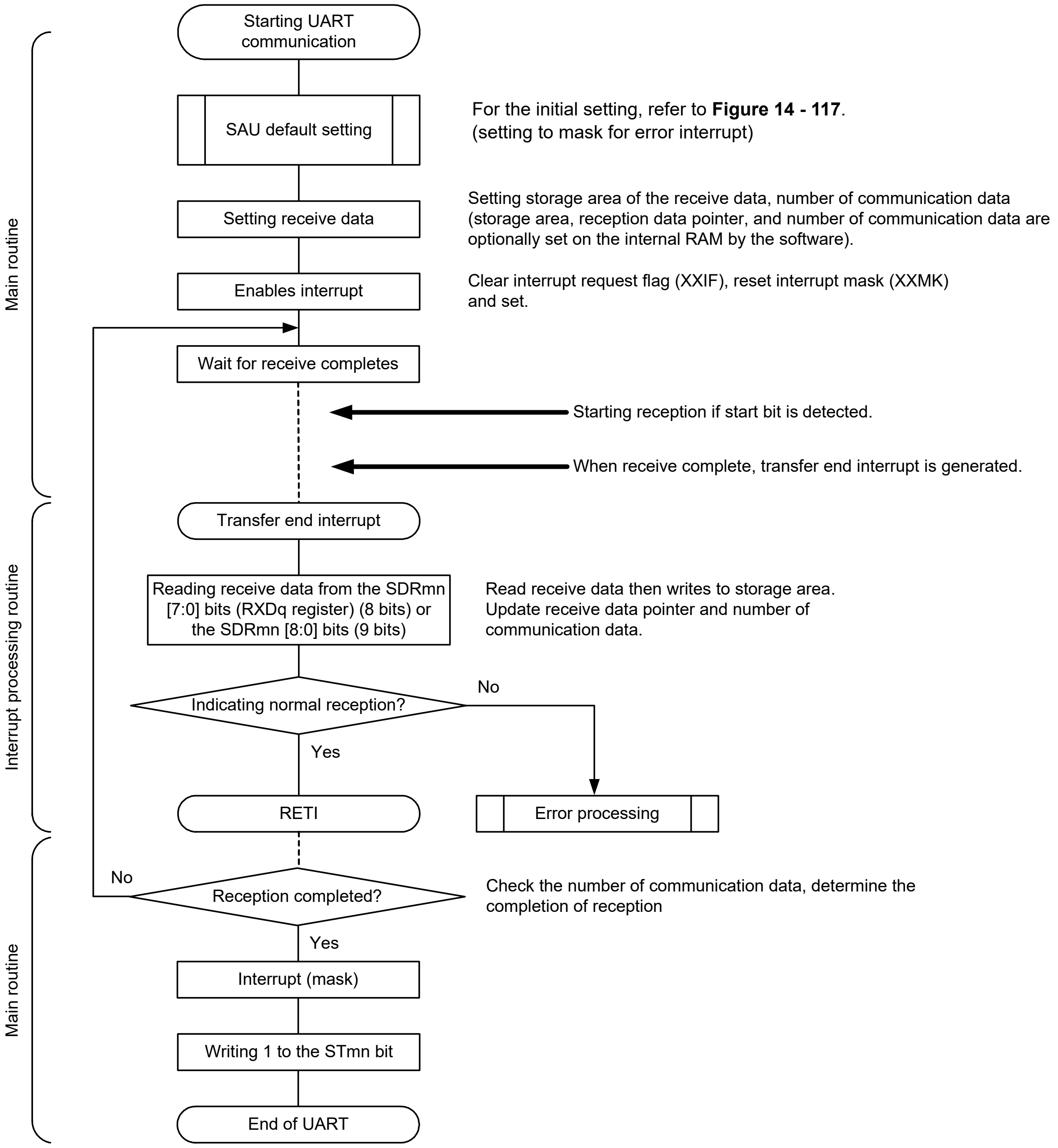
(3) Processing flow

Figure 14 - 120 Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n - 1), q: UART number (q = 0, 1)

Figure 14 - 121 Flowchart of UART Reception



14.7.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only the following UART can be specified when FRQSEL4 in the option byte (000C2H) = 0 in the SNOOZE mode.

- UART0

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 14 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 14 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 14 - 4.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Caution 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

Caution 2. The transfer rate in the SNOOZE mode is only 4800 bps.

Caution 3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal.

Note, however, that reception through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART reception.

Table 14 - 4 Baud Rate Setting for UART Reception in SNOOZE Mode

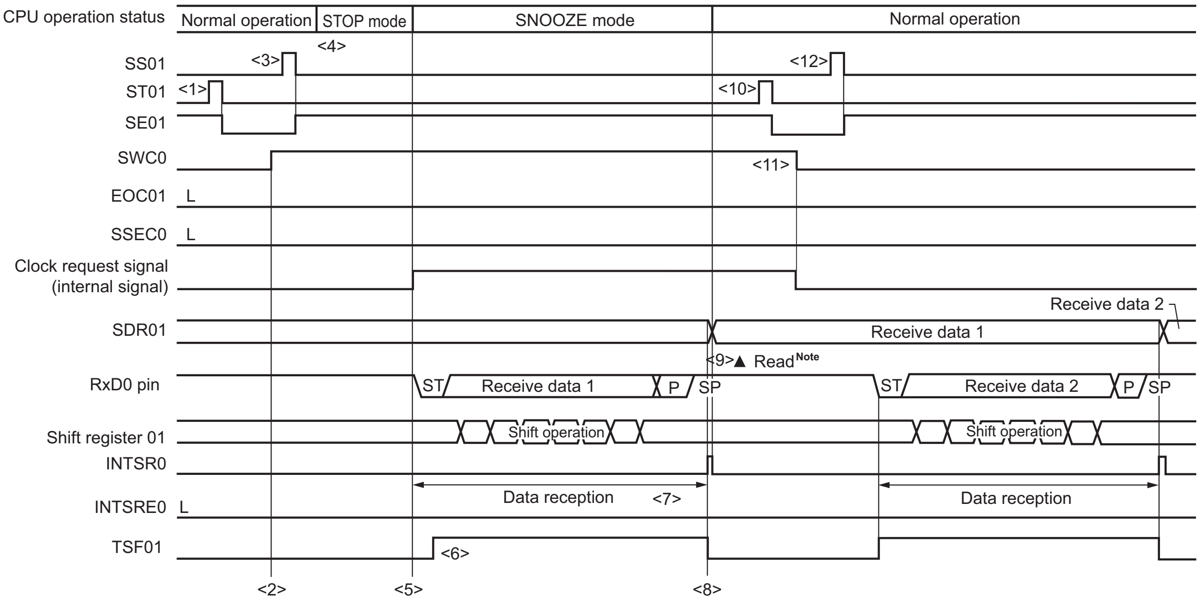
High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz ± 2.0%	$f_{CLK}/2^5$	79	0.60%	- 1.18%
16 MHz ± 2.0%	$f_{CLK}/2^4$	105	1.27%	-0.53%
12 MHz ± 2.0%	$f_{CLK}/2^4$	79	0.60%	-1.19%
8 MHz ± 2.0%	$f_{CLK}/2^3$	105	1.27%	-0.53%
6 MHz ± 2.0%	$f_{CLK}/2^3$	79	0.60%	-1.19%
4 MHz ± 2.0%	$f_{CLK}/2^2$	105	1.27%	-0.53%
3 MHz ± 2.0%	$f_{CLK}/2^2$	79	0.60%	-1.19%
2 MHz ± 2.0%	$f_{CLK}/2$	105	1.27%	-0.54%
1 MHz ± 2.0%	f_{CLK}	105	1.27%	-0.57%

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 14 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

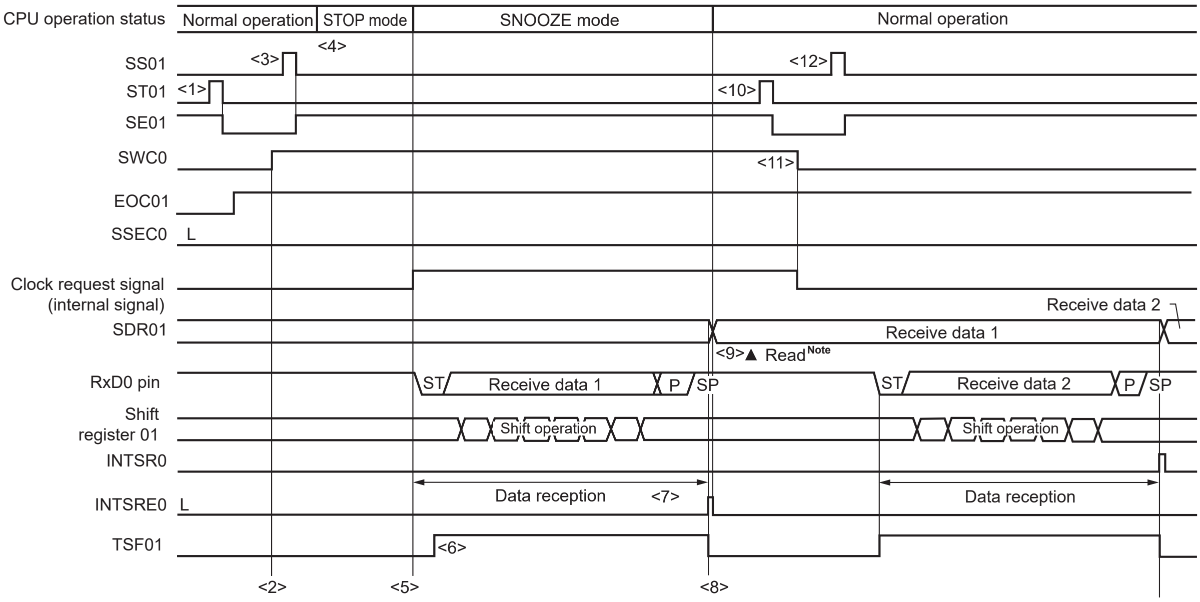
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).**
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 14 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)
Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 14 - 123 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



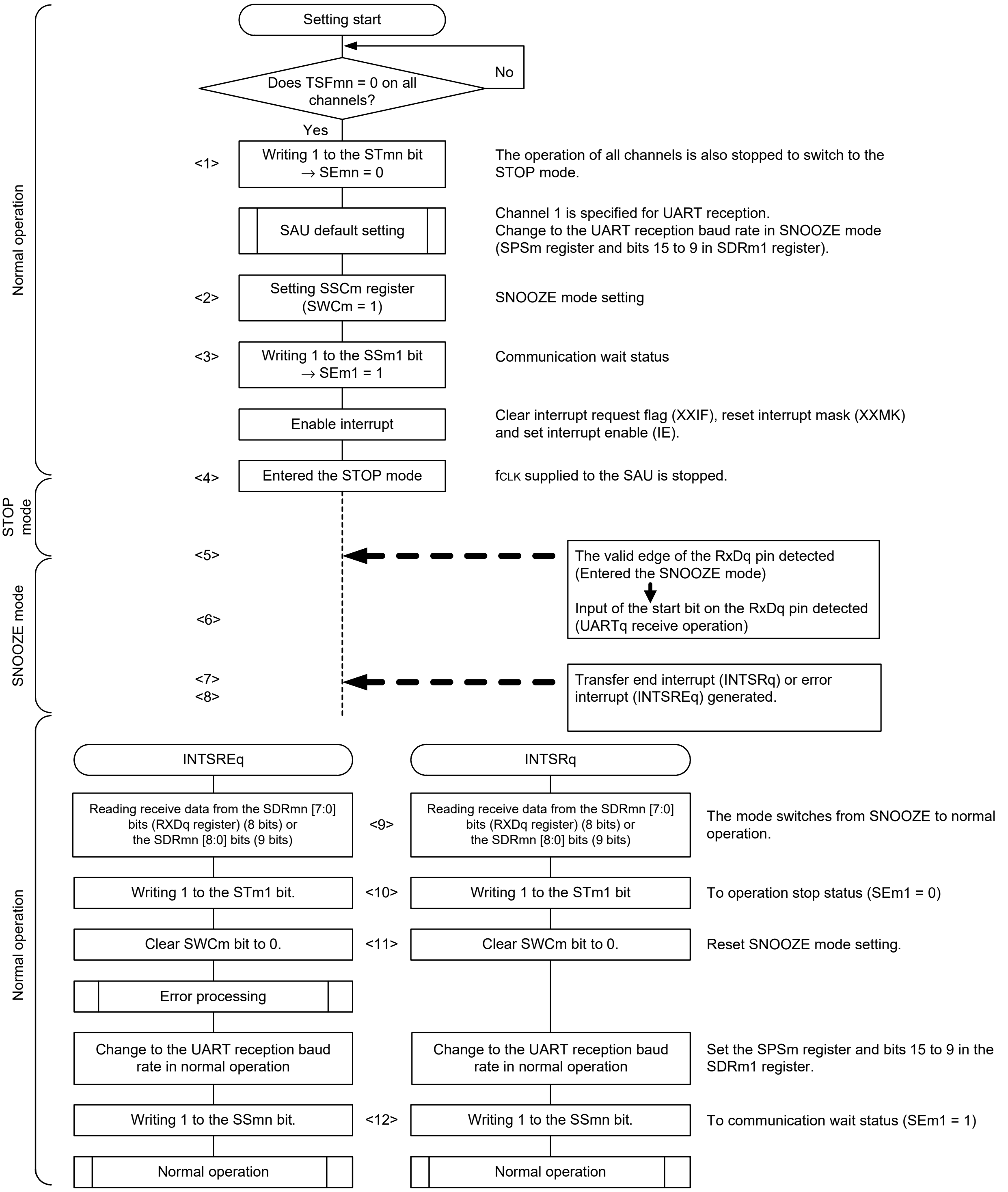
Note Read the received data when SWCm = 1.

Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).**
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 14 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

Figure 14 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



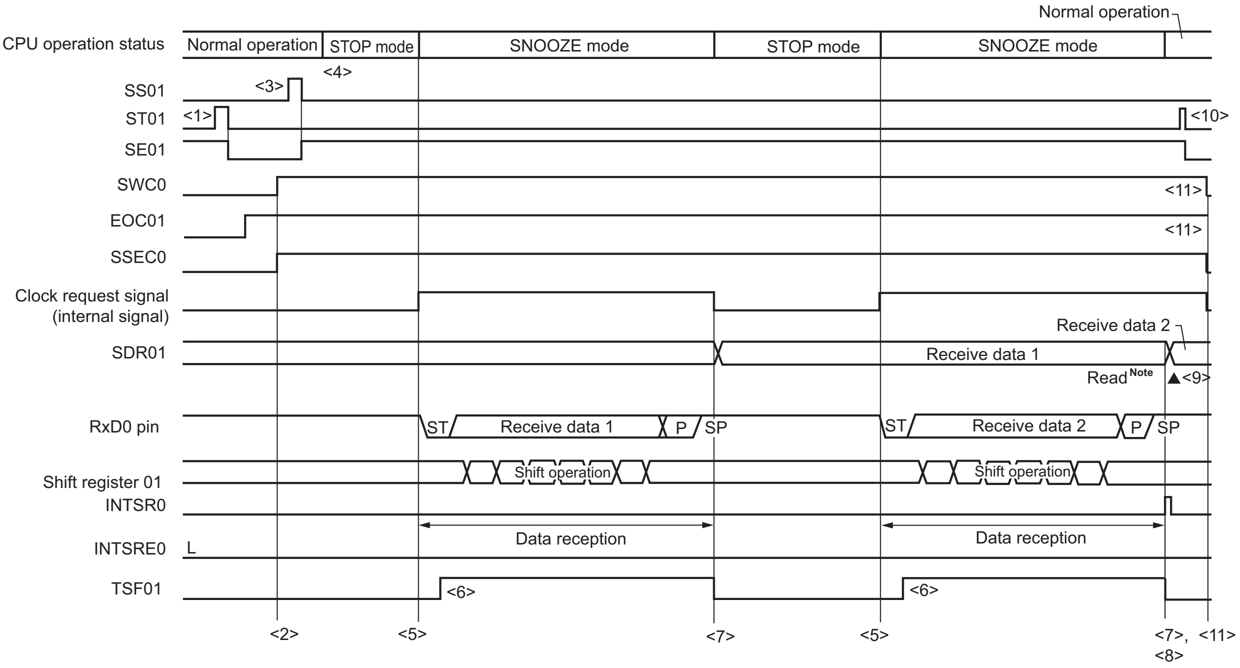
Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 14 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 14 - 123 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0; n = 0 to 3

(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)

Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 14 - 125 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Only read received data while SWCm = 1.

Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).

And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

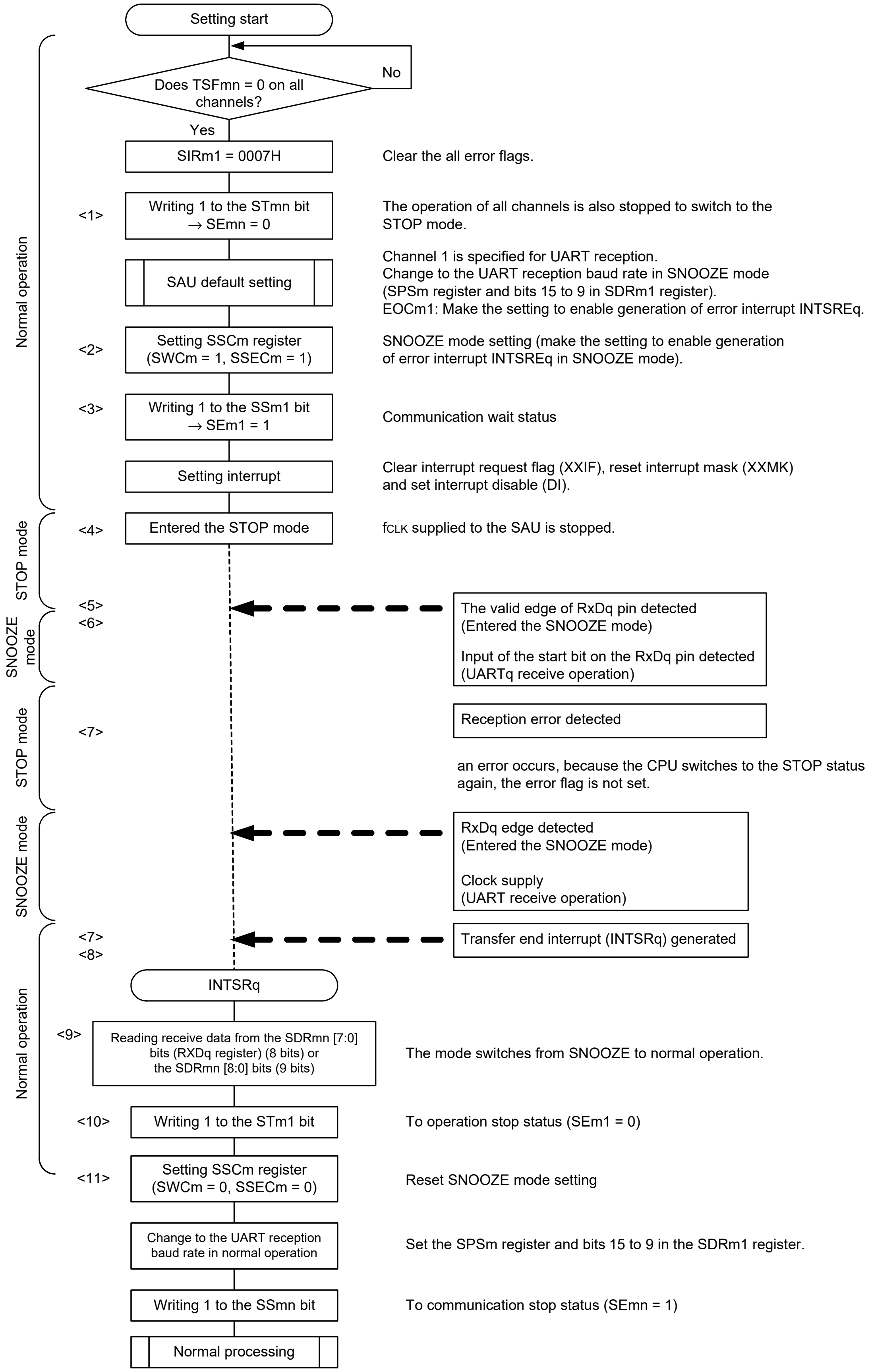
Caution 2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated.

Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 14 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

Figure 14 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



(Caution and Remarks are listed on the next page.)

Caution **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 14 - 125 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0; n = 0 to 3

14.7.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14 - 5 Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.88 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.75 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.88 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 - 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16%
153600 bps	f_{CLK}	64	153846 bps	+0.16%
312500 bps	f_{CLK}	31	312500 bps	±0.0%

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **14.7.4 (1) Baud rate calculation expression**.)

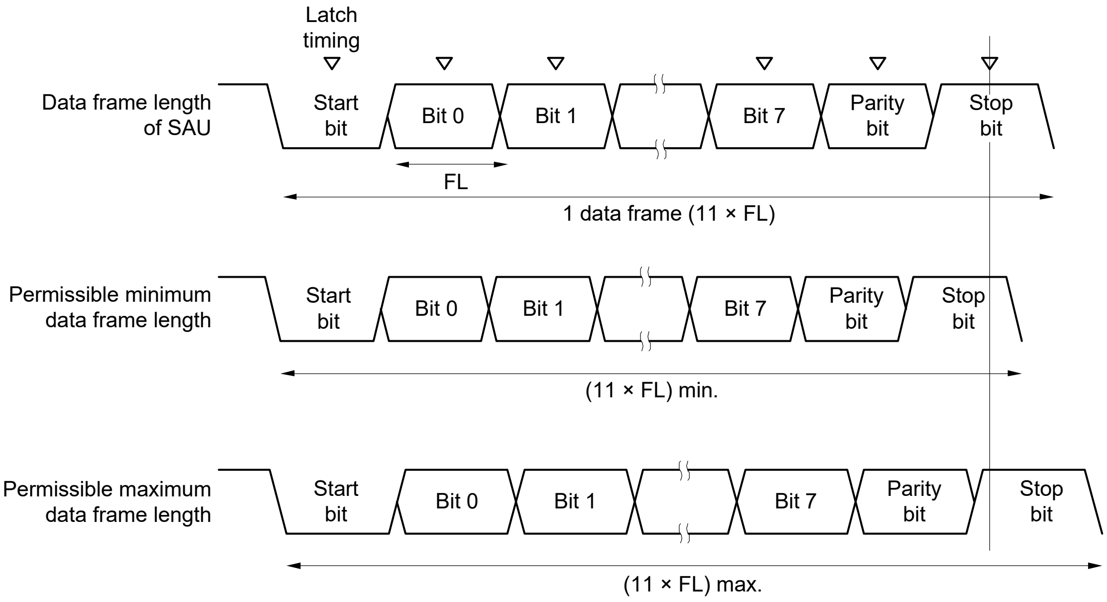
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

Figure 14 - 127 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 14 - 127, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

14.7.5 Procedure for processing errors that occurred during UART (UART0, UART1) communication

The procedure for processing errors that occurred during UART (UART0, UART1) communication is described in Figures 14 - 128 and 14 - 129.

Figure 14 - 128 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn) →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 14 - 129 Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

14.8 Operation of Simplified I^{2}C (IIC00) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]
- Transfer end interrupt

[Error detection flag]
- Overrun error
- ACK error

* [Functions not supported by simplified I^{2}C]
- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **14.8.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

The channel supporting simplified I^{2}C (IIC00) is channel 0 of SAU0.

- 30, 32, 44-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supporting slave select input)	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

Simplified I^{2}C (IIC00) performs the following four types of communication operations.

- Address field transmission (See **14.8.1**.)
- Data transmission (See **14.8.2**.)
- Data reception (See **14.8.3**.)
- Stop condition generation (See **14.8.4**.)

14.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^{2}C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^{2}C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00 Note 1
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error detection flag (PEFmn)
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)
Transfer rate Note 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^{2}C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Non-reversed output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

Note 1. To perform communication via simplified I^{2}C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Setting for Used Port and Alternate Functions**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers**.

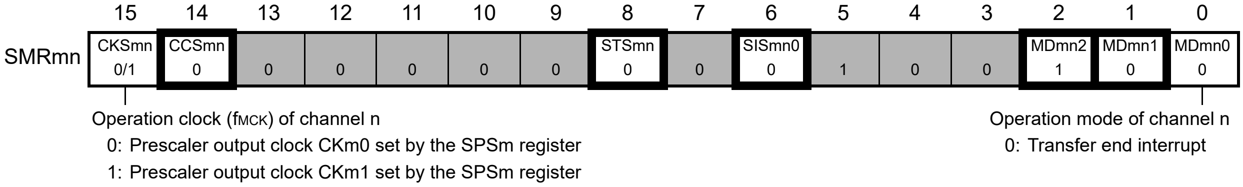
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 130 Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC00)

(a) Serial mode register mn (SMRmn)

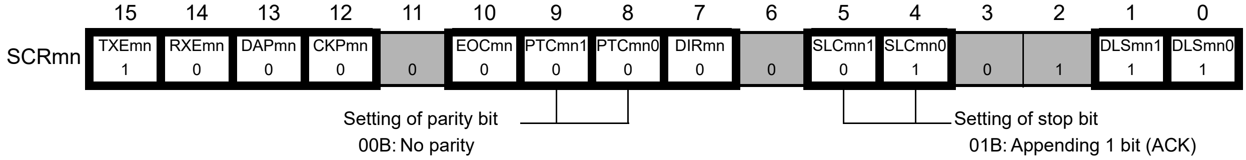


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Operation mode of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

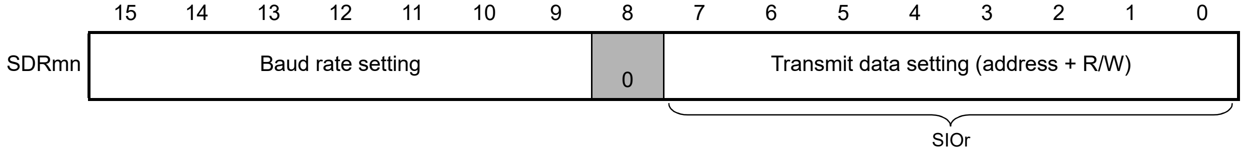


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

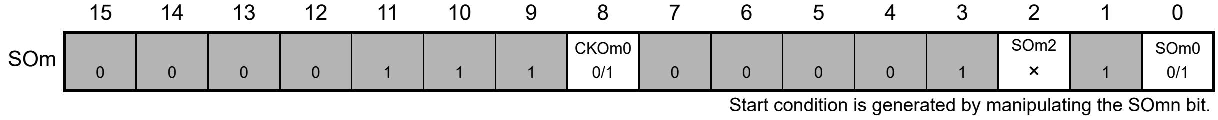
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							

SIOr

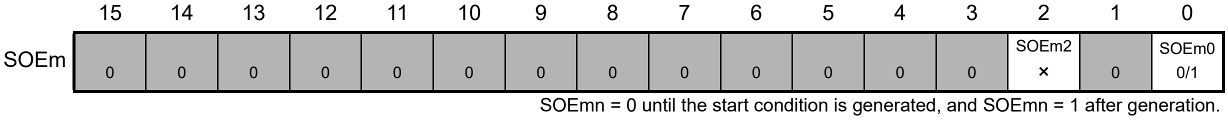
(d) Serial output register m (SOm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

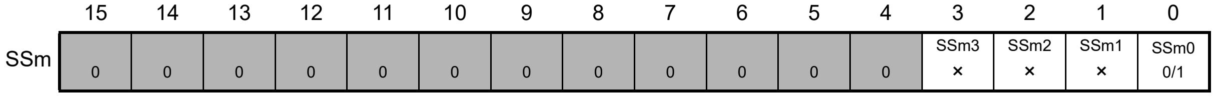
(e) Serial output enable register m (SOEm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.



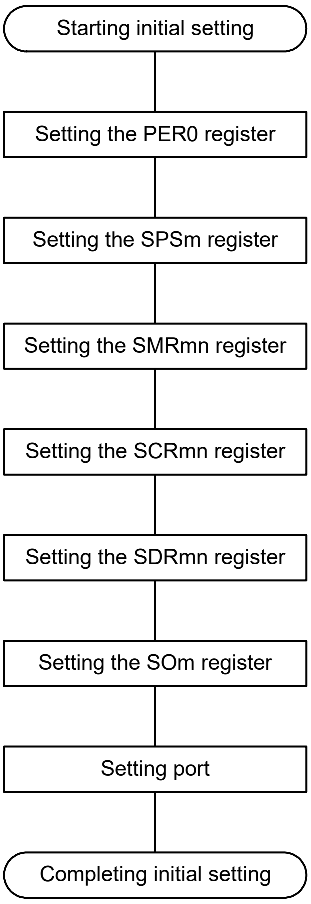
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

Remark 2. □: Setting is fixed in the IIC mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 14 - 131 Initial Setting Procedure for Address Field Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

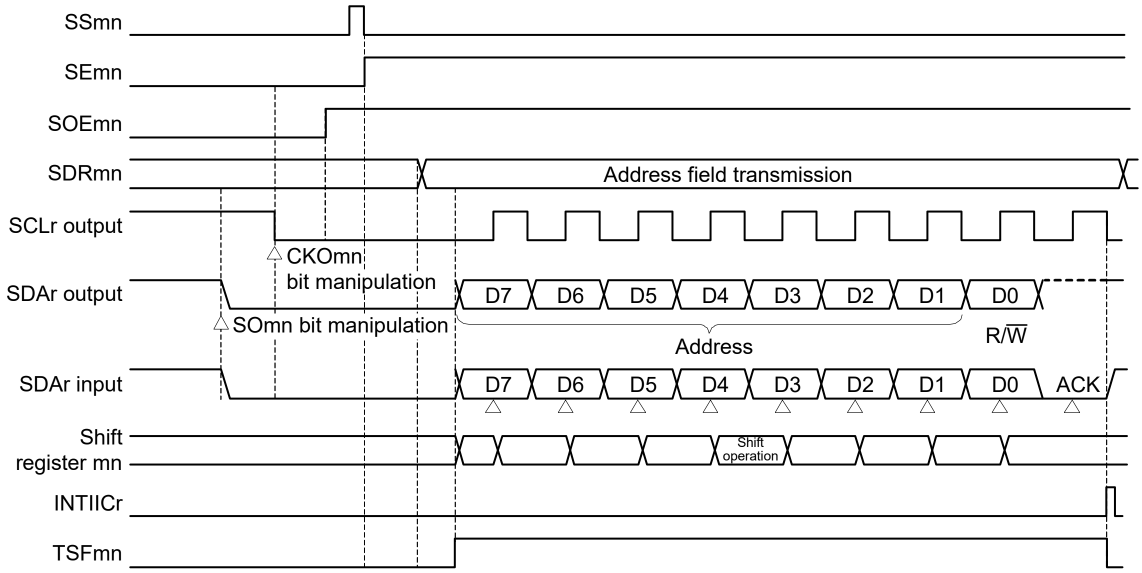
Set the initial output level (1) of the serial data (SOmn) and serial clock (CKOmn).

Enable data output, clock output, and N-ch open-drain output (V_{DD} tolerance) mode of the target channel by setting the port register, port mode register, and port output mode register.

Remark At the end of the initial setting, the simplified I^{2}C (IIC00) must be set so that output is disabled and operations are stopped.

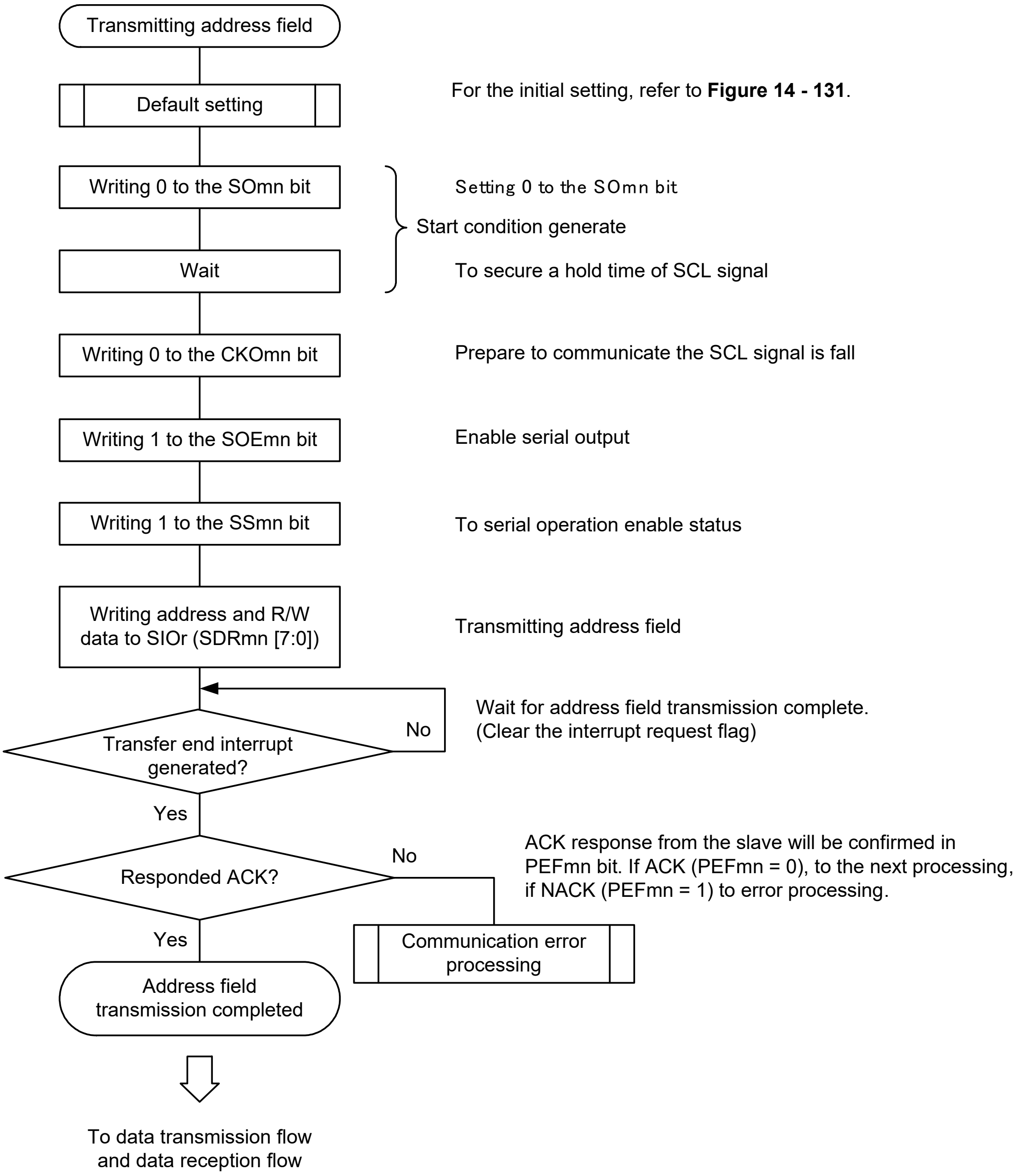
(3) Processing flow

Figure 14 - 132 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

Figure 14 - 133 Flowchart of Address Field Transmission



14.8.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00 Note 1
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error flag (PEFmn)
Transfer data length	8 bits
Transfer rate Note 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Non-reverse output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

Note 1. To perform communication via simplified I²C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Setting for Used Port and Alternate Functions**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 134 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)...During data transmission/reception, valid only lower 8-bits (SIOr).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 1							0	Transmit data setting							
									SIOr							

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1 Note 2	0	0	0	0	1	SOm2 ×	1	SOm0 0/1 Note 2

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Note 1. Because the setting is completed by address field transmission, setting is not required.

Note 2. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

Remark 2. □: Setting is fixed in the IIC mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 14 - 135 Timing Chart of Data Transmission

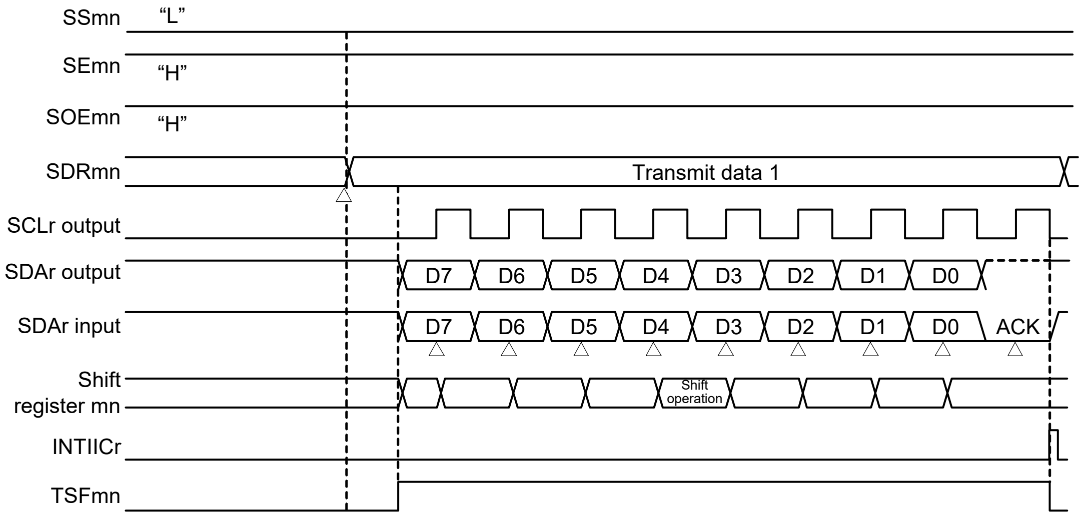
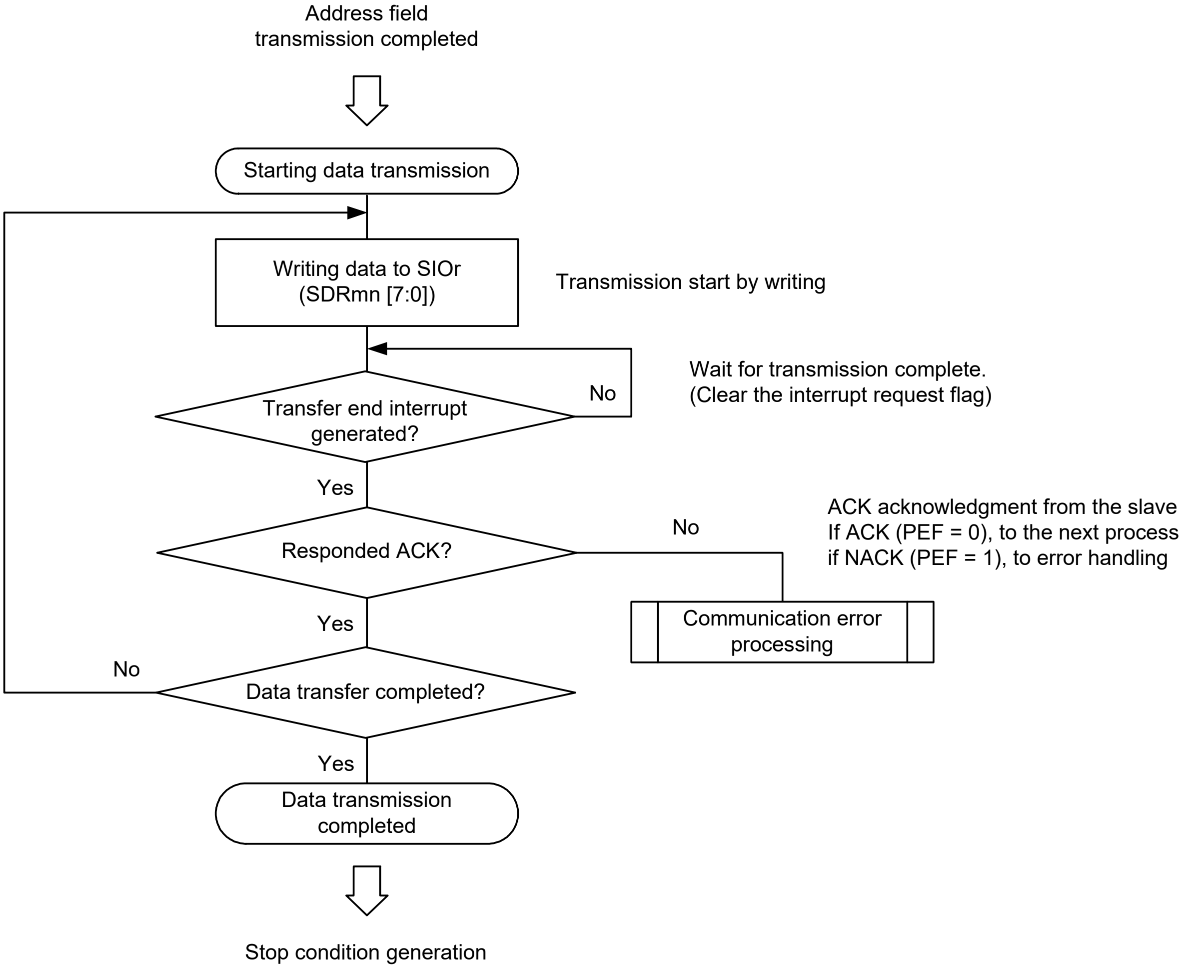


Figure 14 - 136 Flowchart of Data Transmission



14.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00 Note 1
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	8 bits
Transfer rate Note 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Non-reverse output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (ACK transmission)
Data direction	MSB first

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Setting for Used Port and Alternate Functions**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 14 - 137 Example of Contents of Registers for Data Reception of Simplified I^2C (IIC00)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 1							0	Dummy transmit data setting (FFH)							
									SIOr							

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	1	CKOm0 0/1 Note 2	0	0	0	0	1	SOm2 0/1 Note 2	1	SOm0 0/1 Note 2

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.

Note 2. The value varies depending on the communication data during communication operation.

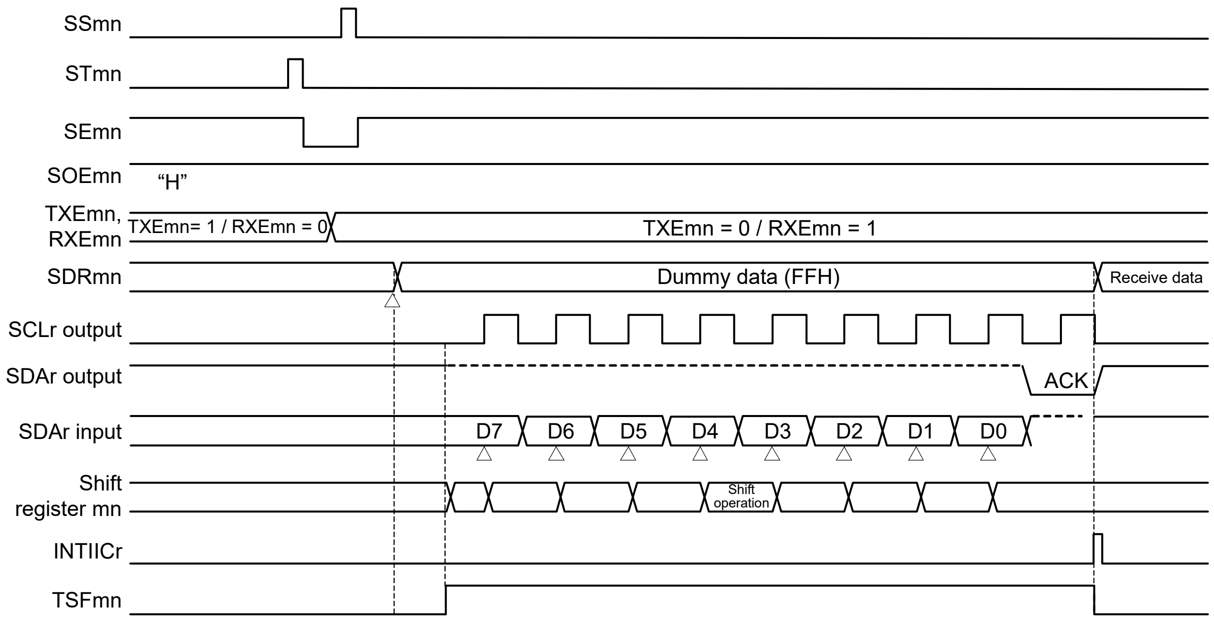
Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

Remark 2. □: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

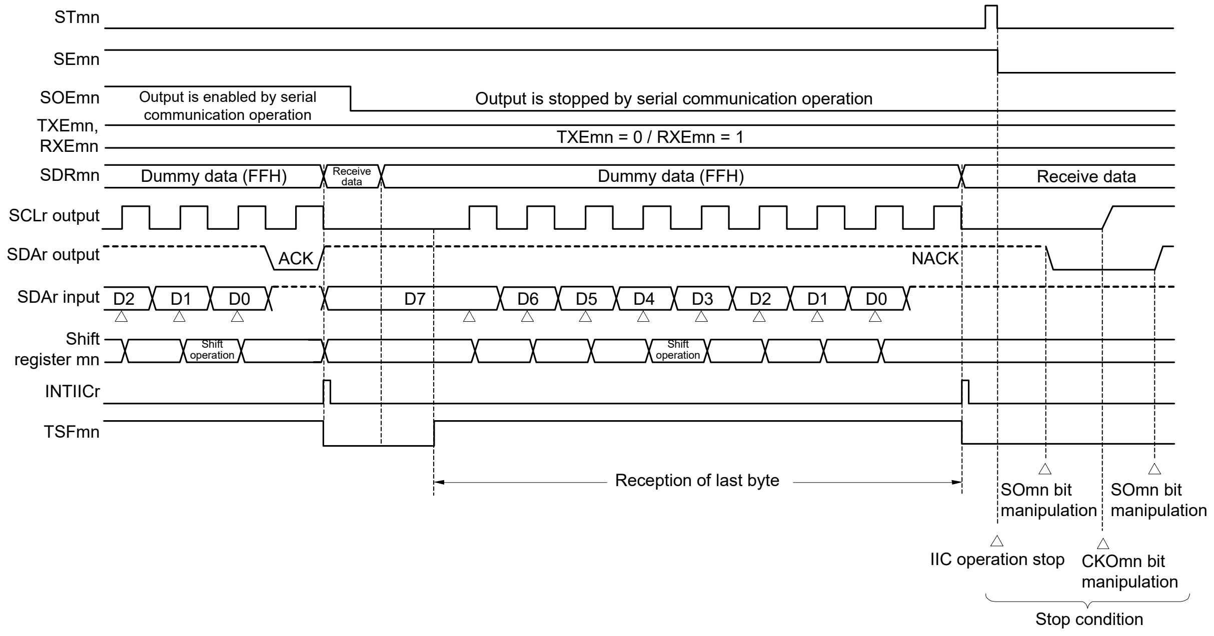
(2) Processing flow

Figure 14 - 138 Timing Chart of Data Reception

(a) When starting data reception

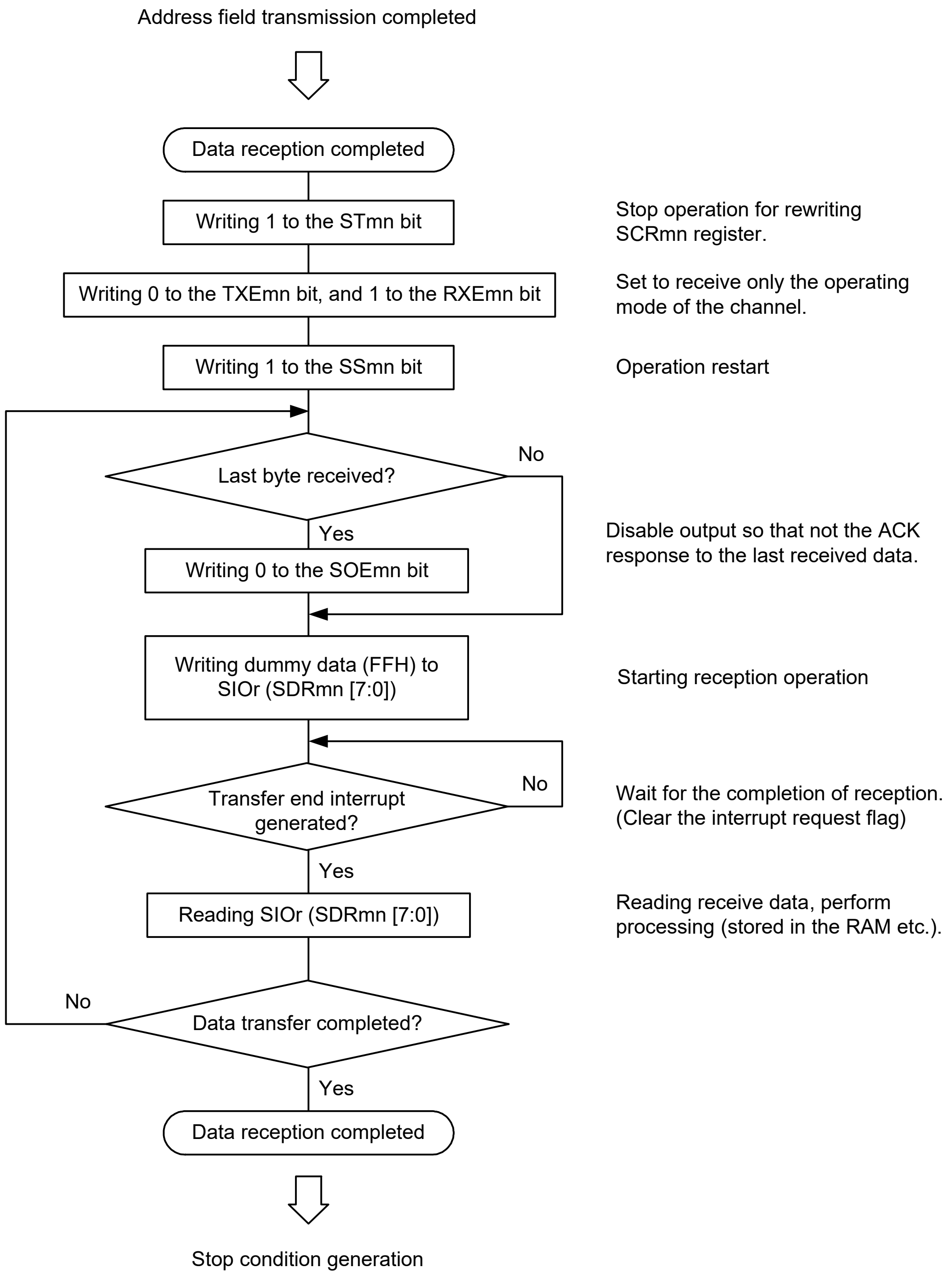


(b) When receiving last data



Remark m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

Figure 14 - 139 Flowchart of Data Reception



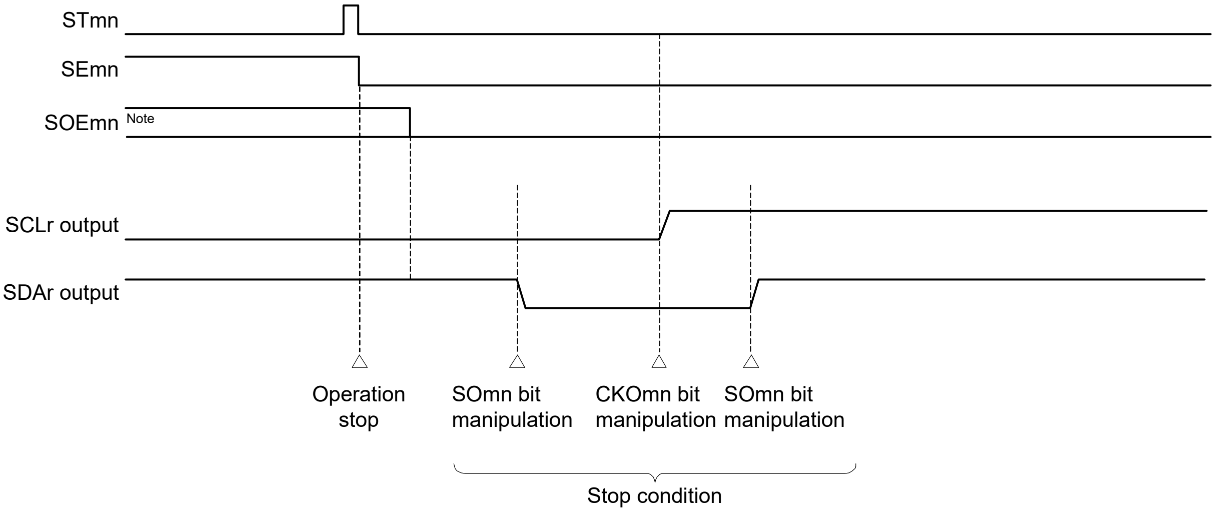
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting “1” to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

14.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

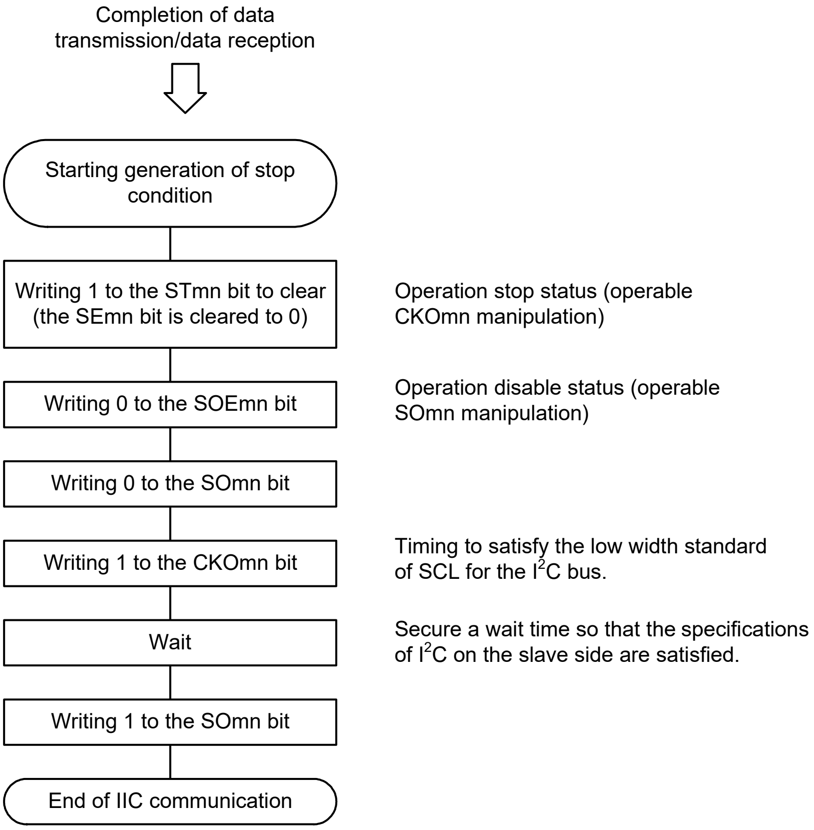
(1) Processing flow

Figure 14 - 140 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 14 - 141 Flowchart of Stop Condition Generation



14.8.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC00) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 00000000B. Be sure to set a value of 00000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I^2C is 50%. The I^2C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I^2C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I^2C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14 - 6 Selection of Operation Clock For Simplified I2C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.88 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.75 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.88 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.72 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don’t care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK}$ = 24 MHz.

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	31	375 kHz	6.25% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

14.8.6 Procedure for processing errors that occurred during simplified I^{2}C (IIC00) communication

The procedure for processing errors that occurred during simplified I^{2}C (IIC00) communication is described in **Figures 14 - 142** and **14 - 143**.

Figure 14 - 142 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 14 - 143 Processing Procedure in Case of ACK Error in Simplified I^{2}C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0), r: IIC number (r = 00)
mn = 00

CHAPTER 15 EVENT LINK CONTROLLER (ELC)

15.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

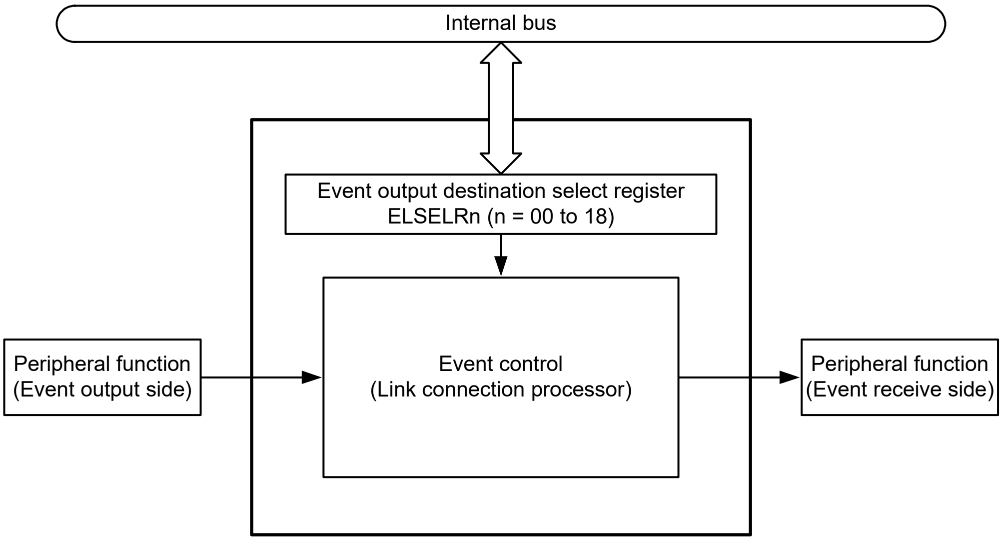
The ELC has the following functions.

- Capable of directly linking event signals from 19 types (44-pin products) or 18 types (30- and 32-pin products) of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of 6 types (30, 32, and 44-pin products) of peripheral functions

15.2 Configuration of ELC

Figure 15 - 1 shows the ELC Block Diagram.

Figure 15 - 1 ELC Block Diagram



15.3 Registers Controlling ELC

Table 15 - 1 lists the Registers Controlling ELC.

Table 15 - 1 Registers Controlling ELC

Register name	Symbol
Event output destination select register 00	ELSELR00
Event output destination select register 01	ELSELR01
Event output destination select register 02	ELSELR02
Event output destination select register 03	ELSELR03
Event output destination select register 04	ELSELR04
Event output destination select register 05	ELSELR05
Event output destination select register 06	ELSELR06
Event output destination select register 07	ELSELR07
Event output destination select register 08	ELSELR08
Event output destination select register 09	ELSELR09
Event output destination select register 10	ELSELR10
Event output destination select register 11	ELSELR11
Event output destination select register 12	ELSELR12
Event output destination select register 13	ELSELR13
Event output destination select register 14	ELSELR14
Event output destination select register 15	ELSELR15
Event output destination select register 16	ELSELR16
Event output destination select register 17	ELSELR17
Event output destination select register 18	ELSELR18

15.3.1 Event output destination select register n (ELSELRn) (n = 00 to 18)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 15 - 2 lists the Correspondence between ELSELRn (n = 00 to 18) Registers and Peripheral Functions, and Table 15 - 3 lists the Correspondence between Values Set to ELSELRn (n = 00 to 18) Registers and Operation of Link Destination Peripheral Functions at Reception.

Figure 15 - 2 Format of Event output destination select register n (ELSELRn)

Address: F0300H (ELSELR00) to F0312H (ELSELR18) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	Event link disabled
0	0	1	Select operation of peripheral function to link Note
0	1	0	Select operation of peripheral function to link Note
0	1	1	Select operation of peripheral function to link Note
1	0	0	Select operation of peripheral function to link Note
1	0	1	Select operation of peripheral function to link Note
1	1	0	Select operation of peripheral function to link Note
Other than above			Setting prohibited

Note See **Table 15 - 3 Correspondence between Values Set to ELSELRn (n = 00 to 18) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 15 - 2 Correspondence between ELSELRn (n = 00 to 18) Registers and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01	External interrupt edge detection 1	INTP1
ELSELR02	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	External interrupt edge detection 4	INTP4
ELSELR05	External interrupt edge detection 5	INTP5
ELSELR06 Note	Key return signal detection	INTKR
ELSELR07	Timer RD0 Input capture A/Compare match A	INTTRD0
ELSELR08	Timer RD0 Input capture B/Compare match B	INTTRD0
ELSELR09	Timer RD1 Input capture A/Compare match A	INTTRD1
ELSELR10	Timer RD1 Input capture B/Compare match B	INTTRD1
ELSELR11	Timer RD1 Underflow	TRD1 underflow signal
ELSELR12	Timer RJ0 Underflow	INTTRJ0
ELSELR13	TAU channel 00 Count end/Capture end	INTTM00
ELSELR14	TAU channel 01 Count end/Capture end	INTTM01
ELSELR15	TAU channel 02 Count end/Capture end	INTTM02
ELSELR16	TAU channel 03 Count end/Capture end	INTTM03
ELSELR17	Comparator detection 0	INTCMP0
ELSELR18	Comparator detection 1	INTCMP1

Note In 44-pin products only.

Table 15 - 3 Correspondence between Values Set to ELSELRn (n = 00 to 18) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSEL2 to ELSEL0 in ELSELRn Register	Link Destination Peripheral Function	Operation When Receiving Event
001B	A/D converter	A/D conversion starts
010B	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
011B	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
100B	Timer RJ0	Count source
101B	Timer RD0	TRDIOD0 input capture, pulse output forced cutoff
110B	Timer RD1	TRDIOD1 input capture, pulse output forced cutoff

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

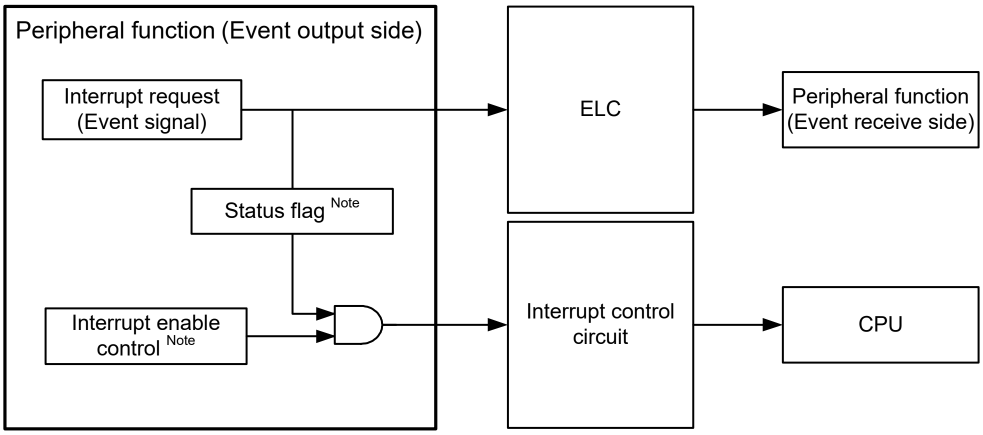
15.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 15 - 3 shows the Relationship between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (See **Table 15 - 3 Correspondence between Values Set to ELSELRn (n = 00 to 18) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 15 - 3 Relationship between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 15 - 4 lists the Response of Peripheral Functions That Receive Events.

Table 15 - 4 Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
4	Timer RJ	Count source	An event from the ELC is directly used as the count source of timer RJ.
5	Timer RD0	TRDIOD0 input capture	A count start trigger is generated 2 or 3 cycles of the timer RD operating clock after an ELC event is generated.
		Pulse output forced cutoff	The pulse is forcibly cut off 2 or 3 cycles of the timer RD operating clock after an ELC event is generated.
6	Timer RD1	TRDIOD1 input capture	A count start trigger is generated 2 or 3 cycles of the timer RD operating clock after an ELC event is generated.
		Pulse output forced cutoff	The pulse is forcibly cut off 2 or 3 cycles of the timer RD operating clock after an ELC event is generated.

CHAPTER 16 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.
The number of interrupt sources differs, depending on the product.

		30, 32-pin	44-pin
Maskable interrupts	External	6	7
	Internal	20	20

16.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts
These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).
Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. Default priority, see **Tables 16 - 1** and **16 - 2**.
A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.
External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt
This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Tables 16 - 1** and **16 - 2**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 16 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority [Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type [Note 2]	44-pin	32-pin	30-pin
		Name	Trigger						
Maskable	0	INTWDTI	Watchdog timer interval [Note 3] (75% of overflow time + 1/2 f_{IL})	Internal	00004H	(A)	√	√	√
	1	INTLVI	Voltage detection [Note 4]		00006H		√	√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√	√
	3	INTP1			0000AH		√	√	√
	4	INTP2			0000CH		√	√	√
	5	INTP3			0000EH		√	√	√
	6	INTP4			00010H		√	√	√
	7	INTP5			00012H		√	√	√
	8	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end	Internal	0001EH	(A)	√	√	√
	9	INTSR0	UART0 reception transfer end		00020H		√	√	√
	10	INTSRE0	UART0 reception communication error occurrence		00022H		√	√	√
		INTTM01H	End of timer channel 1 count or capture (at 8-bit timer operation)				√	√	√
	11	INTST1	UART1 transmission transfer end or buffer empty interrupt		00024H		√	√	√
	12	INTSR1	UART1 reception transfer end		00026H		√	√	√
	13	INTSRE1	UART1 reception communication error occurrence		00028H		√	√	√
		INTTM03H	End of timer channel 3 count or capture (at 8-bit timer operation)				√	√	√
	14	INTTM00	End of timer channel 0 count or capture		0002CH		√	√	√
	15	INTTM01	End of timer channel 1 count or capture		0002EH		√	√	√
	16	INTTM02	End of timer channel 2 count or capture		00030H		√	√	√
	17	INTTM03	End of timer channel 3 count or capture		00032H		√	√	√
	18	INTAD	End of A/D conversion		00034H		√	√	√
	19	INTIT	Interval signal detection		00038H		√	√	√
	20	INTKR	Key return signal detection	External	0003AH	(C)	√	×	×

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 26 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 16 - 1.

Note 3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 16 - 2 Interrupt Source List (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	44-pin	32-pin	30-pin
		Name	Trigger						
Maskable	21	INTTRJ0	Timer RJ underflow	Internal	00040H	(A)	√	√	√
	22	INTCMP0	Comparator detection 0		00052H		√	√	√
	23	INTCMP1	Comparator detection 1		00054H		√	√	√
	24	INTTRD0	Timer RD0 input capture, compare match, overflow, underflow interrupt		00056H		√	√	√
	25	INTTRD1	Timer RD1 input capture, compare match, overflow, underflow interrupt		00058H		√	√	√
	26	INTFL	Reserved Note 5		00062H		√	√	√
Software	—	BRK	Execution of BRK instruction	—	0007EH	(D)	√	√	√
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	√	√	√
		POR	Power-on-reset				√	√	√
		LVD	Voltage detection Note 3				√	√	√
		WDT	Overflow of watchdog timer				√	√	√
		TRAP	Execution of illegal instruction Note 4				√	√	√
		IAW	Illegal-memory access				√	√	√
		RPE	RAM parity error				√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 26 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 16 - 1.

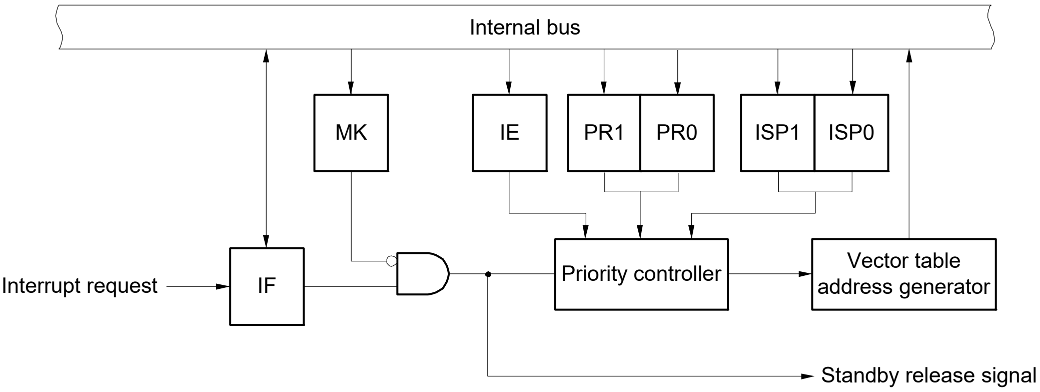
Note 3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

Note 4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

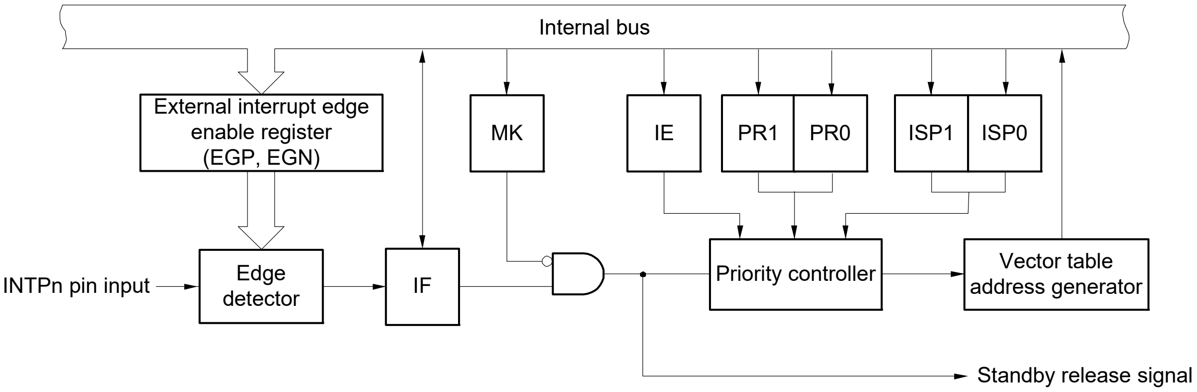
Note 5. Be used at the flash self-programming library.

Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

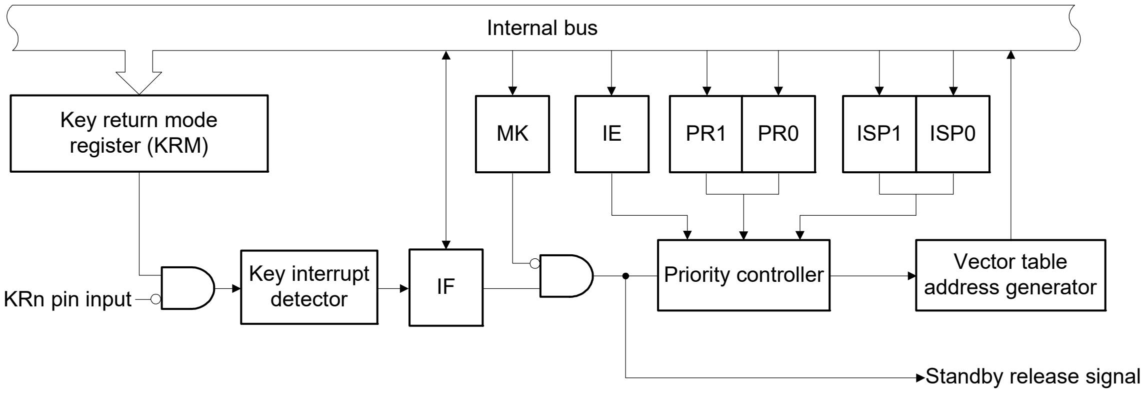


IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

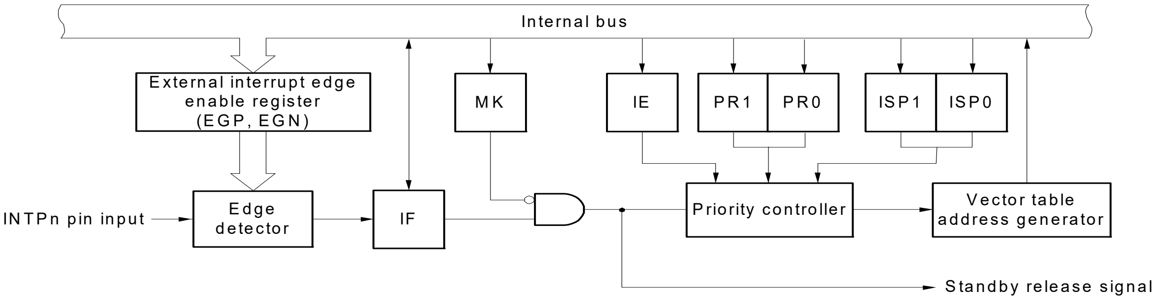
Remark n = 0 to 5

Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

Remark 44-pin: n = 0 to 3

16.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 16 - 3 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16 - 3 Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		44-pin	30, 32-pin
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√
INTST0 Note 1	STIF0 Note 1	IF0H	STMK0 Note 1	MK0H	STPR00, STPR10 Note 1	PR00H, PR10H	√	√
INTCSI00 Note 1	CSIIF00 Note 1		CSIMK00 Note 1		CSIPR000, CSIPR100 Note 1		√	√
INTIIC00 Note 1	IICIF00 Note 1		IICMK00 Note 1		IICPR000, IICPR100 Note 1		√	√
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		√	√
INTSRE0 Note 2	SREIF0 Note 2		SREMK0 Note 2		SREPR00, SREPR10 Note 2		√	√
INTTM01H Note 2	TMIF01H Note 2		TMMK01H Note 2		TMPR001H, TMPR101H Note 2		√	√
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	√	√
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√
INTSRE1 Note 3	SREIF1 Note 3		SREMK1 Note 3		SREPR01, SREPR11 Note 3		√	√
INTTM03H Note 3	TMIF03H Note 3		TMMK03H Note 3		TMPR003H, TMPR103H Note 3		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	×
INTTRJ0	TRJIF0		TRJMK0		TRJPR00, TRJPR10		√	√
INCMP0	CMPIF0	IF2L	CMPMK0	MK2L	CMPPR00, CMPPR10	PR02L, PR12L	√	√
INCMP1	CMPIF1	IF2H	CMPMK1	MK2H	CMPPR01, CMPPR11	PR02H, PR12H	√	√
INTTRD0	TRDIF0		TRDMK0		TRDPR00, TRDPR10		√	√
INTTRD1	TRDIF1		TRDMK1		TRDPR01, TRDPR11		√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√

Note 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE0 or INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers support these two interrupt sources.

Note 3. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 or INTTM03H is generated, bit 2 of the IF1H register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

16.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
IF0H	SREIF0 TMIF01H	SRIF0	STIF0 CSIIF00 IICIF00	0	0	0	0	0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	0	SREIF1 TMIF03H	SRIF1	STIF1

Address: FFFE3H After reset: 00H R/W

Symbol	7	<6>	5	4	<3>	<2>	1	<0>
IF1H	0	TRJIF0	0	0	KRIF	TMKAIF	0	ADIF

Figure 16 - 4 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
IF2L	CMPIF0	0	0	0	0	0	0	0

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
IF2H	FLIF	0	0	0	0	TRDIF1	TRDIF0	CMPIF1

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 16 - 3. Be sure to set bits that are not available to the initial value.

Caution 2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm ("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

16.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.
The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16 - 5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
MK0H	SREMK0 TMMK01H	SRMK0	STMK0 CSIMK00 IICMK00	1	1	1	1	1

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	1	SREMK1 TMMK03H	SRMK1	STMK1

Address: FFFE7H After reset: FFH R/W

Symbol	7	<6>	5	4	<3>	<2>	1	<0>
MK1H	1	TRJMK0	1	1	KRMK	TMKAMK	1	ADMK

Figure 16 - 6 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (2/2)

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	6	5	4	3	2	1	0
MK2L	CMPMK0	1	1	1	1	1	1	1

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
MK2H	FLMK	1	1	1	1	TRDMK1	TRDMK0	CMPMK1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 16 - 3. Be sure to set bits that are not available to the initial value.

16.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.
A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).
The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and the PR12H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12 they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16 - 7 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset:FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PR00H	SREPR00 TMPR001H	SRPR00	STPR00 CSIPR000 IICPR000	1	1	1	1	1

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PR10H	SREPR10 TMPR101H	SRPR10	STPR10 CSIPR100 IICPR100	1	1	1	1	1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	1	SREPR01 TMPR003H	SRPR01	STPR01

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	1	SREPR11 TMPR103H	SRPR11	STPR11

Figure 16 - 8 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	7	<6>	5	4	<3>	<2>	1	<0>
PR01H	1	TRJPR00	1	1	KRPR0	TMKAPR0	1	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	7	<6>	5	4	<3>	<2>	1	<0>
PR11H		TRJPR10	1	1	KRPR1	TMKAPR1	1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	6	5	4	3	2	1	0
PR02L	CMPPR00	1	1	1	1	1	1	1

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	6	5	4	3	2	1	0
PR12L	CMPPR10	1	1	1	1	1	1	1

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
PR02H	FLPR0	1	1	1	1	TRDPR01	TRDPR00	CMPPR01

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
PR12H	FLPR1	1	1	1	1	TRDPR11	TRDPR10	CMPPR11

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution **The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 16 - 3. Be sure to set bits that are not available to the initial value.**

16.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to INTP5.

The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 16 - 9 Format of External Interrupt Rising Edge Enable Registers (EGP0) and External Interrupt Falling Edge Enable Registers (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 16 - 4 shows the Ports Corresponding to EGPn and EGNn bits.

Table 16 - 4 Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	30, 32, 44-pin
EGP0	EGN0	P137	INTP0	√
EGP1	EGN1	P50	INTP1	√
EGP2	EGN2	P51	INTP2	√
EGP3	EGN3	P30	INTP3	√
EGP4	EGN4	P31	INTP4	√
EGP5	EGN5	P16	INTP5	√

Caution **When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge.**
When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark 1. For details on the edge detection ports, see **2.1 Port Function**.
Remark 2. n = 0 to 5

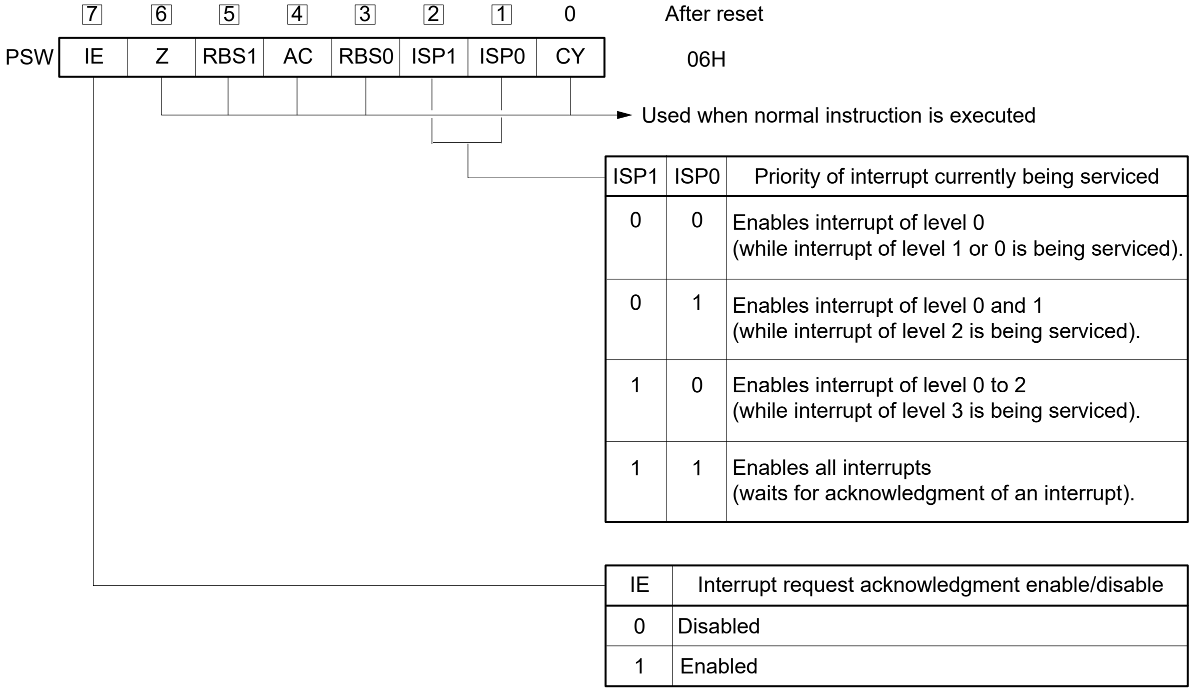
16.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 16 - 10 Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

16.4 Interrupt Servicing Operations

16.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 16 - 5 below.

For the interrupt request acknowledgment timing, see **Figures 16 - 12** and **16 - 13**.

Table 16 - 5 Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time Note
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

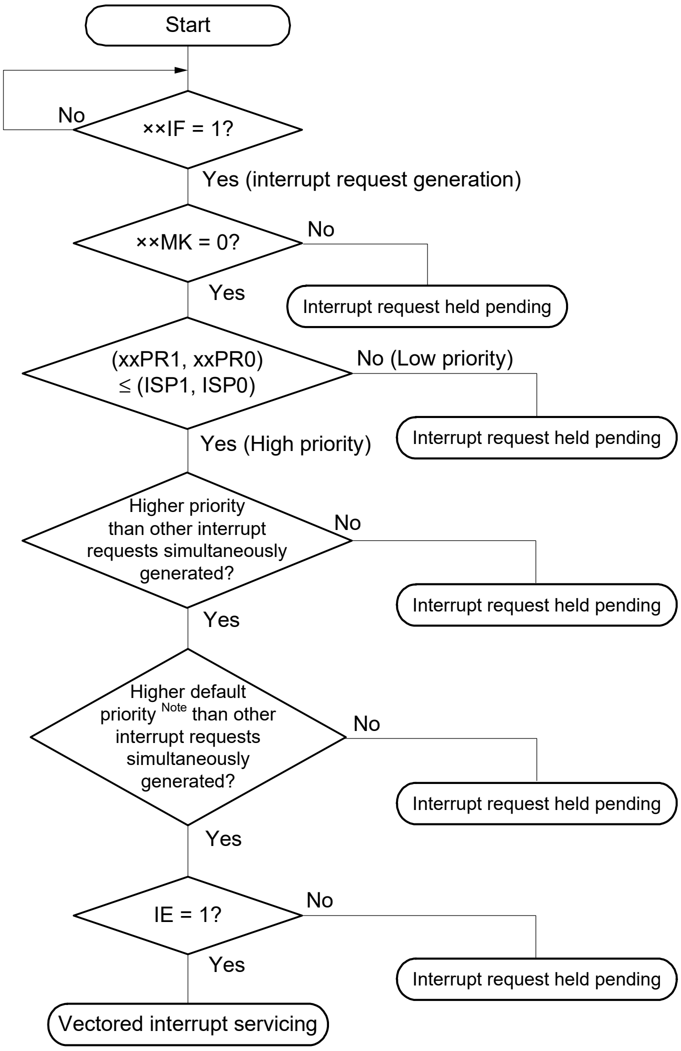
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16 - 11 shows the Interrupt Request Acknowledgment Processing Algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

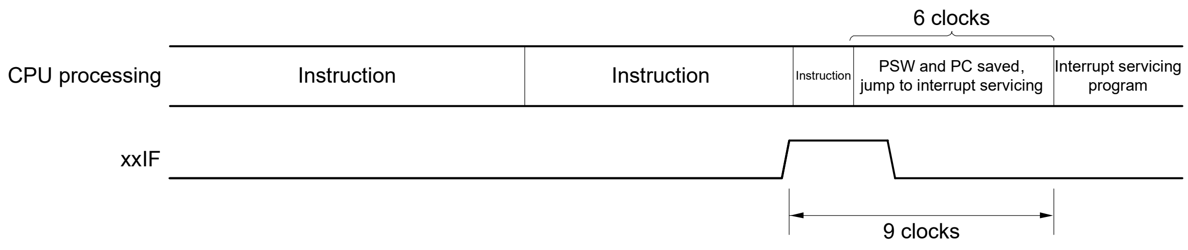
Figure 16 - 11 Interrupt Request Acknowledgment Processing Algorithm



xxIF:	Interrupt request flag
xxMK:	Interrupt mask flag
xxPR0:	Priority specification flag 0
xxPR1:	Priority specification flag 1
IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1:	Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 16 - 10**)

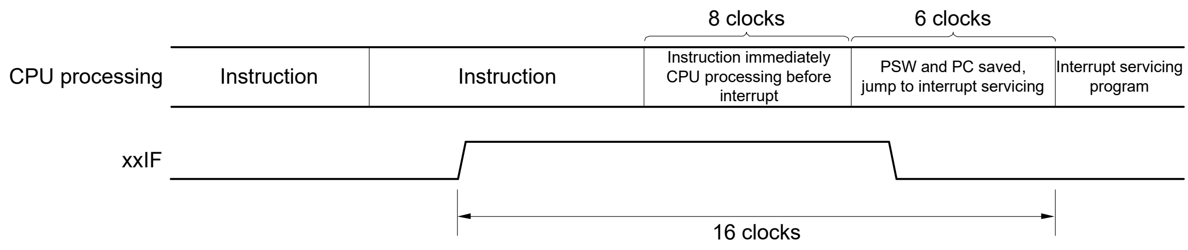
Note For the default priority, refer to **Tables 16 - 1** and **16 - 2 Interrupt Source List**.

Figure 16 - 12 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 16 - 13 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

16.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

16.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 16 - 6 shows Relationship between Interrupt Requests Enabled for Multiple Interrupt Servicing during Interrupt Servicing and Figures 16 - 14 and 16 - 15 show multiple interrupt servicing examples.

Table 16 - 6 Relationship between Interrupt Requests Enabled for Multiple Interrupt Servicing during Interrupt Servicing

Multiple Interrupt Request Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	√	×	×	×	×	×	×	×	√
	ISP1 = 0 ISP0 = 1	√	×	√	×	×	×	×	×	√
	ISP1 = 1 ISP0 = 0	√	×	√	×	√	×	×	×	√
	ISP1 = 1 ISP0 = 1	√	×	√	×	√	×	√	×	√
Software interrupt		√	×	√	×	√	×	√	×	√

Remark 1. √: Multiple interrupt servicing enabled

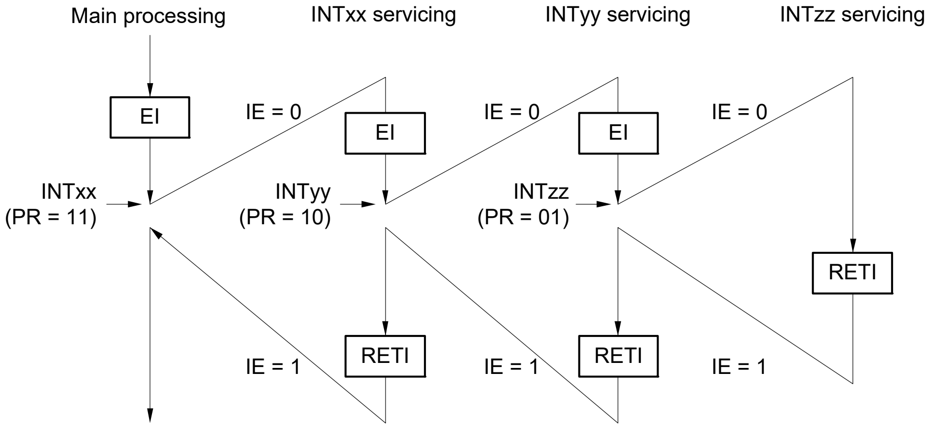
Remark 2. ×: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.
PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

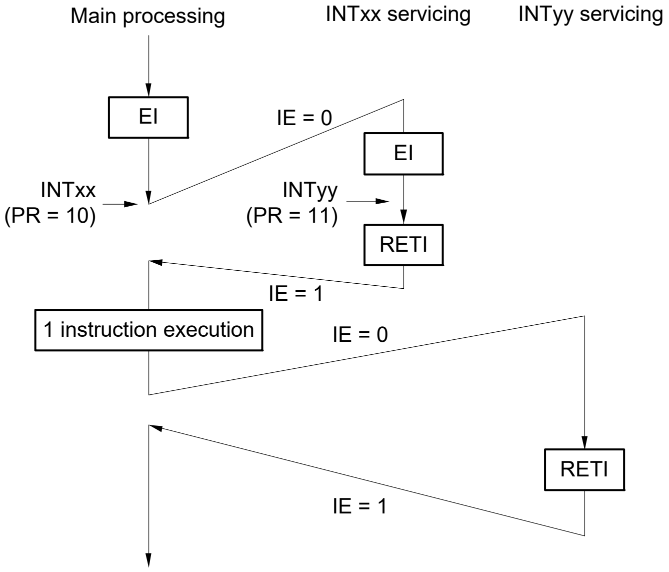
Figure 16 - 14 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

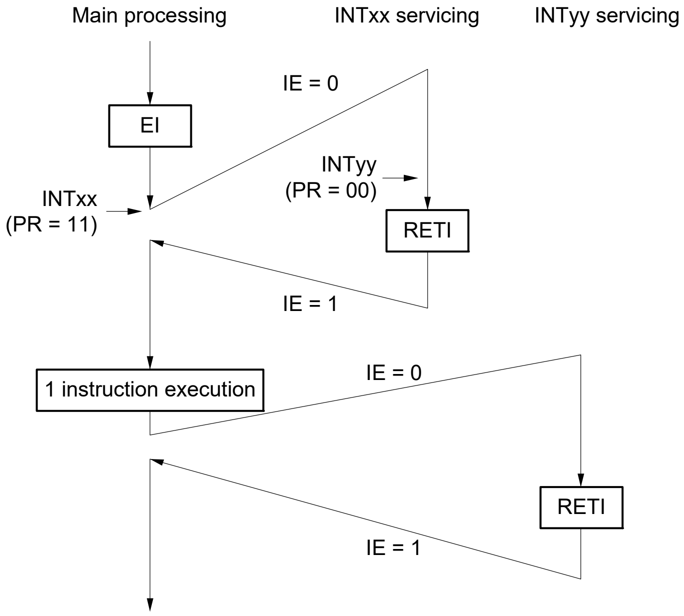


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 16 - 15 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

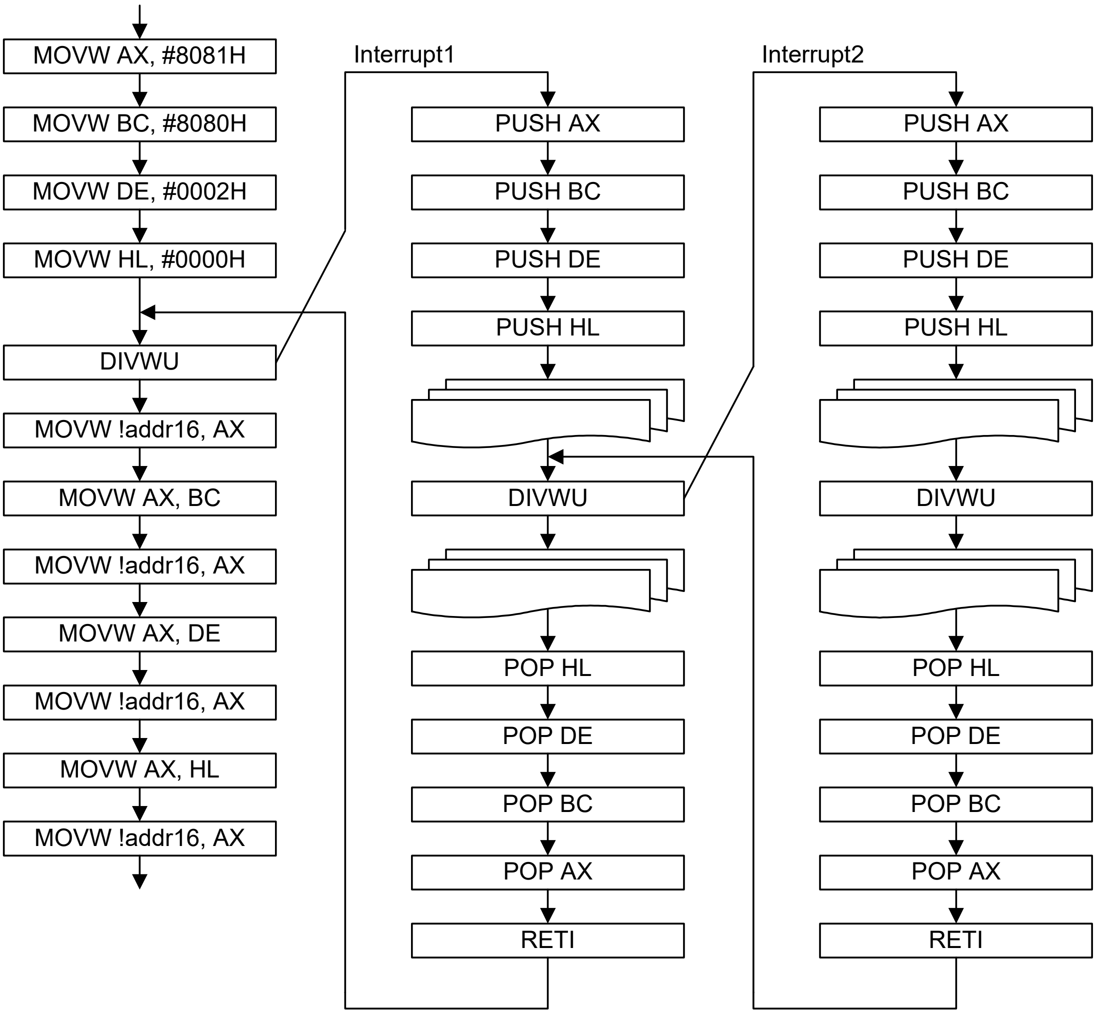
16.4.4 Interrupt servicing during division instruction

The RL78 microcontroller handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

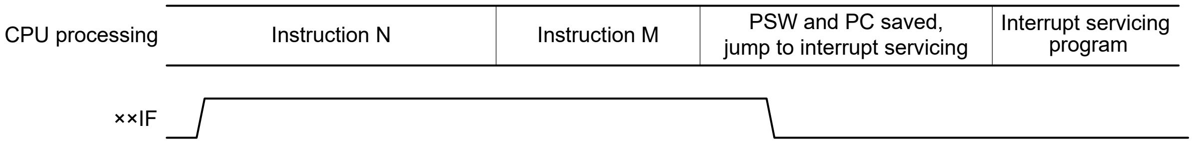
16.4.5 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 16 - 16 shows the timing at which interrupt requests are held pending.

Figure 16 - 16 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction

Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 17 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	30, 32-pin	44-pin
Key interrupt input channels	—	4 ch

17.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by inputting a falling edge to the key interrupt input pins (KR0 to KR3).

Table 17 - 1 Assignment of Key Interrupt Detection Pins

Key interrupt pins	Key return mode register (KRM)
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3

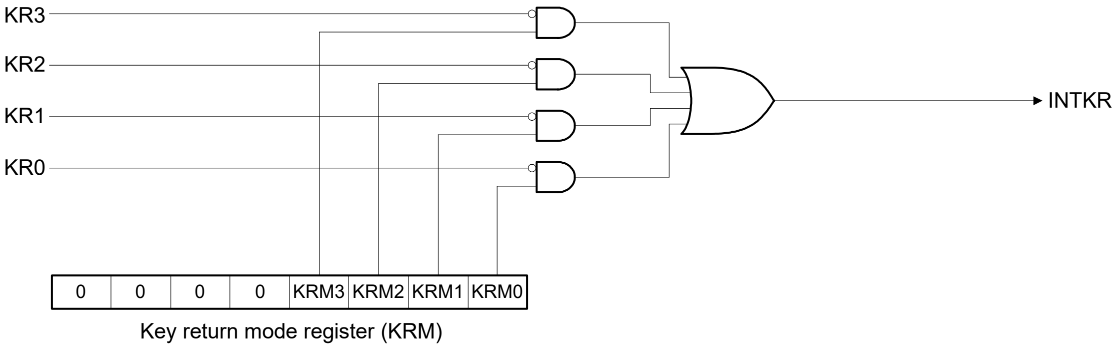
17.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 17 - 2 Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM) Port mode register 7 (PM7)

Figure 17 - 1 Block Diagram of Key Interrupt



17.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following registers.

- Key return mode register (KRM)
- Port mode register 7 (PM7)

17.3.1 Key return mode register (KRM)

The KRM0 to KRM3 bits are registers for controlling signals KR0 to KR3.

The KRM register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17 - 2 Format of Key return mode register (KRM)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	0	0	0	0	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. If any of the KRM0 to KRM3 bits used is set to 1, pull up the corresponding input pin to V_{DD} via an external resistor. Or, among bits 0 to 3 (PU70 to PU73) of pull-up resistor register 7 (PU7) for the corresponding input pin, set the target bit to 1 and use an internal pull-up resistor.

Caution 2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (t_{KR}) (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

Caution 3. The pins not used in the key interrupt mode can be used as normal ports.

17.3.2 Port mode register 7 (PM7)

When port 7 is used as the key interrupt input pins (KR0 to KR3), set the PM7n bit to 1. The output latches of P7n at this time may be 0 or 1. The PM7 register can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation sets this register to FFH.

Use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register 7 (PU7).

Figure 17 - 3 Format of Port mode register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	1	1	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator or high-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSIp or UARTq data reception, an A/D conversion request by the timer trigger signal (the interrupt request signal (INTIT) or ELC event input), the STOP mode is exited, the CSIp or UARTq data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution 1. The STOP mode can be used only when the CPU is operating on the main system clock.

Caution 2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

Caution 3. When using CSIp, UARTq, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 12.3 Registers Controlling A/D Converter and 14.3 Registers Controlling Serial Array Unit.

Caution 4. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

Caution 5. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.

Remark p = 00; q = 0; m = 0

18.2 Registers Controlling Standby Function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 12 A/D CONVERTER** and **CHAPTER 14 SERIAL ARRAY UNIT**.

18.2.1 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case.

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.
When reset is released (reset by $\overline{\text{RESET}}$ input, POR, LVD, WDT, and executing an illegal instruction), the STOP instruction and MSTOP bit (bit 7 of clock operation status control register (CSC)) = 1 clear this register to 00H.

Figure 18 - 1 Format of Oscillation stabilization time counter status register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.2 ms min.	13.1 ms min.

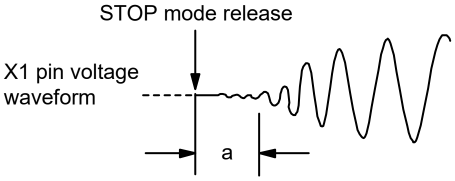
Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS). If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after STOP mode is released.

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

18.2.2 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using the OSTS register after the STOP mode is released.

When the high-speed on-chip oscillator clock is selected as the CPU clock, confirm with the oscillation stabilization time counter status register (OSTC) that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using the OSTC register.

The OSTS register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 07H.

Figure 18 - 2 Format of Oscillation stabilization time select register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

Caution 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

Caution 2. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

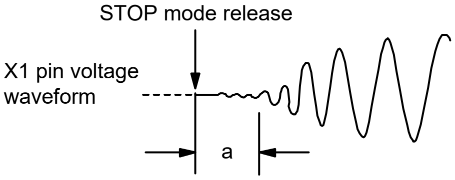
Caution 3. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

Caution 4. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register. If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after STOP mode is released.

Caution 5. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

18.3 Standby Function Operation

18.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 18 - 1 Operating Statuses in HALT Mode

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Main System Clock		
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
RAM			Operation stopped		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
12-bit Interval timer					
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.		
Timer RJ			Operable		
Timer RD					
Clock output/buzzer output					
A/D converter					
Comparator					
Programmable gain amplifier					
Serial array unit (SAU)					
Event link controller (ELC)			Operable function blocks can be linked		
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
Key interrupt					
CRC operation function	High-speed CRC				
	General-purpose CRC		Operation stopped		
Illegal-memory access detection function			Operation stopped		
RAM parity error detection function					
RAM guard function					
SFR guard function					

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.

Operation disabled: Operation is stopped before switching to the HALT mode.

f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock

f_X: X1 clock f_{EX}: External main system clock

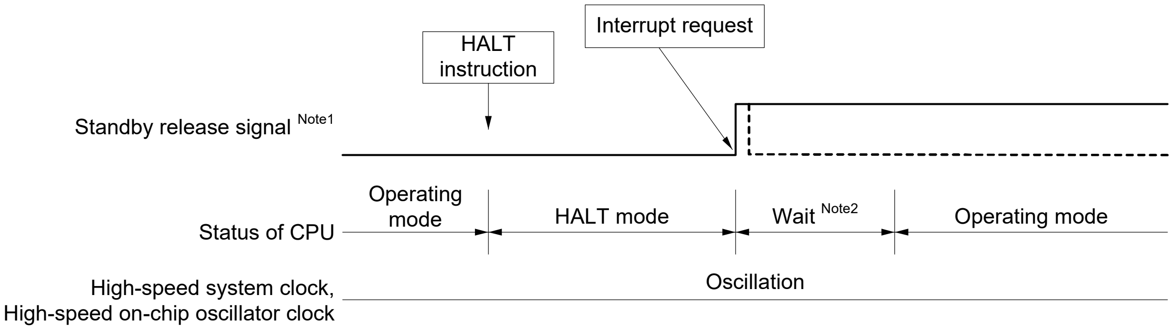
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 18 - 3 HALT Mode Release by Interrupt Request Generation



Note 1. Refer to **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out: 15 to 16 clock
- When vectored interrupt servicing is not carried out: 9 to 10 clock

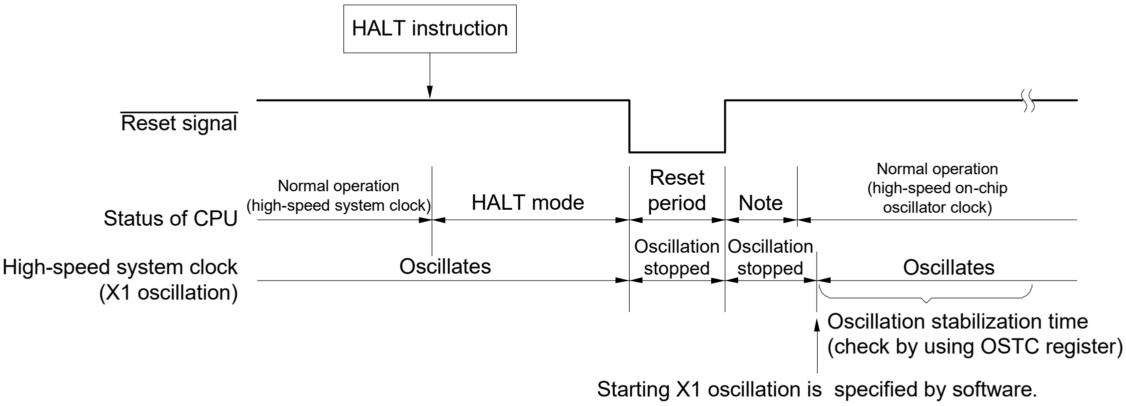
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

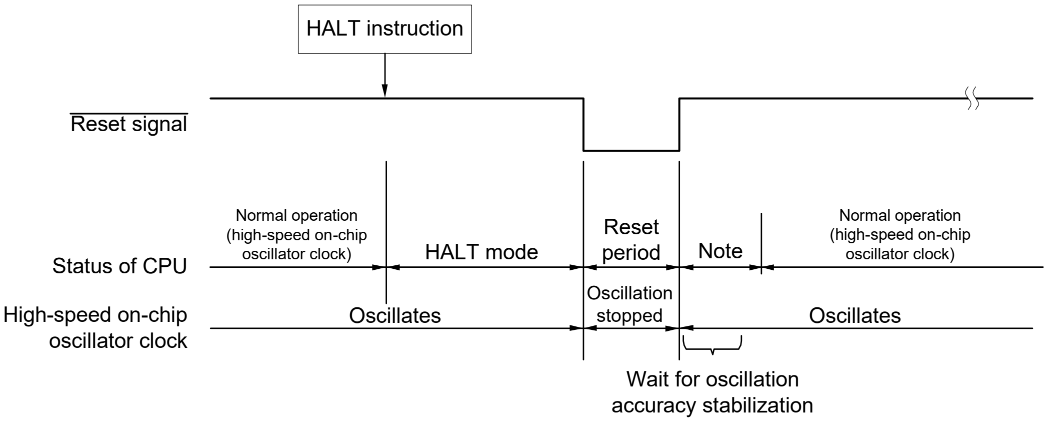
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18 - 4 HALT Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When high-speed on-chip oscillator clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 19 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 20 POWER-ON-RESET CIRCUIT**.

18.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution 1. Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation.

Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

Caution 2. When using CSIp, UARTq, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 12.3 Registers Controlling A/D Converter and 14.3 Registers Controlling Serial Array Unit.

Remark p = 00; q = 0; m = 0

The operating statuses in the STOP mode are shown below.

Table 18 - 2 Operating Statuses in STOP Mode

STOP Mode Setting Item			When STOP Instruction is Executed While CPU is Operating on Main System Clock		
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EX}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
RAM			Operation stopped		
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
12-bit Interval timer			Operable		
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.		
Timer RJ			• Timer RJ can operate in the event counting mode when the TRJIO input filter is not selected. • Timer RJ can operate when the low-speed on-chip oscillator is selected as the clock source to drive counting. • Timer RJ cannot operate with settings other than those above.		
Timer RD			Operation disabled		
Clock output/buzzer output			Operation disabled		
A/D converter			Wakeup operation is enabled (switching to the SNOOZE mode)		
Comparator			Operable when the digital filter is not used		
Programmable gain amplifier			Operable		
Serial array unit (SAU)			Wakeup operation is enabled only for CSIp and UARTq (switching to the SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq		
Event link controller (ELC)			Operable function blocks can be linked		
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
Key interrupt					
CRC operation function	High-speed CRC		Operation stopped		
	General-purpose CRC				
Illegal-memory access detection function					
RAM parity error detection function					
RAM guard function					
SFR guard function					

(**Remarks** are listed on the next page.)

Caution 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

Caution 2. To stop the low-speed on-chip oscillator clock in the STOP mode, must previously be set an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0).

Caution 3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator

clock before the execution of the STOP instruction. Before changing the CPU clock from the high-speed on-chip oscillator clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock	f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock	f_{EX}: External main system clock

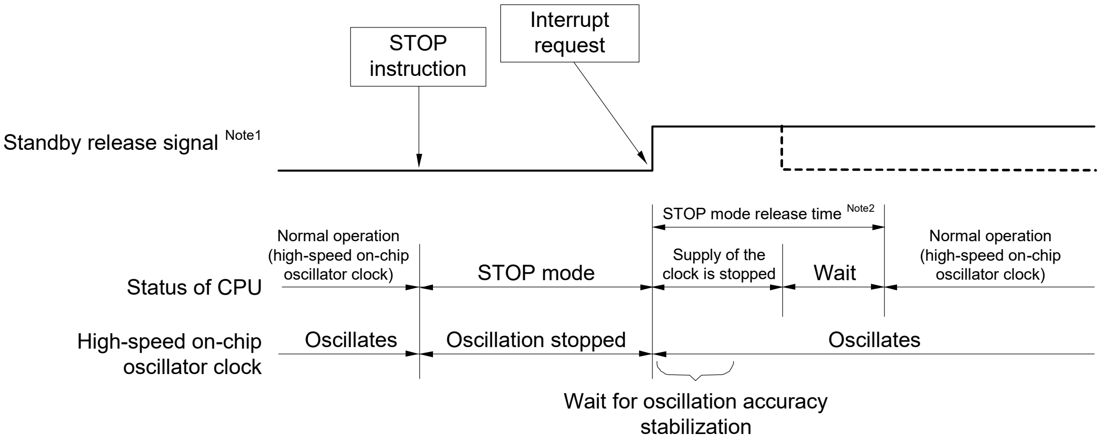
Remark 2. p = 00; q = 0

(2) STOP mode release
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 18 - 5 STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)** and **Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)**.

Note 2. STOP mode release time
Supply of the clock is stopped:
- When FRQSEL4 = 0: 18 μs to 65 μs
- When FRQSEL4 = 1: 18 μs to 135 μs

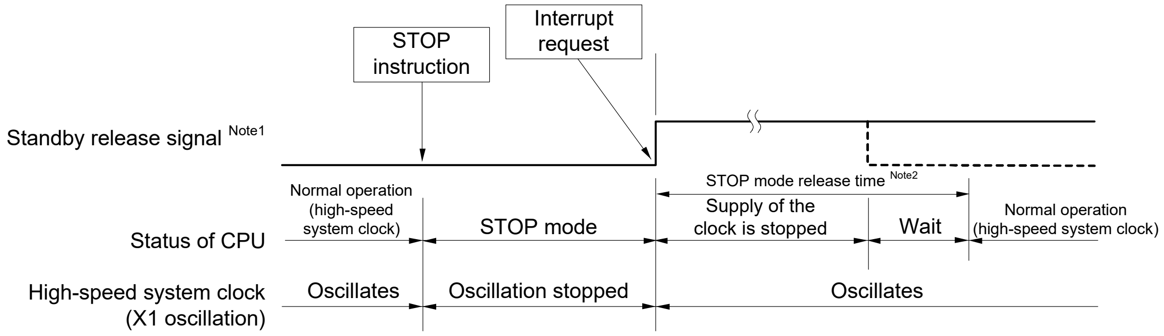
Wait
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 18 - 6 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Note 1. For details on the standby release signal, refer to **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)** and **Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)**.

Note 2. STOP mode release time

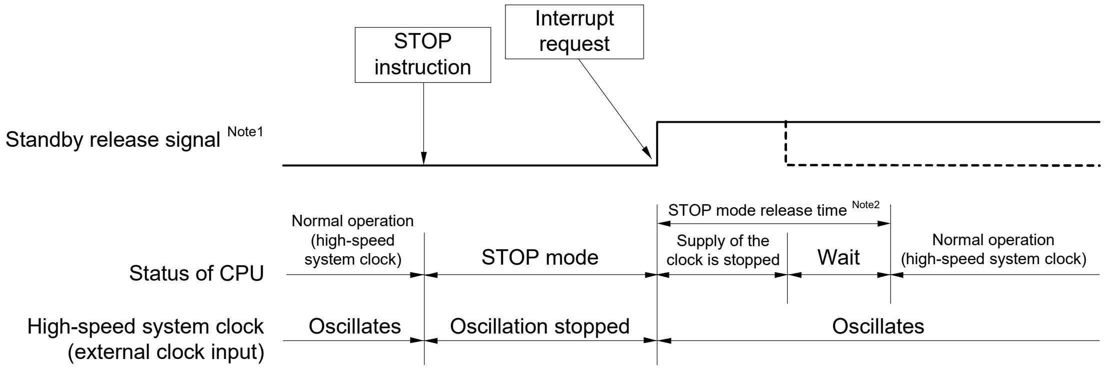
Supply of the clock is stopped:

- When FRQSEL4 = 0: 18 μs to “whichever is longer 65 μs or the oscillation stabilization time (set by OSTS)”
- When FRQSEL4 = 1: 18 μs to “whichever is longer 135 μs or the oscillation stabilization time (set by OSTS)”

Wait

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details on the standby release signal, refer to **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)** and **Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)**

Note 2. STOP mode release time

Supply of the clock is stopped:

- When FRQSEL4 = 0: 18 μs to 65 μs
- When FRQSEL4 = 1: 18 μs to 135 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution **To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.**

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

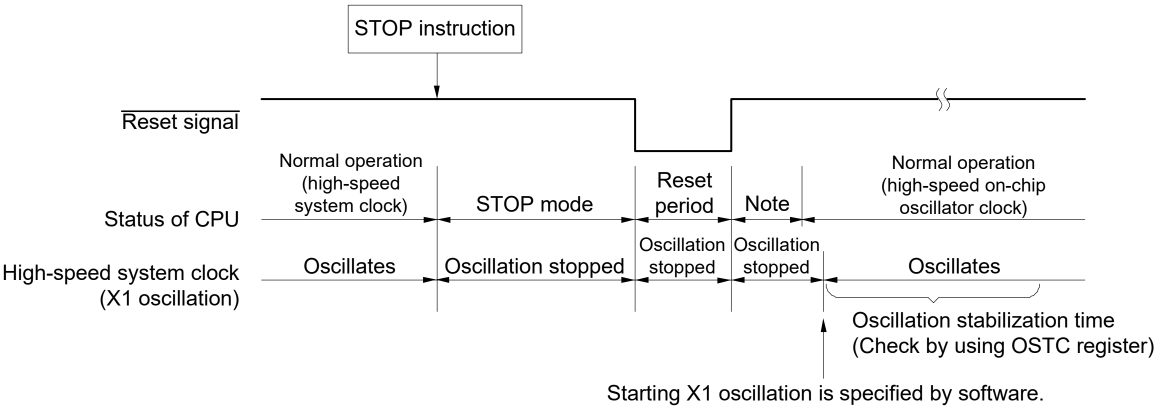
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

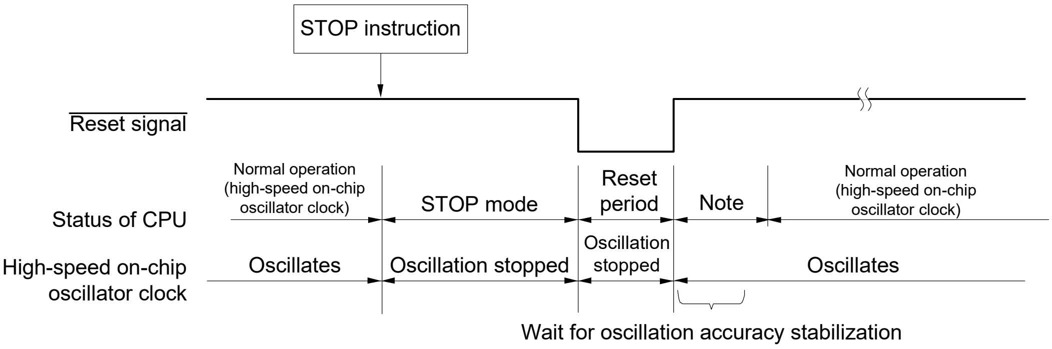
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18 - 7 STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When high-speed on-chip oscillator clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 19 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 20 POWER-ON-RESET CIRCUIT**.

18.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSIp, UARTq, or the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.
When using CSIp or UARTq in the SNOOZE mode, set the SWCm bit of the serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode. For details, **14.3 Registers Controlling Serial Array Unit**.
When using the A/D converter in the SNOOZE mode, set the AWC bit of the A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **12.4 A/D Converter Conversion Operations**.

Remark p = 00; q = 0; m = 0

In SNOOZE mode transition, wait status to be only following time.
Transition time from STOP mode to SNOOZE mode:
When FRQSEL4 = 0: 18 μs to 65 μs
When FRQSEL4 = 1: 18 μs to 135 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:
- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 7 clocks
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 7 clocks

- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 1 clock
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

Table 18 - 3 Operating Statuses in SNOOZE Mode

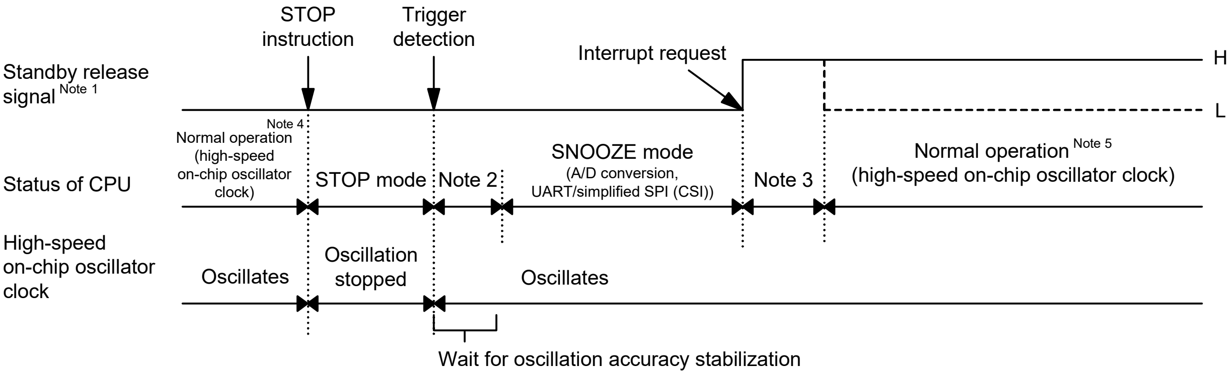
STOP Mode Setting Item			During STOP mode, receiving data signal from CSIp and UARTq, and inputting timer trigger signal to A/D converter by interrupt
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Operation started
		f_X	Stopped
		f_{EX}	
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops
CPU			Operation stopped
Code flash memory			
RAM			Operation stopped
Port (latch)			Use of the status while in the STOP mode continues
Timer array unit			Operation disabled
12-bit Interval timer			Operable
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.
Timer RJ			Operation disabled
Timer RD			
Clock output/buzzer output			Operation disabled
A/D converter			Operable
Comparator			Operable when the digital filter is not used
Programmable gain amplifier			Operable
Serial array unit (SAU)			Operable only CSIp and UARTq only. Operation disabled other than CSIp and UARTq.
Event link controller (ELC)			Operable function blocks can be linked
Power-on-reset function			Operable
Voltage detection function			
External interrupt			
Key interrupt			
CRC operation function	High-speed CRC		Operation stopped
	General-purpose CRC		
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock

Remark 2. p = 00; q = 0

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 18 - 8 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)** and **Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)**.

Note 2. Transition time from STOP mode to SNOOZE mode

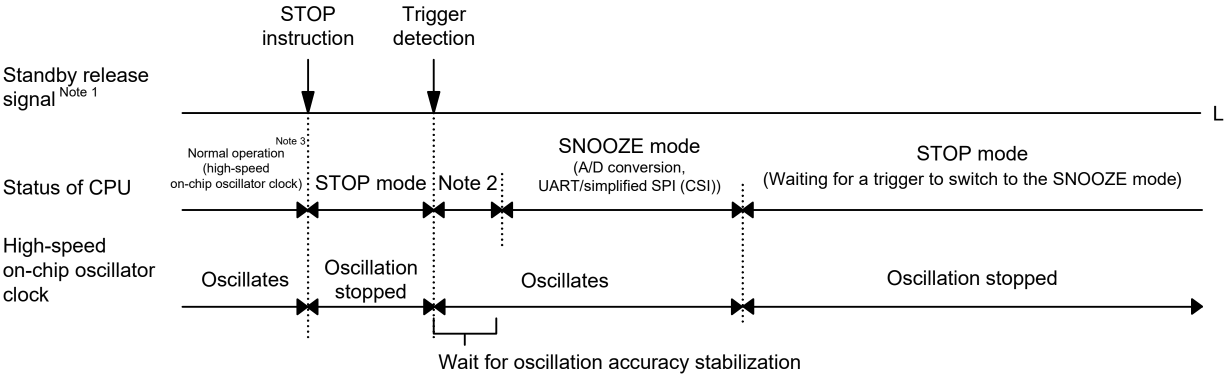
Note 3. Transition time from SNOOZE mode to normal operation

Note 4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 18 - 9 When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 16 - 1 Basic Configuration of Interrupt Function (1/2)** and **Figure 16 - 2 Basic Configuration of Interrupt Function (2/2)**.

Note 2. Transition time from STOP mode to SNOOZE mode

Note 3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

Remark For details of the SNOOZE mode function, see **CHAPTER 12 A/D CONVERTER** and **CHAPTER 14 SERIAL ARRAY UNIT**.

CHAPTER 19 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction $^{\text{Note}}$
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.
A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction $^{\text{Note}}$, RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 19 - 1.
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the high-speed on-chip oscillator clock after reset processing. A reset by the watchdog timer overflow, execution of illegal instruction, detection of RAM parity error, or detection of illegal memory access is automatically released, and program execution starts using the high-speed on-chip oscillator clock (see **Figures 19 - 2** to **19 - 4**) after reset processing. Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset, and program execution starts using the high-speed on-chip oscillator clock (see **CHAPTER 20 POWER-ON-RESET CIRCUIT** and **CHAPTER 21 VOLTAGE DETECTOR**) after reset processing.

Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range.
The operating voltage range depends on the setting of the user option byte (000C2H).
The following shows the operating voltage range.
HS (high-speed main) mode: V_{DD} = 2.7 to 5.5 V@1 MHz to 24 MHz
LS (low-speed main) mode: V_{DD} = 2.7 to 5.5 V@1 MHz to 8 MHz

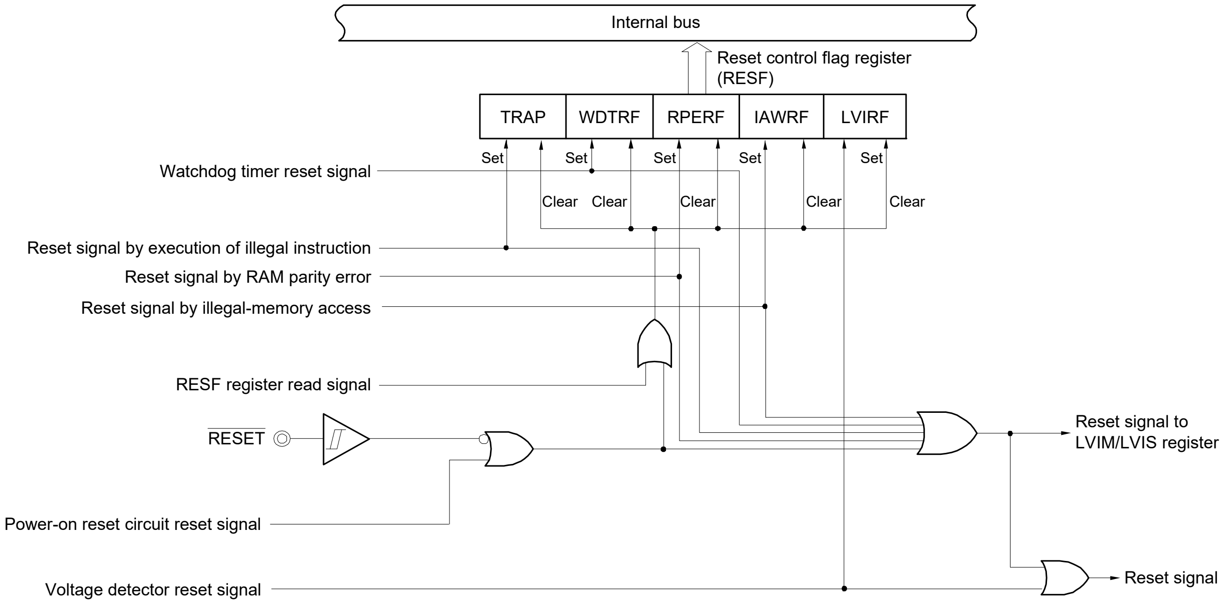
Caution 2. During reset input, the X1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock oscillating. External main system clock input become invalid.

Caution 3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.

- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).**
- **Ports other than P40: High-impedance during the reset period or after receiving a reset signal.**

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Figure 19 - 1 Block Diagram of Reset Function



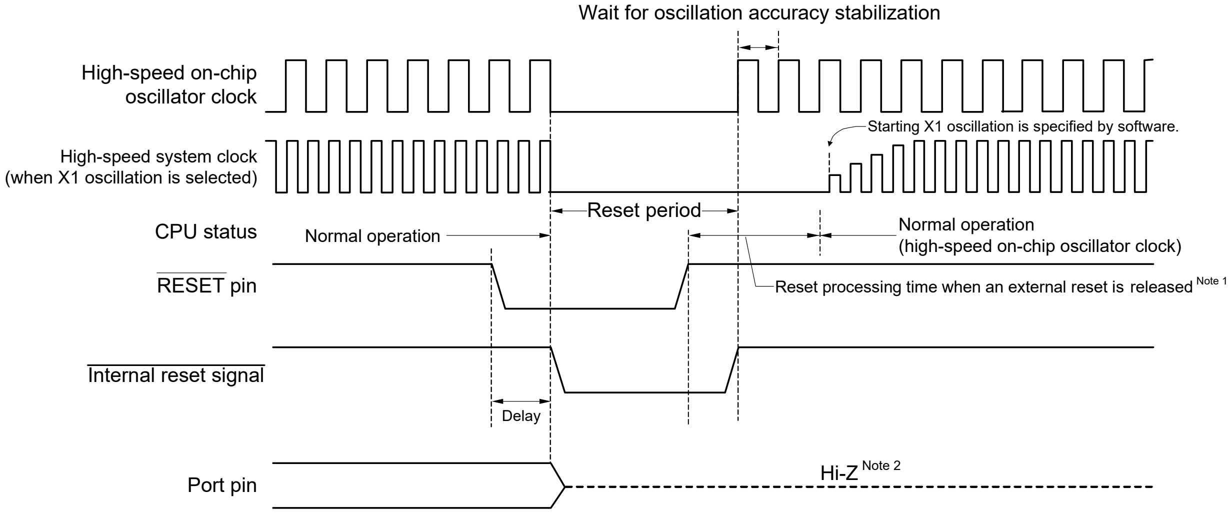
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remark 1. LVIM: Voltage detection register
Remark 2. LVIS: Voltage detection level register

19.1 Timing of Reset Operation

This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

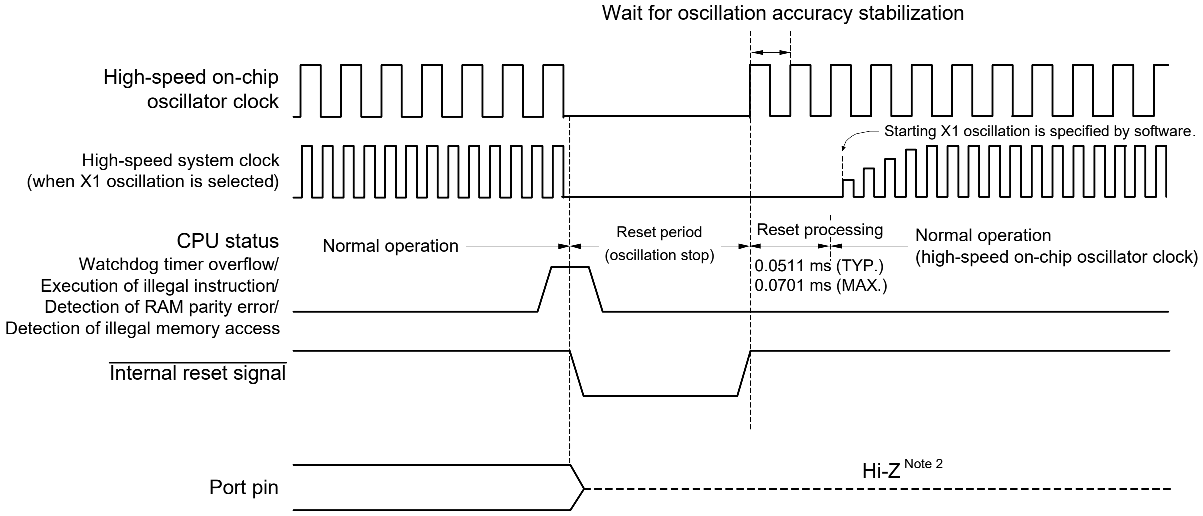
Figure 19 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



(Notes and Caution are listed on the next page.)

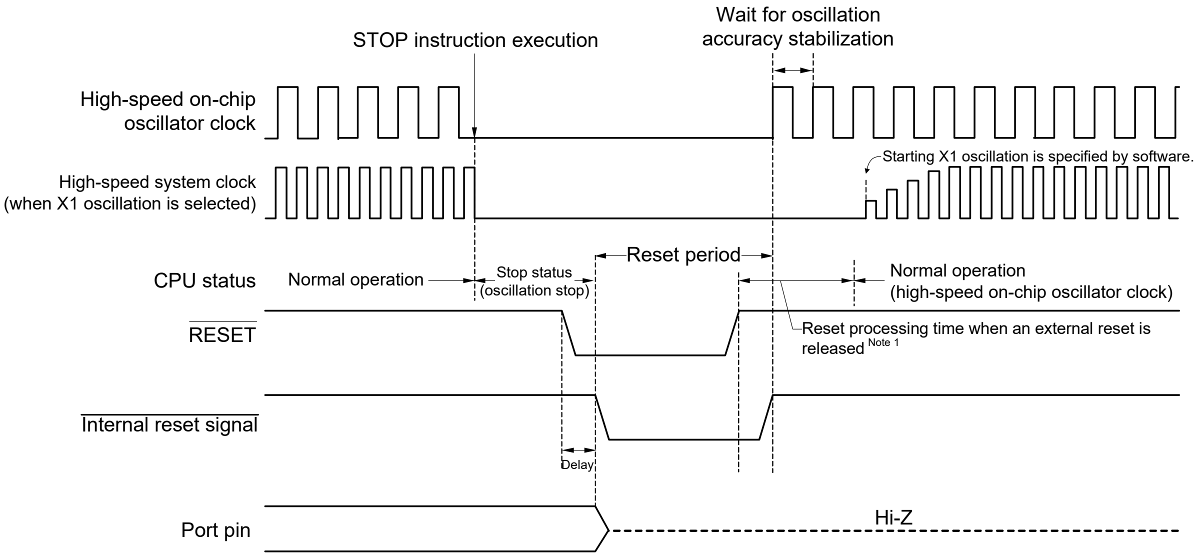
Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, program execution starts with the high-speed on-chip oscillator clock as the operating clock.

Figure 19 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



(Notes and Caution are listed on the next page.)

Figure 19 - 4 Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note 1. Reset processing time when an external reset is released:

First time after POR release: 0.672 ms (TYP.), 0.832 ms (MAX.) (when LVD is used)
0.399 ms (TYP.), 0.519 ms (MAX.) (when LVD is off)

Second and subsequent times after POR release: 0.531 ms (TYP.), 0.675 ms (MAX.) (when LVD is used)
0.259 ms (TYP.), 0.362 ms (MAX.) (when LVD is off)

When the supply voltage rises, the voltage stabilization wait time 0.99 ms (TYP.) to 2.30 ms (MAX.) is required before the reset processing time when an external reset is released.

Note 2. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).

Remark For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 20 POWER-ON-RESET CIRCUIT** and **CHAPTER 21 VOLTAGE DETECTOR**.

Table 19 - 1 Operation Statuses during Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Code flash memory			Operation stopped
RAM			Operation stopped
Port (latch)		P40	High impedance after a pin reset or POR. Pulled up after a reset other than a pin reset or POR.
		Other than P40	High-impedance
Timer array unit			Operation stopped
Timer RJ			
Timer RD			
12-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Comparator			
Programmable gain amplifier			
Serial array unit (SAU)			
Event link controller (ELC)			
PWM option unit			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
Key interrupt function			
CRC operation function		High-speed CRC	
		General-purpose CRC	
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark f_{IH}: High-speed on-chip oscillator clock f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock

Table 19 - 2 Hardware Statuses After Reset Acknowledgment (1/4)

Hardware		After Reset Acknowledgment Note 1
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
Multiplier and divider/multiply-accumulator	Multiply and accumulation register (L) (MACRL)	0000H
	Multiply and accumulation register (H) (MACRH)	0000H
RAM	Data memory	Undefined
	General-purpose registers	Undefined
Processor mode control register (PMC)		00H
Port registers (P0 to P7, P14) (output latches)		00H
Port registers (P12, P13) (output latches)		Undefined
Port mode registers (PM0 to PM7, PM12, PM14)		FFH
Port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14)		FFH
Port input mode registers 0, 1, 3, 5 (PIM0, PIM1, PIM3, PIM5)		00H
Port output mode registers 0, 1, 3, 5 (POM0, POM1, POM3, POM5)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU12, PU14)		00H (PU4 is 01H)
Peripheral I/O redirection register 1 (PIOR1)		00H
Port mode select register (PMS)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
System clock control register (CKC)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable registers 0, 1 (PER0, PER1)		00H
High-speed on-chip oscillator frequency select register (HOCODIV)		Undefined
High-speed on-chip oscillator trimming register (HIOTRM)		Note 2
Operation speed mode control register (OSMC)		00H
Timer array unit	Timer data registers 00 to 03 (TDR00 to TDR03)	0000H
	Timer mode registers 00 to 03 (TMR00 to TMR03)	0000H
	Timer status registers 00 to 03 (TSR00 to TSR03)	0000H
	Timer input select register 0 (TIS0)	00H
	Timer counter registers 00 to 03 (TCR00 to TCR03)	FFFFH
	Timer channel enable status register 0 (TE0)	0000H
	Timer channel start register 0 (TS0)	0000H
	Timer channel stop register 0 (TT0)	0000H
	Timer clock select register 0 (TPS0)	0000H
	Timer output register 0 (TO0)	0000H
	Timer output enable register 0 (TOE0)	0000H
	Timer output level register 0 (TOL0)	0000H
	Timer output mode registers 0 (TOM0)	0000H

Note 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Note 2. The reset value differs for each chip.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

Table 19 - 3 Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment Note 1
Timer RJ	Timer RJ Counter Register 0 (TRJ0)	FFFFH
	Timer RJ Control Register 0 (TRJCR0)	00H
	Timer RJ I/O Control Register 0 (TRJIOC0)	00H
	Timer RJ Mode Register 0 (TRJMR0)	00H
	Timer RJ Event Pin Select Register 0 (TRJISR0)	00H
Timer RD	Timer RD ELC Register (TRDELC)	00H Note 2
	Timer RD Start Register (TRDSTR)	0CH Note 2
	Timer RD Mode Register (TRDMR)	00H Note 2
	Timer RD PWM Function Select Register (TRDPMR)	00H Note 2
	Timer RD Function Control Register (TRDFCR)	80H Note 2
	Timer RD Output Master Enable Register 1 (TRDOER1)	FFH Note 2
	Timer RD Output Master Enable Register 2 (TRDOER2)	00H Note 2
	Timer RD Output Control Register (TRDOCR)	00H Note 2
	Timer RD Digital Filter Function Select Registers 0, 1 (TRDDF0, TRDDF1)	00H Note 2
	Timer RD Control Registers 0, 1 (TRDCR0, TRDCR1)	00H Note 2
	Timer RD I/O Control Registers A0, A1 (TRDIORA0, TRDIORA1)	00H Note 2
	Timer RD I/O Control Registers C0, C1 (TRDIORC0, TRDIORC1)	88H Note 2
	Timer RD Status Registers 0, 1 (TRDSR0, TRDSR1)	00H Note 2
	Timer RD Interrupt Enable Registers 0, 1 (TRDIER0, TRDIER1)	00H Note 2
	Timer RD PWM Function Output Level Control Registers 0, 1 (TRDPOCR0, TRDPOCR1)	00H Note 2
	Timer RD Counters 0, 1 (TRD0, TRD1)	0000H Note 2
	Timer RD General Registers A0, A1, B0, B1, C0, C1, D0, D1 (TRDGRA0, TRDGRA1, TRDGRB0, TRDGRB1, TRDGRC0, TRDGRC1, TRDGRD0, TRDGRD1)	FFFFH Note 2
12-bit Interval timer	Control register (ITMC)	0FFFH
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH Note 3

Note 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Note 2. The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 3. The reset value of WDTE depends on the option byte setting.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

Table 19 - 4 Hardware Statuses After Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment Note
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode registers 0 to 2 (ADM0 to ADM2)	00H
	Conversion result comparison upper limit setting register (ADUL)	FFH
	Conversion result comparison lower limit setting register (ADLL)	00H
	A/D test register (ADTES)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H
Comparator/programmable gain amplifier	Comparator mode setting register (COMPMDR)	00H
	Comparator filter control register (COMPFIR)	00H
	Comparator output control register (COMPOCR)	00H
	Comparator internal reference voltage control register (CVRCTL)	00H
	Comparator internal reference voltage select register 0 (C0RVM)	00H
	Comparator internal reference voltage select register 1 (C1RVM)	00H
	PGA control register (PGACTL)	00H
PWM option unit	6-phase PWM option mode register (OPMR)	00H
	6-phase PWM option status register (OPSR)	00H
	6-phase PWM option Hi-Z start trigger register (OPHS)	00H
	6-phase PWM option Hi-Z stop trigger register (OPHT)	00H
Serial array unit (SAU)	Serial data registers 00 to 03 (SDR00 to SDR03)	0000H
	Serial status registers 00 to 03 (SSR00 to SSR03)	0000H
	Serial flag clear trigger registers 00 to 03 (SIR00 to SIR03)	0000H
	Serial mode registers 00 to 03 (SMR00 to SMR03)	0020H
	Serial communication operation setting registers 00 to 03 (SCR00 to SCR03)	0087H
	Serial channel enable status registers 0 (SE0)	0000H
	Serial channel start registers 0 (SS0)	0000H
	Serial channel stop registers 0 (ST0)	0000H
	Serial clock select registers 0 (SPS0)	0000H
	Serial output registers 0 (SO0)	0F0FH
	Serial output enable registers 0 (SOE0)	0000H
	Serial output level registers 0 (SOL0)	0000H
	Serial standby control register 0 (SSC0)	0000H
	Input switch control register (ISC)	00H
ELC	Event output destination select registers 00 to 18 (ELSELR00 to ELSELR18)	00H
Key interrupt	Key return mode register (KRM)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

Table 19 - 5 Hardware Statuses After Reset Acknowledgment (4/4)

Hardware		Status After Reset Acknowledgment Note 1
Reset function	Reset control flag register (RESF)	Undefined Note 2
Voltage detector	Voltage detection register (LVIM)	00H Note 2
	Voltage detection level register (LVIS)	00H/01H/81H Notes 2, 3
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	Mask flag registers 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR10L, PR10H, PR11L, PR11H, PR02L, PR12L, PR02H, PR12H)	FFH
	External interrupt rising edge enable registers 0 (EGP0)	00H
	External interrupt falling edge enable registers 0 (EGN0)	00H
Safety functions	Flash memory CRC control register (CRC0CTL)	00H
	Flash memory CRC operation result register (PGCRCL)	0000H
	CRC input register (CRCIN)	00H
	CRC data register (CRCD)	0000H
	Invalid memory access detection control register (IAWCTL)	00H
	RAM parity error control register (RPECTL)	00H
BCD correction circuit	BCD correction result register (BCDADJ)	Undefined

Note 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Note 2. These values vary depending on the reset source.

<table>
<tr><th colspan="2">Reset Source
Register</th><th>$\overline{\text{RESET}}$ Input</th><th>Reset by POR</th><th>Reset by Execution of Illegal Instruction</th><th>Reset by WDT</th><th>Reset by RAM parity error</th><th>Reset by illegal-memory access</th><th>Reset by LVD</th></tr>
<tr><td rowspan="5">RESF</td><td>TRAP bit</td><td colspan="2" rowspan="5">Cleared (0)</td><td>Set (1)</td><td colspan="4">Held</td></tr>
<tr><td>WDTRF bit</td><td>Held</td><td>Set (1)</td><td colspan="3">Held</td></tr>
<tr><td>RPERF bit</td><td colspan="2">Held</td><td>Set (1)</td><td colspan="2">Held</td></tr>
<tr><td>IAWRF bit</td><td colspan="3">Held</td><td>Set (1)</td><td>Held</td></tr>
<tr><td>LVIRF bit</td><td colspan="4">Held</td><td>Set (1)</td></tr>
<tr><td rowspan="3">LVIM</td><td>LVISEN bit</td><td colspan="6">Cleared (0)</td><td>Held</td></tr>
<tr><td>LVIOMSK bit</td><td colspan="6" rowspan="2">Cleared (0)</td><td rowspan="2">Held</td></tr>
<tr><td>LVIF bit</td></tr>
<tr><td colspan="2">LVIS</td><td colspan="6">Cleared (00H/01H/81H)</td><td>Held</td></tr>
</table>

Note 3. The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

19.2 Register for Confirming Reset Source

Many internal reset generation sources exist in the RL78 microcontroller The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 19 - 5 Format of Reset control flag register (RESF)

Address: FFFA8H After reset: Undefined Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the source. See **Table 19 - 6**.

Note 2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 22.5 RAM Parity Error Detection Function.

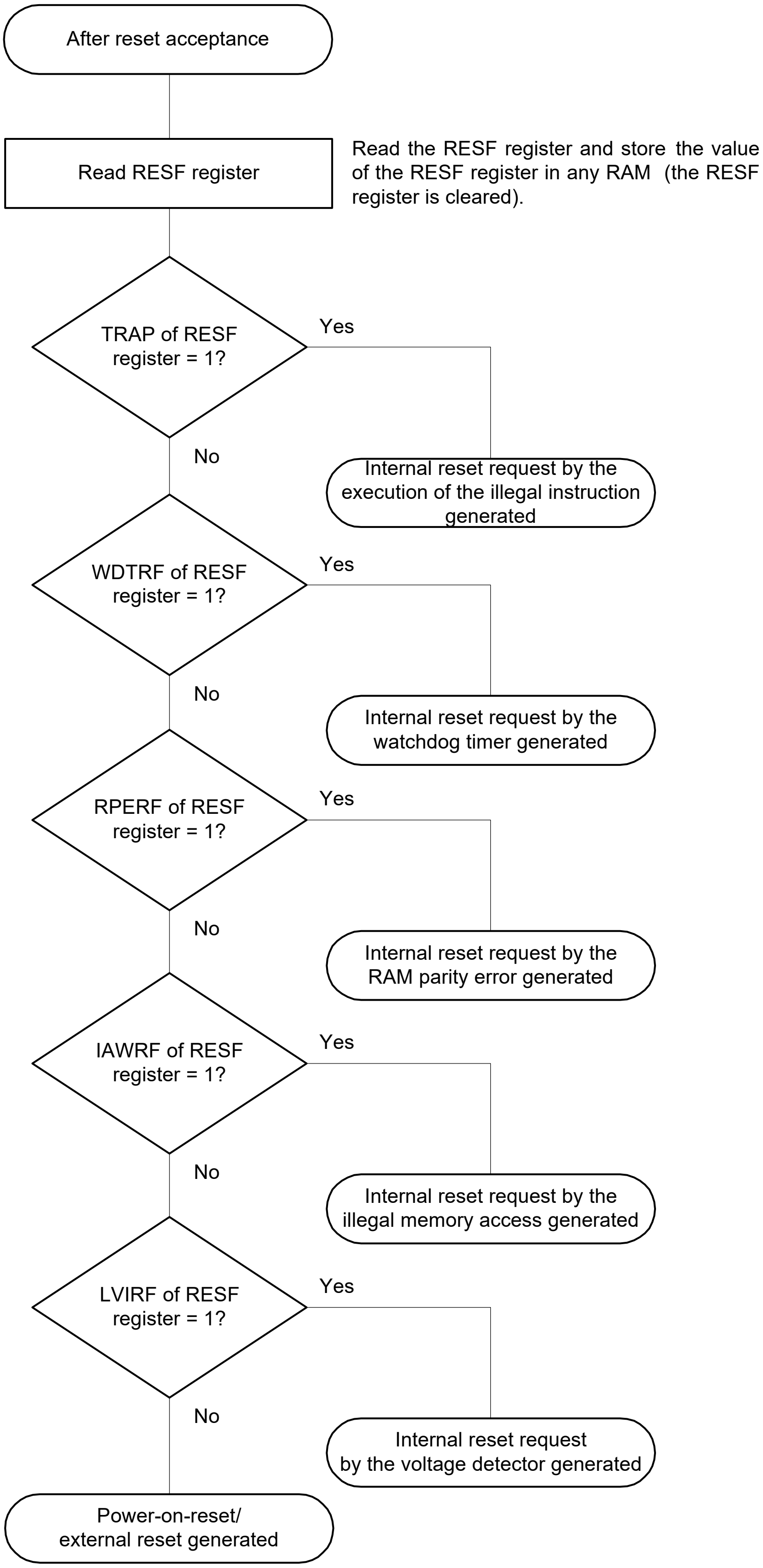
The status of the RESF register when a reset request is generated is shown in Table 19 - 6.

Table 19 - 6 RESF Register Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
TRAP bit	Cleared (0)		Set (1)	Held			
WDTRF bit			Held	Set (1)	Held		
RPERF bit			Held		Set (1)	Held	
IAWRF bit			Held			Set (1)	Held
LVIRF bit			Held				Set (1)

Figure 19 - 6 shows an example of procedure for checking reset source.

Figure 19 - 6 Example of Procedure for Checking Reset Source



CHAPTER 20 POWER-ON-RESET CIRCUIT

20.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **29.5 AC Characteristics** . This is done by utilizing the voltage detection circuit or controlling the externally input reset signal.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$.
  Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in **29.5 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) is cleared.

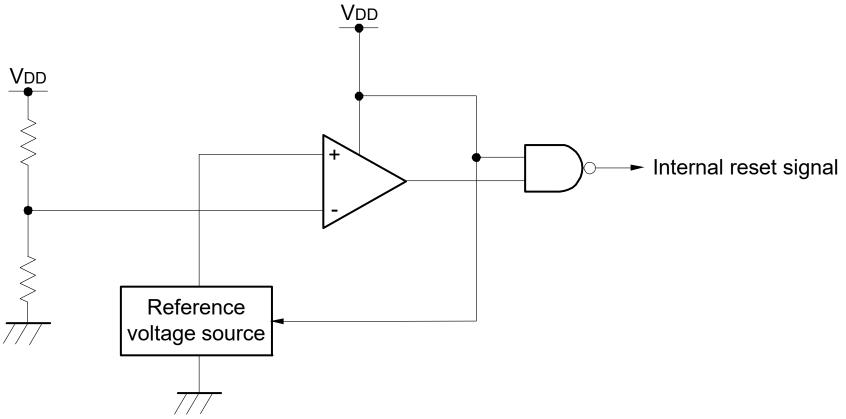
Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 19 RESET FUNCTION**.

Remark 2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **29.7.5 POR circuit characteristics**.

20.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 20 - 1.

Figure 20 - 1 Block Diagram of Power-on-reset Circuit

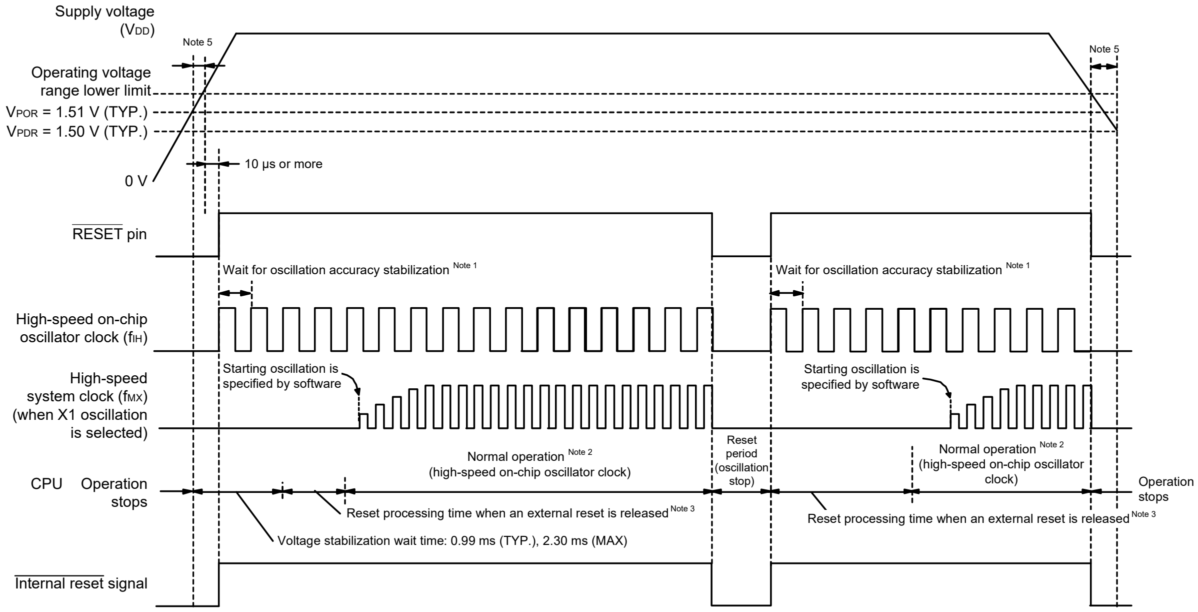


20.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 20 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When using an external reset by the $\overline{\text{RESET}}$ pin



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following reset processing time when the external reset is released (in the first reset processing following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, typ.) is reached.
The reset processing time when the external reset is released is shown below.
In the first reset processing following release from the POR state:
0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)
0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

Note 4. The reset processing time when the external reset is released in the second reset processing following release from the POR state is shown below.
In the second reset processing following release from the POR state:
0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)
0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

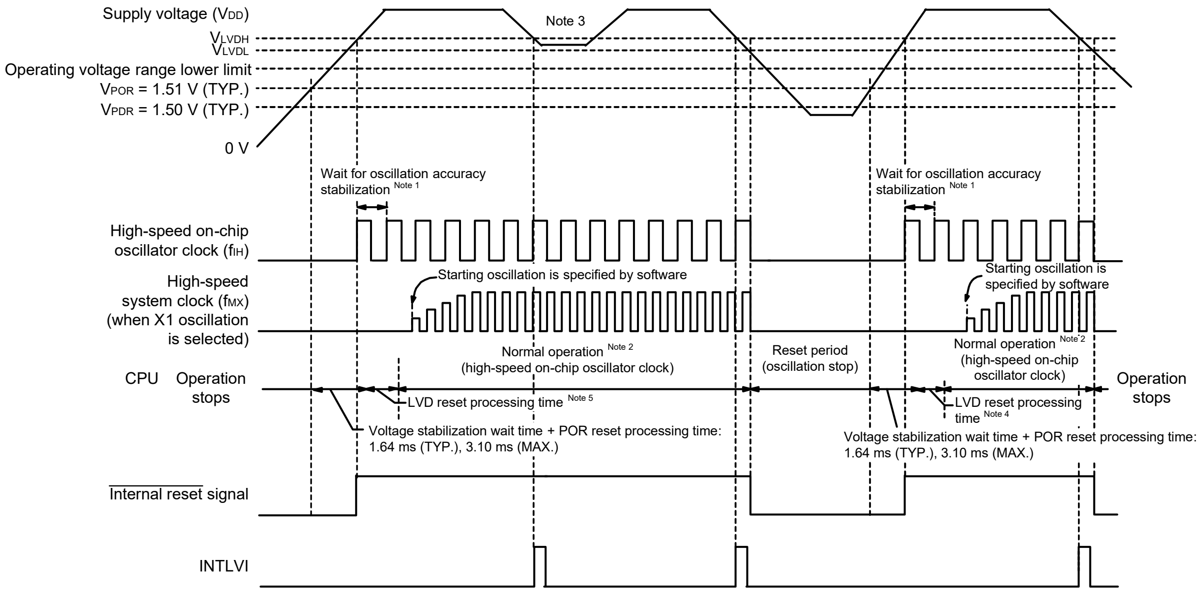
Note 5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **29.5 AC Characteristics**. This is done by controlling the externally input reset signal.
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution **For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 21 VOLTAGE DETECTOR.**

Figure 20 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD is interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 21 - 8 Processing Procedure After an Interrupt Is Generated** and **Figure 21 - 9 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

Note 4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Remark

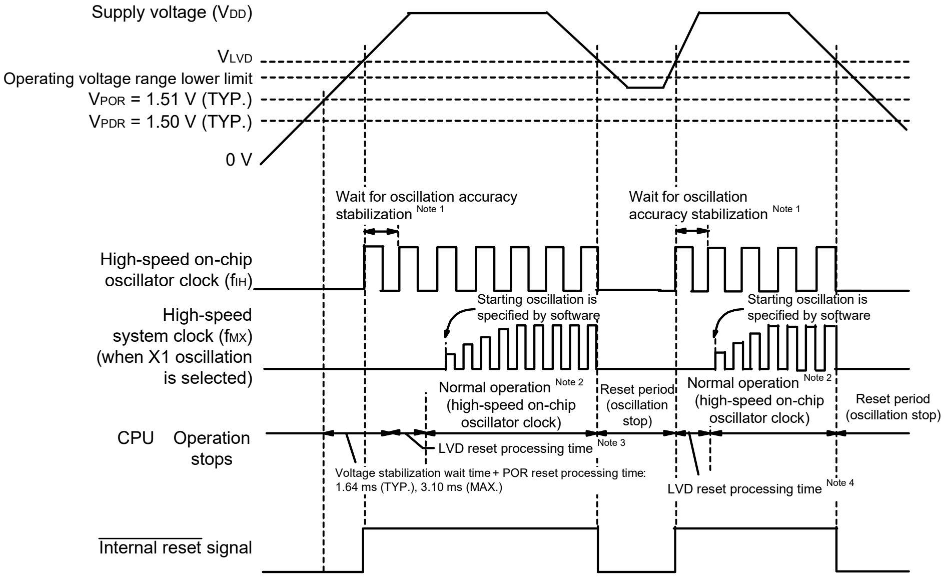
V_{LVDH}, V_{LVDL}: LVD detection voltage

V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 20 - 4 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.

LVD reset processing time: 0.0511 ms (TYP.), 0.0701 ms (MAX.)

Remark 1. V_{LVDH}, V_{LVDL}: LVD detection voltage

V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 20 - 4 (3).

CHAPTER 21 VOLTAGE DETECTOR

21.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 6 levels (For details, see **CHAPTER 24 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **29.5 AC Characteristics**. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts.
The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating/releasing resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for releasing resets/generating interrupts.

The reset and internal interrupt signals are generated in each mode as follows.

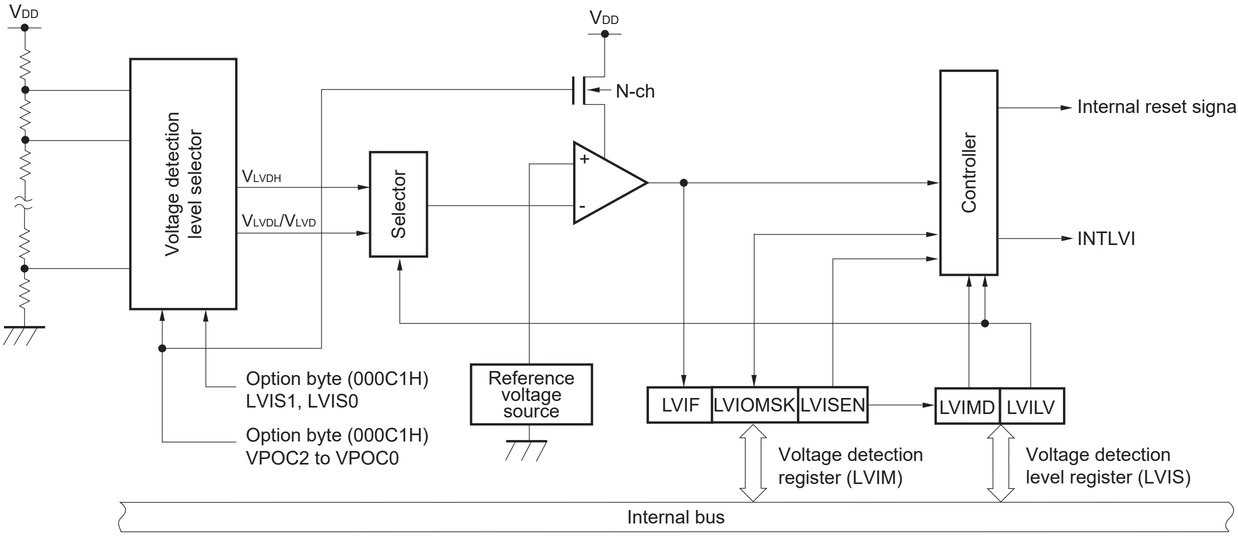
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 19 RESET FUNCTION**.

21.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 21 - 1.

Figure 21 - 1 Block Diagram of Voltage Detector



21.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

21.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21 - 2 Format of Voltage detection register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register Note 3 (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Note 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

Note 2. Bits 0 and 1 are read-only.

Note 3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.

Note 4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

21.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation input sets this register to 00H/01H/81H Note 1.

Figure 21 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset:00H/01H/81H Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Note 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

Note 2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Caution 1. Rewrite the value of the LVIS register according to Figures 21 - 8 and 21 - 9.

Caution 2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 21 - 4 shows the format of the user option byte (000C1H). For details about the option byte, see CHAPTER 24 OPTION BYTE.

Figure 21 - 4 Format of User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.92 V	2.86 V	2.75 V	0	1	1	1	0	1	0
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	1	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Remark 1. For details on the LVD circuit, see **CHAPTER 21 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a TYP. value. For details, see **29.7.6 LVD circuit characteristics**.

(**Cautions** are listed on the next page.)

Figure 21 - 4 Format of User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	0	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

- LVD off (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting of values other than above is prohibited.						

Caution 1. Set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 29.5 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H).

Remark 1. ×: Don’t care

Remark 2. For details on the LVD circuit, see **CHAPTER 21 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **29.7.6 LVD circuit characteristics**.

21.4 Operation of Voltage Detector

21.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

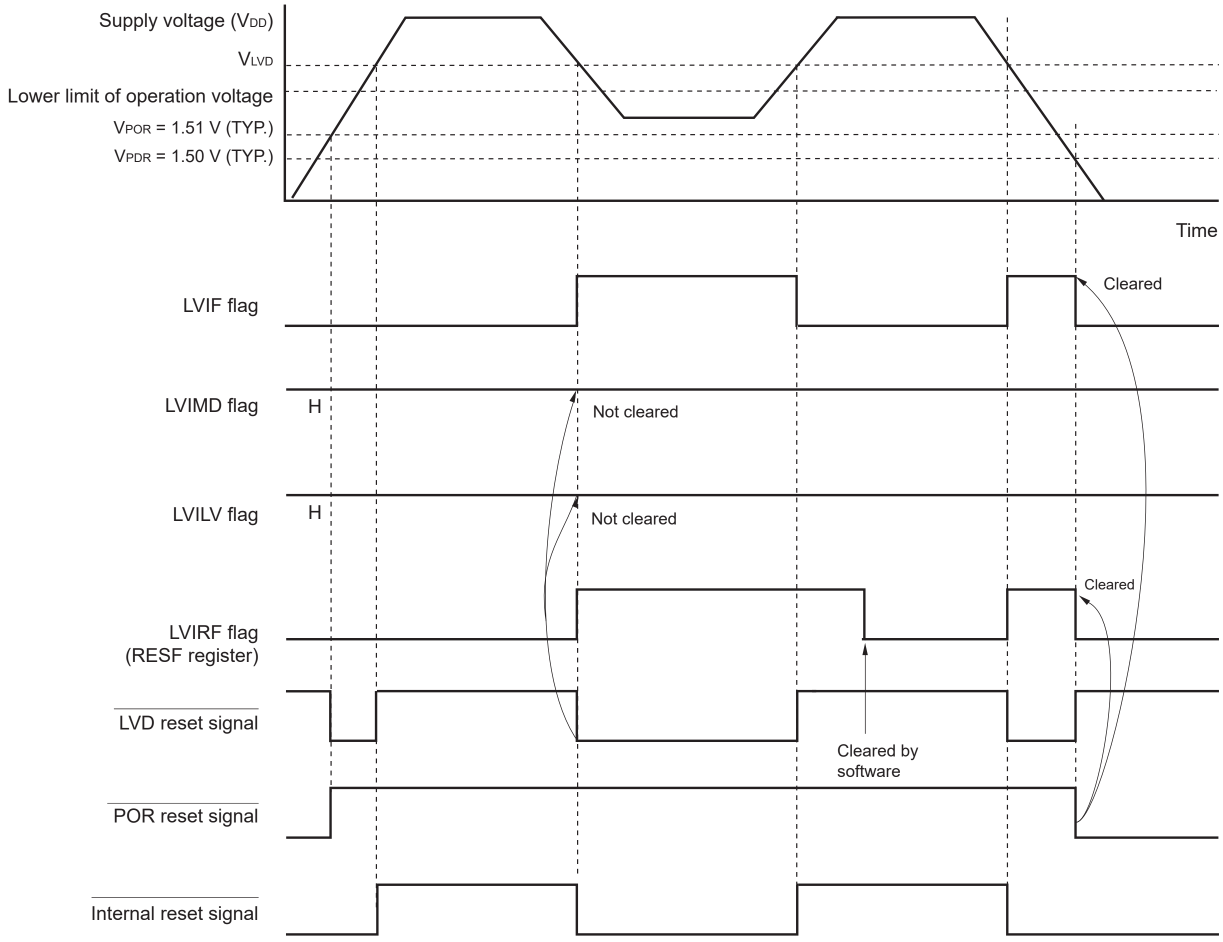
The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (voltage detection level: V_{LVD}).

- Operation in LVD reset mode
  In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

Figure 21 - 5 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 21 - 5 Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

21.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

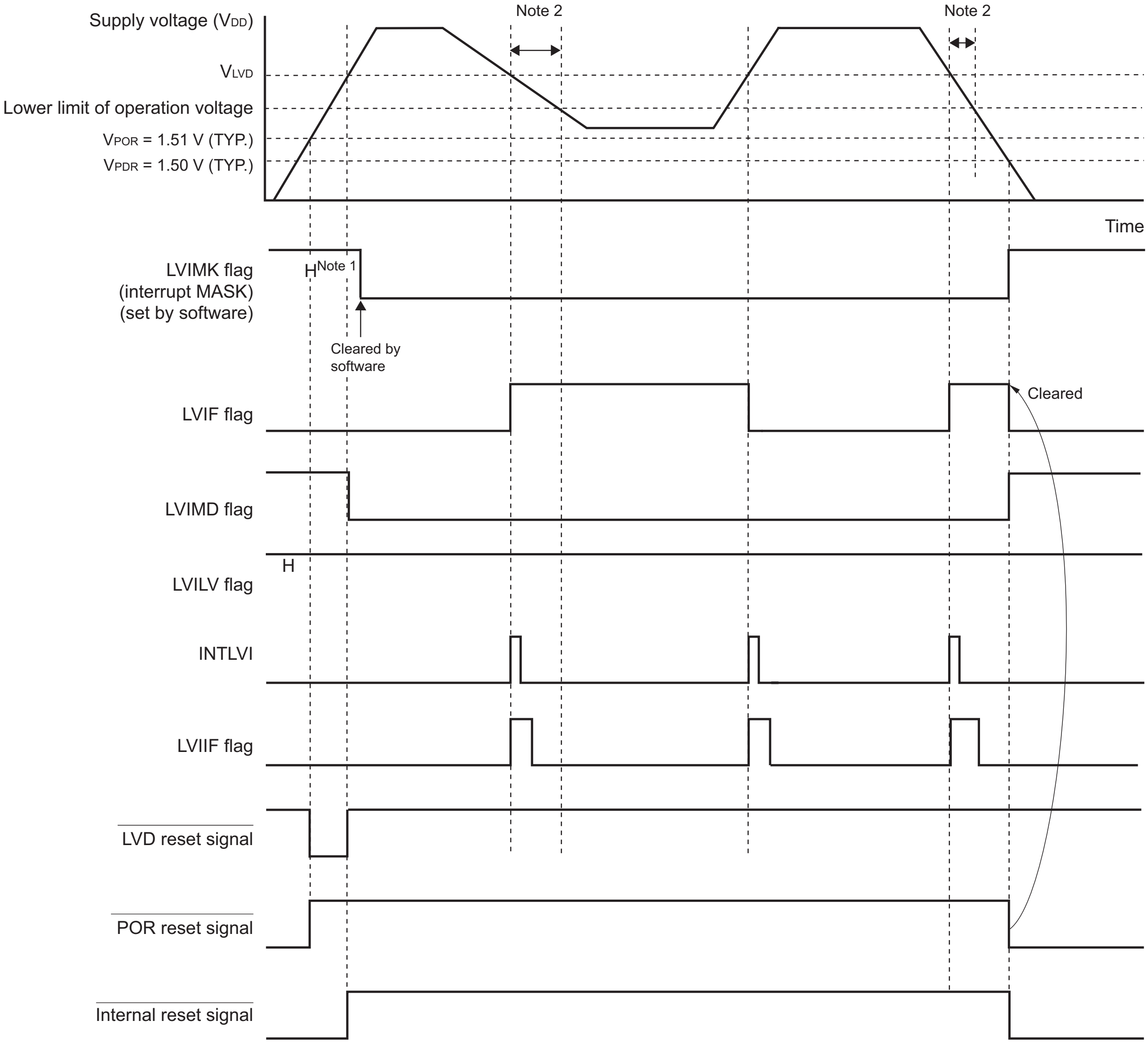
The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (voltage detection level: V_{LVD}).

- Operation in LVD interrupt mode
  In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **29.5 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 21 - 6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 21 - 6 Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **29.5 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

21.4.3 When used as interrupt & reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

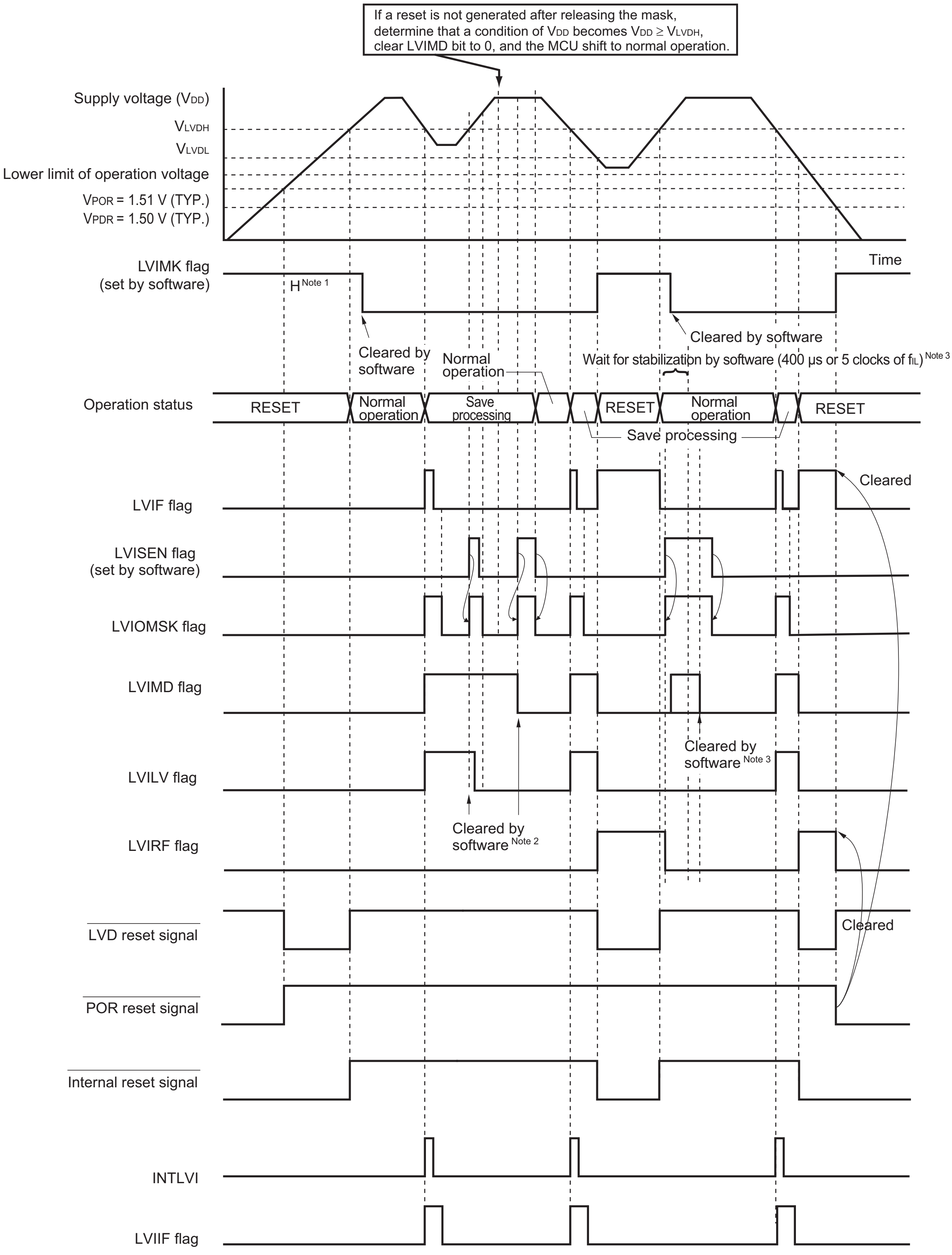
The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

- Operation in LVD interrupt & reset mode
  In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}). An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}). To use the LVD reset & interrupt mode, perform the processing according to **Figure 21 - 8 Processing Procedure After an Interrupt Is Generated** and **Figure 21 - 9 Initial Setting of Interrupt and Reset Mode**.

Figure 21 - 7 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 21 - 7 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



(**Notes** and **Remark** are listed on the next page.)

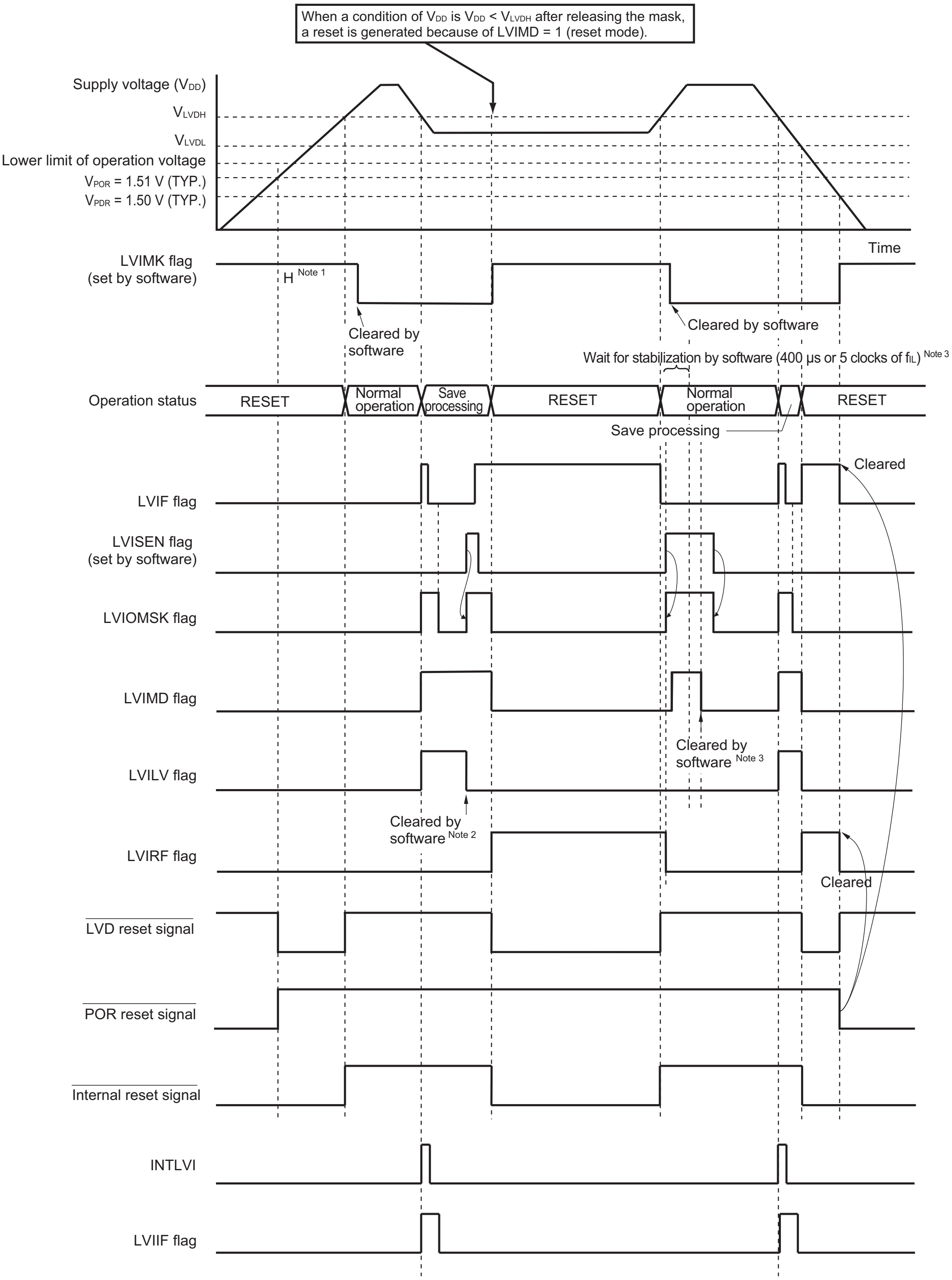
Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 21 - 8 Processing Procedure After an Interrupt Is Generated in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 21 - 9 Initial Setting of Interrupt and Reset Mode.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Figure 21 - 7 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



(**Notes** and **Remark** are listed on the next page.)

Note 1. The LVIMK flag is set to “1” by reset signal generation.

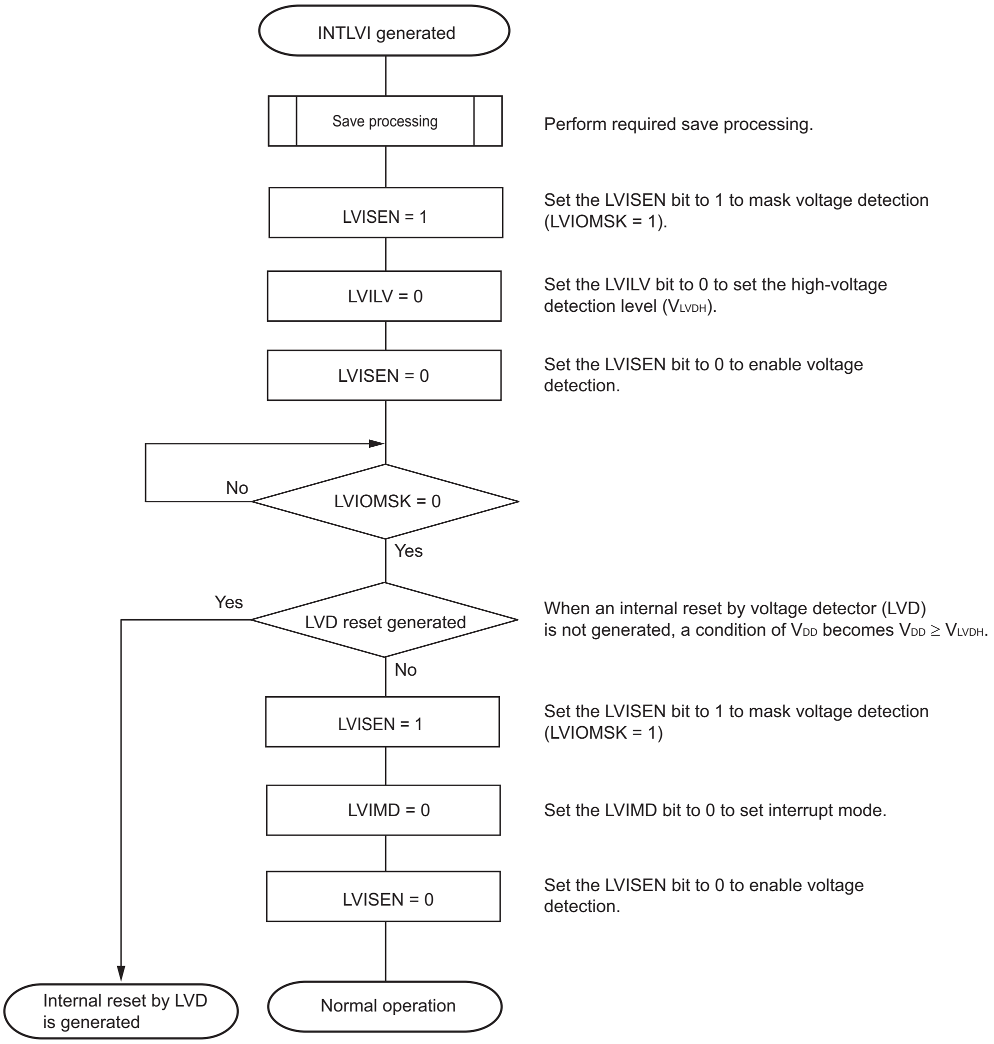
Note 2. After an interrupt is generated, perform the processing according to Figure 21 - 8 Processing Procedure After an Interrupt Is Generated in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 21 - 9 Initial Setting of Interrupt and Reset Mode.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

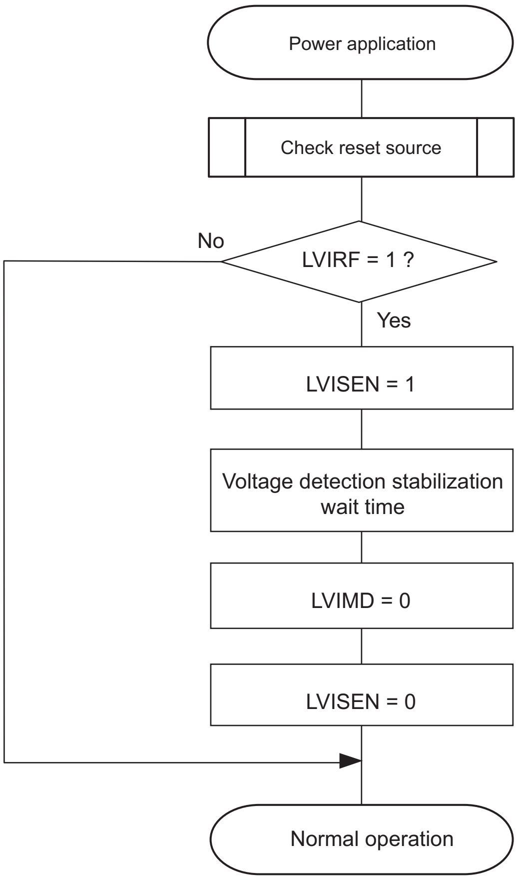
Figure 21 - 8 Processing Procedure After an Interrupt Is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 21 - 9 shows the procedure for initial setting of interrupt and reset mode.

Figure 21 - 9 Initial Setting of Interrupt and Reset Mode



See **Figure 19-6 Example of Procedure for Checking Reset Source**.

Check internal reset generation by LVD circuit

Set the LVISEN bit to 1 to mask voltage detection (LVIOMSK = 1)

Count 400 μs or 5 clocks of f_{IL} by software.

Set the LVIMD bit to 0 to set interrupt mode.

Set the LVISEN bit to 0 to enable voltage detection.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

21.5 Cautions for Voltage Detector

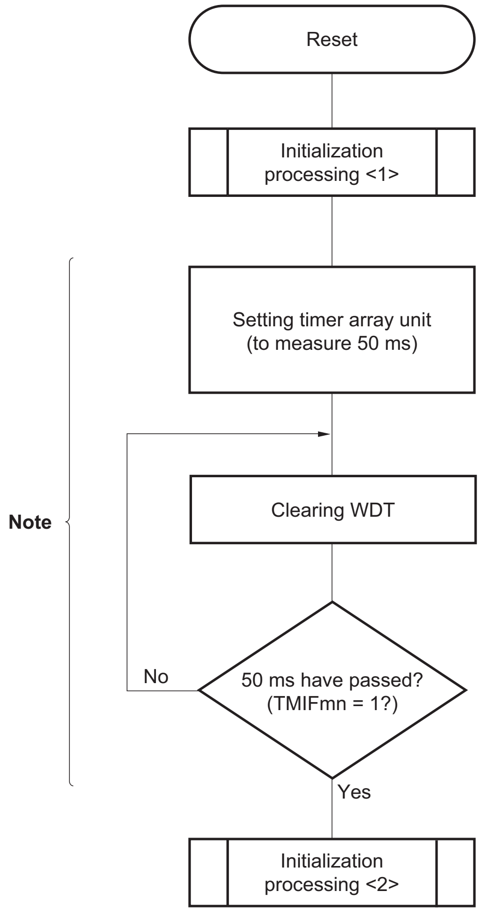
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 21 - 10 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



See **Figure 19-6 Example of Procedure for Checking Reset Source**.

; e.g. f_{CLK} = High-speed on-chip oscillator clock (4.04 MHz (MAX.))
Source: f_{MCK} = (4.04 MHz (MAX.))/2^8,
where comparison value = 789: ≅ 50 ms
Timer starts (TSmn = 1).

; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note If reset is generated again during this period, initialization processing <2> is not started.

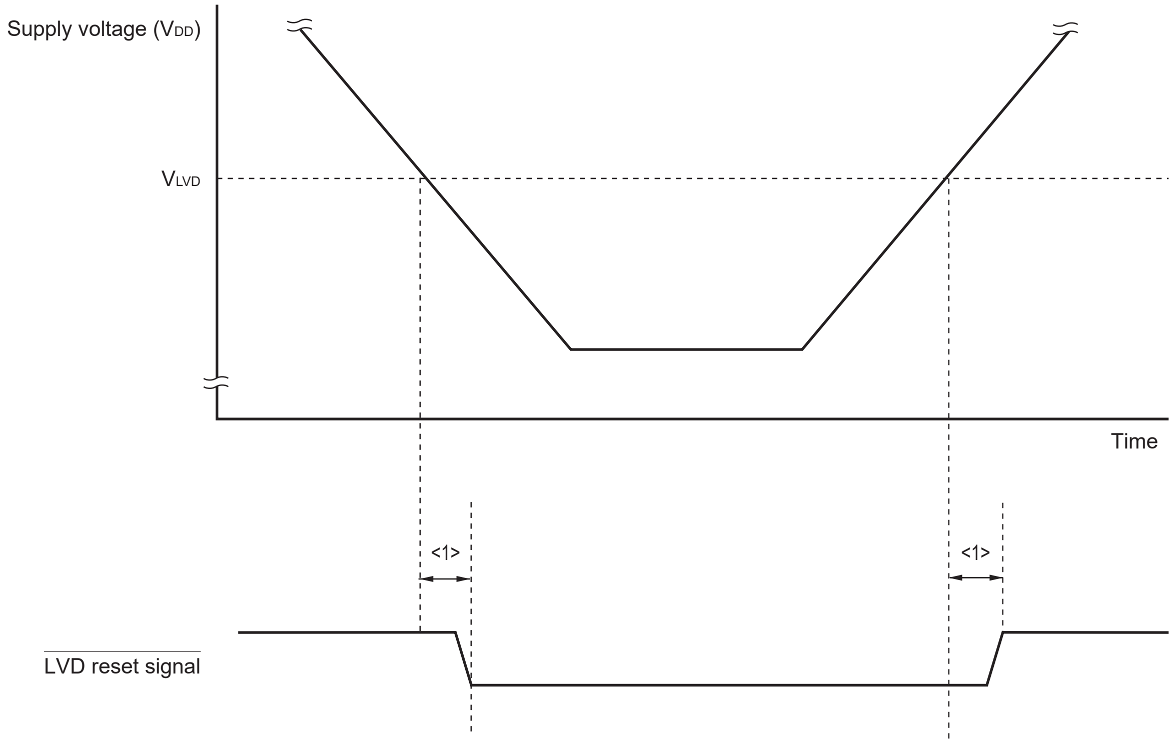
Remark m = 0
n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 21 - 11**).

Figure 21 - 11 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **29.5 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **29.5 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 22 SAFETY FUNCTIONS

22.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/G1G to comply with the IEC60730 safety standards. These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the RL78/G1G that can be used according to the application or purpose of use.

• High-speed CRC:	The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
• General CRC:	This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when the RAM is read as data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function

This uses the timer array unit to perform a self-check of the CPU/peripheral hardware clock frequency.

(7) A/D test function

This is used to perform a self-check of A/D converter by performing A/D conversion on the positive internal reference voltage, negative reference voltage, analog input channel (ANI), temperature sensor output, and internal reference voltage output.

(8) Digital output signal level detection function for I/O ports

When the I/O ports are output mode in which PMmn bit of the port mode register (PMm) is 0, the output level of the pin can be read.

Remark m = 0 to 7, 12, 14, n = 0 to 7

22.2 Registers Used by Safety Functions

The safety functions use the following registers:

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O ports

The content of each register is described in Operation of Safety Functions below.

22.3 Operation of Flash Memory CRC Operation Function (High-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/G1G can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

The high-speed CRC stops the CPU to read and calculate 32 bits of data from the flash memory in one clock. This feature allows a shorter time until checking is completed. (For example, 16-KB flash memory: 171 μs @ 24 MHz)

The CRC generator polynomial used complies with “$X^{16} + X^{12} + X^5 + 1$” of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Caution The CRC operation result might differ during on-chip debugging because the monitor program is allocated.

Remark The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

22.3.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.

The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 1 Format of Flash memory CRC control register (CRC0CTL)

Address: F02F0H After reset:00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	00000H to 3FFBH (16 K to 4 bytes)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

22.3.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 22 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

7	6	5	4	3	2	1	0
PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

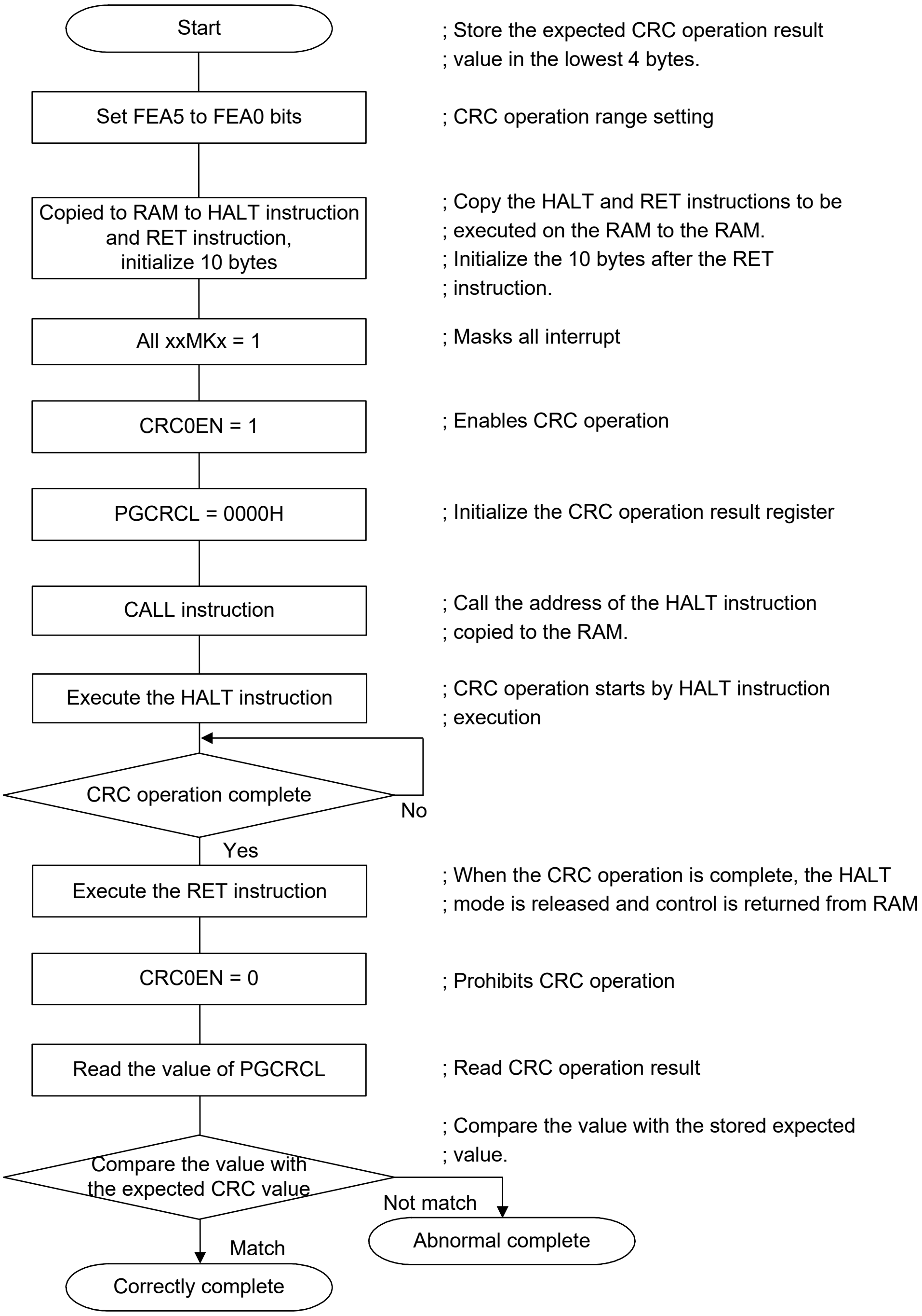
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

22.3.3 Operation flow

Figure 22 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

Figure 22 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.

Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.

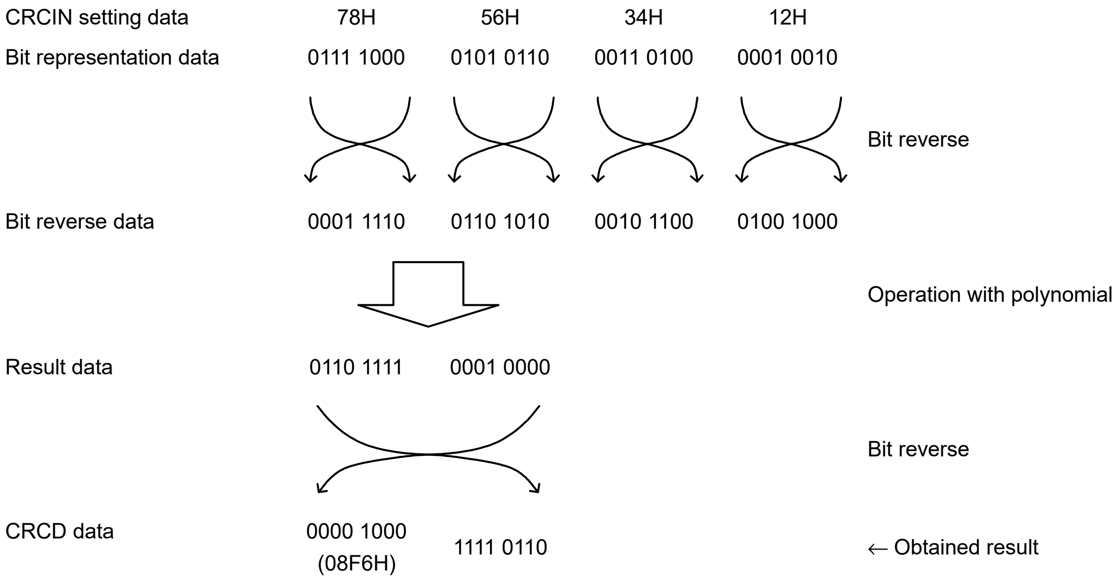
Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

The expected CRC operation value can be calculated by using the integrated development environment. CubeSuite+ development environment. Refer to the CubeSuite+ integrated development environment user’s manual for details.

22.4 CRC Operation Function (General-Purpose CRC)

<R> In the RL78/G1G, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program).

The CRC generator polynomial used is “$X^{16} + X^{12} + X^5 + 1$” of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution **Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.**

22.4.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.
The possible setting range is 00H to FFH.
The CRCIN register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 4 Format of CRC input register (CRCIN)

Address:FFFACH After reset:00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

22.4.2 CRC data register (CRCD)

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 22 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

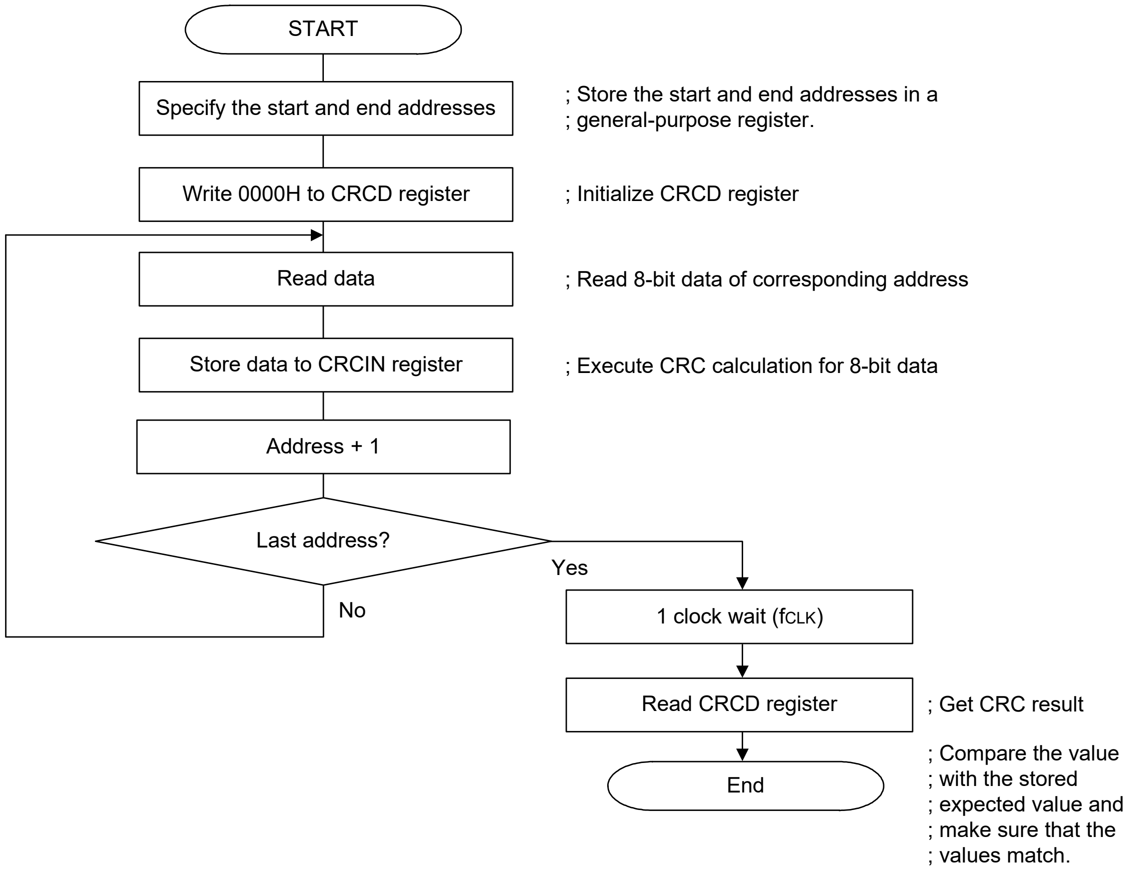
Caution 1. Read the value written to CRCD register before writing to CRCIN register.

Caution 2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

22.4.3 Operation flow

Figure 22 - 6 shows the CRC Operation Function (General-Purpose CRC).

Figure 22 - 6 CRC Operation Function (General-Purpose CRC)



22.5 RAM Parity Error Detection Function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/G1G's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

22.5.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.
The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 7 Format of RAM parity error control register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

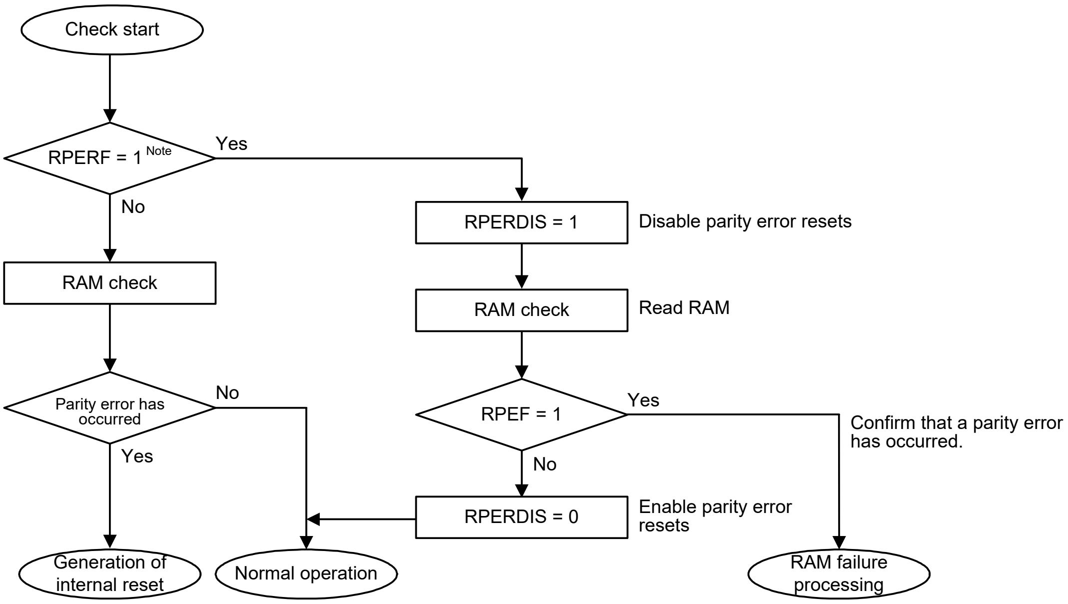
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution **This CPU executes lookahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error.**
Therefore, when enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes. When using the self-programming function while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area to overwrite + 10 bytes before overwriting.
The data read by the instruction is subject to parity error detection.

Remark 1. The RAM parity check is always on, and the result can be confirmed by checking the PREF flag.
Remark 2. The parity error reset is enabled by default (RPERDIS = 0).
Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs.
Remark 3. The RPEF flag is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source.
When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

Figure 22 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 19 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

22.6 RAM Guard Function

<R> This RAM guard function is used to protect data in the specified memory space.
If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

22.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.
GRAM1 and GRAM0 bits are used in RAM guard function.
The IAWCTL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the lower RAM address
1	0	The 256 bytes starting at the lower RAM address
1	1	The 512 bytes starting at the lower RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product. (Refer to **Figure 22 - 11**).
The general-purpose register area (FFEE0H to FFEFFH) is not protected.

22.7 SFR Guard Function

<R> This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

22.7.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIOR Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL

Note Pxx (Port register) is not guarded.

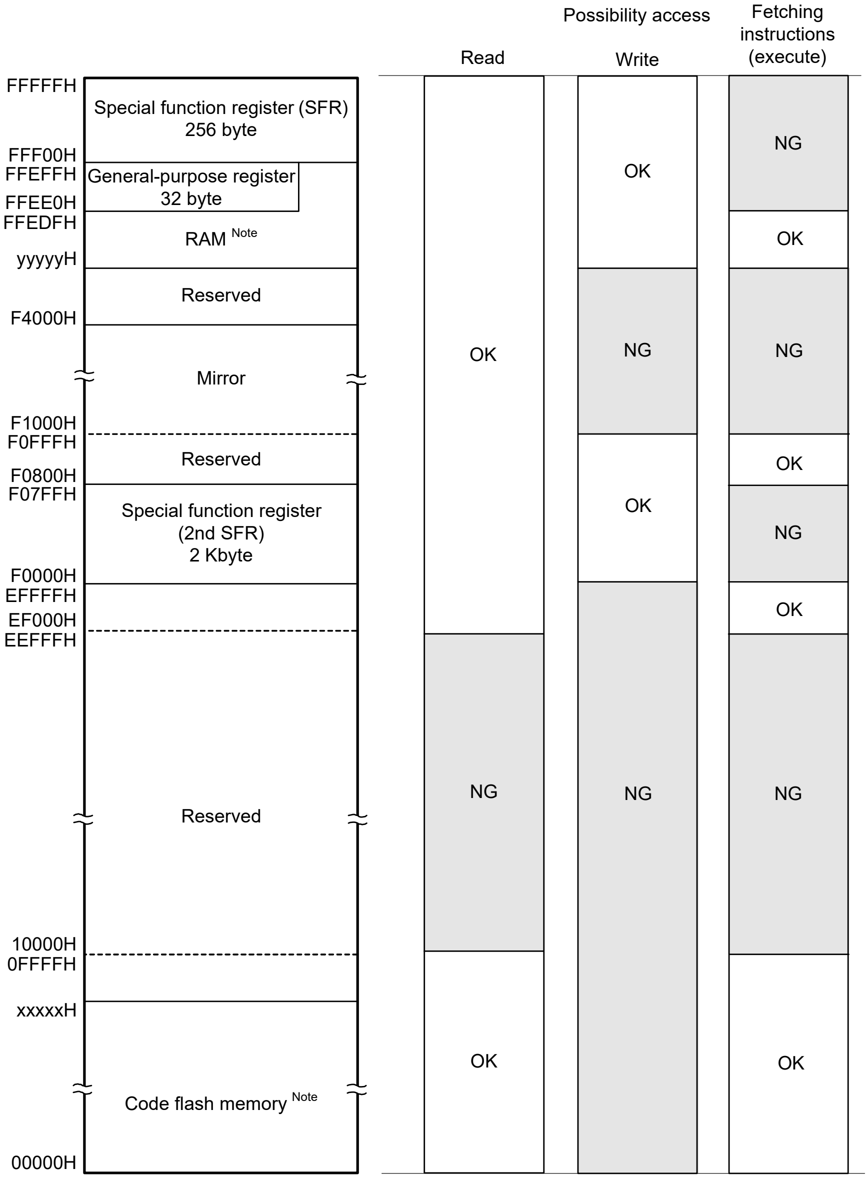
22.8 Invalid Memory Access Detection Function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 22 - 11.

Figure 22 - 11 Invalid Access Detection Area



Note Code flash memory and RAM address of each product are as follows.

Products	Code flash memory (00000H to xxxxxH)	RAM (yyyyyH to FFEFFH)
R5F11Ex8 (x = A, B, F)	8192 × 8 bit (00000H to 01FFFH)	1536 × 8 bit (FF900H to FFEFFH)
R5F11ExA (x = A, B, F)	16384 × 8 bit (00000H to 03FFFH)	

22.8.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN Note	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark When the option byte WDTON = 1 (watchdog timer operation enabled), the invalid memory access detection function is enabled even when IAWEN = 0.

22.9 Frequency Detection Function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

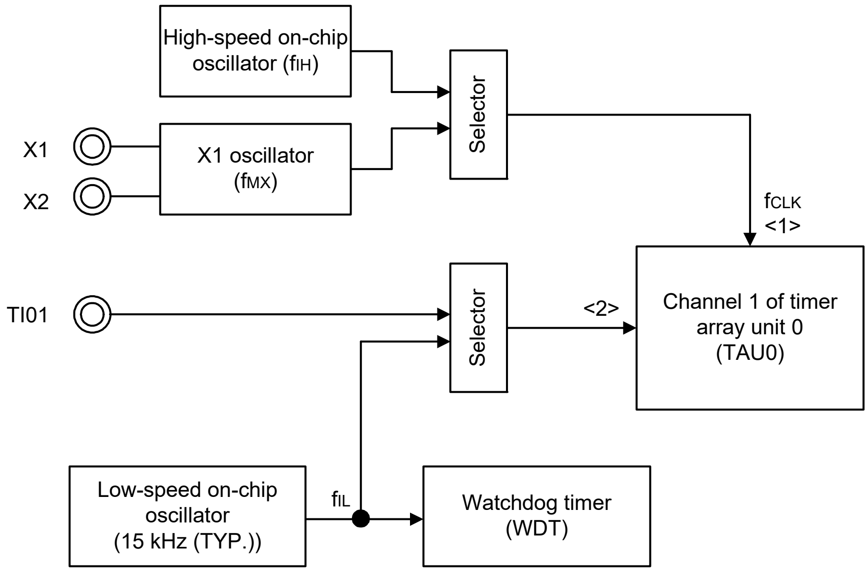
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
  - High-speed on-chip oscillator clock (f_{IH})
  - High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit 0
  - Timer input to channel 1 (TI01)
  - Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))

Figure 22 - 13 Configuration of Frequency Detection Function



If pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute pulse interval measurement, see **6.8.4 Operation as input pulse interval measurement**.

22.9.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channels 0 and 1 of the timer array unit 0 (TAU0).

By selecting the low-speed on-chip oscillator clock for the timer input, its pulse width can be measured to determine whether the proportional relationship between the low-speed on-chip oscillator clock and the timer operation clock is correct.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 14 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

22.10 A/D Test Function

The IEC60730 standard mandates testing the A/D converter. The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of the positive reference voltage and negative reference voltage of the A/D converter, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage. For details on the checking method, refer to the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

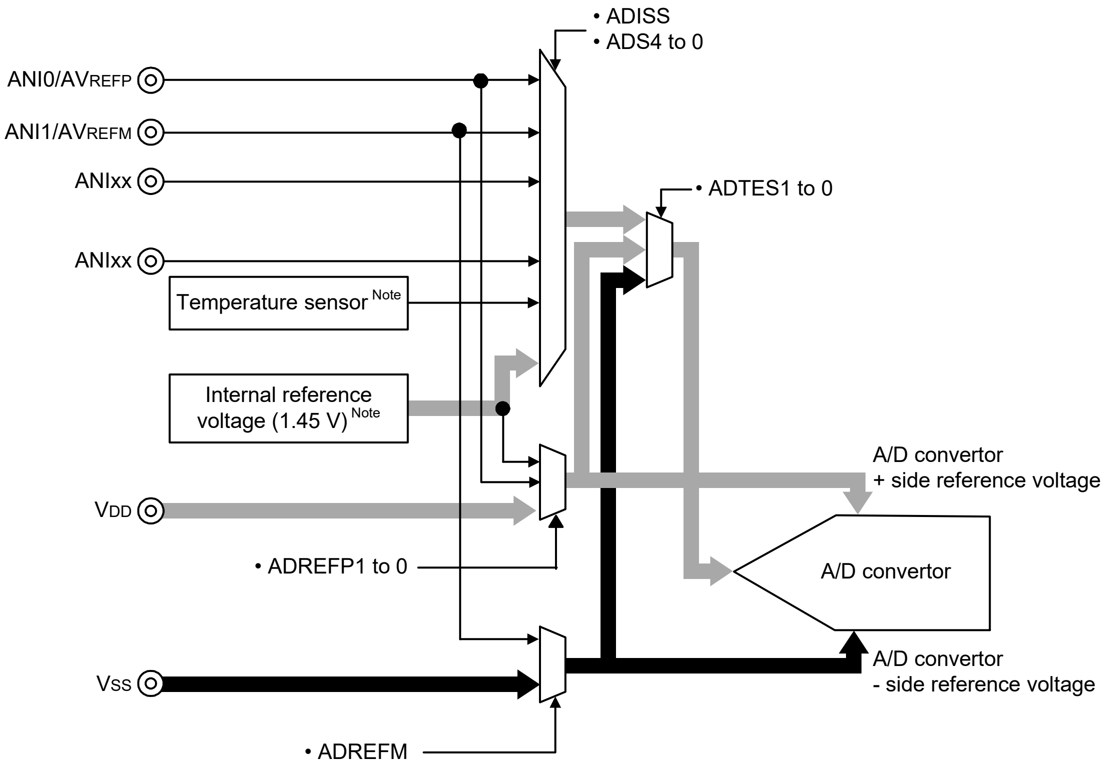
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 0).
(4) Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
(5) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(6) Perform A/D conversion for the ANIx pin (conversion result 1-2).
(7) Select the positive reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 1).
(8) Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
(9) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(10) Perform A/D conversion for the ANIx pin (conversion result 1-3).
(11) Make sure that “conversion result 1-1” = “conversion result 1-2” = “conversion result 1-3”.
(12) Make sure that the A/D conversion results of “conversion result 2-1” are all 0 and those of “conversion result 2-2” are all 1.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during conversion in steps (1) to (10) above, use another method to check the analog multiplexer.

Remark 2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 22 - 15 Configuration of A/D Test Function



Note Selectable only in HS (high-speed main) mode.

22.10.1 A/D test register (ADTES)

This register is used to select the A/D converter’s positive reference voltage AVREFP, the A/D converter’s negative reference voltage AVREFM, or the analog input channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select AVREFM as the target of A/D conversion when converting the internal 0 V.
- Select AVREFP as the target of A/D conversion when converting AVREF.

The ADTES register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output [Note]/internal reference voltage (1.45 V) [Note] (This is specified using the analog input channel specification register (ADS).)
1	0	AVREFM
1	1	AVREFP
Other than the above		Setting prohibited

Note Temperature sensor output/internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

22.10.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output /internal reference voltage (1.45 V).

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 17 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV$_{REFP}$ pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV$_{REFM}$ pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	0	1	0	0	—	PGAO
1	0	0	0	0	0	—	Temperature sensor output Note
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Only rewrite the value of the ADISS bit while operation of the A/D voltage comparator is stopped (ADCS = 0 and ADCE = 0 in A/D converter mode register 0 (ADM0)).

Caution 3. If using AV$_{REFP}$ as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 4. If using AV$_{REFM}$ as the - side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 5. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage.

22.11 Digital Output Signal Level Detection Function for I/O Ports

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O ports, the digital output level of the pin can be read when the port is set to output mode (the PMmn bit in the port mode register (PMm) is 0).

22.11.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the port is output mode in which PMmn bit of port mode register (PMm) is 0.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 22 - 18 Format of Port mode select register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when port is output mode (PMmn = 0)
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 1. When the PMS0 bit in the PMS register is set to 1, do not rewrite the port register (Pm) using a read-modify-write instruction. Use an 8-bit MOV instruction to rewrite the Pm register.

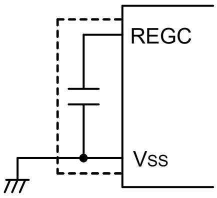
Caution 2. When the digital output level of a pin that is held in the high-impedance state by the timer RD pulse output forced cutoff function, the read value is 0.

Remark m = 0 to 7, 12, 14
n = 0 to 7

CHAPTER 23 REGULATOR

23.1 Regulator Overview

The RL78/G1G contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution **Keep the wiring length as short as possible for the broken-line part in the above figure.**

The regulator output voltage, see **Table 23 - 1**.

Table 23 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LS (low-speed main) mode	1.8 V	—
HS (high-speed main) mode	1.8 V	In STOP mode
	2.1 V	Other than above (include during OCD mode) Note

Note When it shifts to the STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 24 OPTION BYTE

24.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G1G form an option byte area.
Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).
Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.
For the bits to which no function is allocated, do not change their initial values.

Caution The option bytes should always be set regardless of whether each function is used.

24.1.1 User option byte (000C0H to 000C2H)

(1) 000C0H

○ Operation of watchdog timer
- Enabling or disabling of counter operation
- Operation is stopped or enabled in the HALT or STOP mode.

○ Setting of overflow time of watchdog timer

○ Setting of window open period of watchdog timer

○ Setting of interval interrupt of watchdog timer
- Used or not used

(2) 000C1H

○ Setting of LVD operation mode
- Interrupt & reset mode
- Reset mode
- Interrupt mode
- LVD off (external reset input from the $\overline{\text{RESET}}$ pin is used)

○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

(3) 000C2H

○ Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.

- LS (low speed main) mode
- HS (high speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator

- Select from 1 MHz, 4 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, and 48 MHz.

24.1.2 On-chip debug option byte (000C3H)

○ Control of on-chip debug operation

- On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication

- Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

24.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 24 - 1 Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
Other than above			Setting prohibited

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode Note
1	Counter operation enabled in HALT/STOP mode

Note The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 24 - 2 Format of User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- When used as interrupt & reset mode

Detection voltage			Option byte Setting Value						
VLVDH		VLVDL	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge	Falling edge							
2.92 V	2.86 V	2.75 V	1	0	0	1	1	1	0
3.02 V	2.96 V							0	1
4.06 V	3.98 V							0	0
Other than above			Setting prohibited						

- When used as reset mode

Detection voltage		Option byte Setting Value						
VLVD		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
2.81 V	2.75 V	1	1	0	1	1	1	1
2.92 V	2.86 V			0	1	1	1	0
3.02 V	2.96 V			0	1	1	0	1
3.13 V	3.06 V			0	0	1	0	0
3.75 V	3.67 V			0	1	0	0	0
4.06 V	3.98 V			0	1	1	0	0
Other than above		Setting prohibited						

- When used as interrupt mode

Detection voltage		Option byte Setting Value						
VLVD		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
2.81 V	2.75 V	0	1	0	1	1	1	1
2.92 V	2.86 V			0	1	1	1	0
3.02 V	2.96 V			0	1	1	0	1
3.13 V	3.06 V			0	0	1	0	0
3.75 V	3.67 V			0	1	0	0	0
4.06 V	3.98 V			0	1	1	0	0
Other than above		Setting prohibited						

Caution Be sure to set bit 4 to “1”.

Remark 1. For details on the LVD circuit, see **CHAPTER 21 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **29.7.6 LVD circuit characteristics**.

Figure 24 - 3 Format of User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- When LVDOFF

Detection voltage		Option byte Setting Value						
V_{LVD}		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
—	—	0/1	1	1	×	×	×	×
Other than above		Setting prohibited						

Caution 1. Be sure to set bit 4 to “1”.

Caution 2. When the LVD is off, it is necessary to perform an external reset. For an external reset, input a low level of at least 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin before power-on, keep the low level for at least 10 μs during the period in which the supply voltage is within the operating range, and then input a high level. After power is applied, do not input a high level to the $\overline{\text{RESET}}$ pin during a period in which the supply voltage is not within the operating range.

Remark ×: don’t care

Figure 24 - 4 Format of Option Byte (000C2H)

Address: 000C2H

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
1	0	LS (low speed main) mode	1 to 8 MHz	2.7 to 5.5 V
1	1	HS (high speed main) mode	1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	1	0	1	1	4 MHz	4 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Caution Be sure to set bit 5 to 1.

24.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 24 - 5 Format of On-chip Debug Option Byte (000C3H)

Address: 000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution **Only bit7 and bit0(OCDENSET and OCDERSD) can be specified a value.**
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the initial value (0, 1, 0) to bits 3 to 1 as default.

24.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

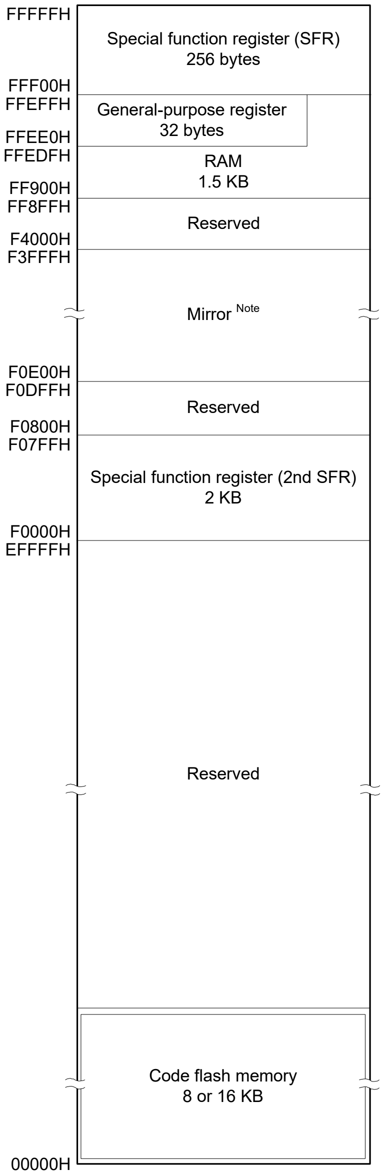
A software description example of the option byte setting is shown below.

```
OPT         CSEG        OPT_BYTE
            DB          36H         ; Does not use interval interrupt of watchdog timer,
                                    ; Enables watchdog timer operation,
                                    ; Window open period of watchdog timer is 50%,
                                    ; Overflow time of watchdog timer is 2^9/fIL,
                                    ; Stops watchdog timer operation during HALT/STOP mode
            DB          7AH         ; Select 2.75 V for VLVDL
                                    ; Select 2.92 V/2.86 V for VLVDH
                                    ; Select the interrupt & reset mode as the LVD operation mode
            DB          ADH         ; Select the LS (low speed main) mode as the flash operation mode
                                      and 1 MHz as the frequency of the high-speed on-chip oscillator clock
            DB          85H         ; Enables on-chip debug operation, does not erase flash memory
                                      data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

CHAPTER 25 FLASH MEMORY

The RL78 microcontroller incorporate the flash memory to which a program can be written, erased, and overwritten while mounted on the board.



Note Products with 8-Kbyte code flash memory cannot be used.

The following three methods for programming the flash memory are available:

- Writing to flash memory by using flash memory programmer (see **25.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Writing to flash memory by using external device (that Incorporates UART) (see **25.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-programming (see **25.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

25.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming
The contents of the flash memory can be rewritten after the RL78 microcontroller have been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming
Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller are mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 25 - 1 Wiring between RL78/G1G and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
Signal Name		I/O	Pin Function		30-pin	32-pin	44-pin
PG-FP5, FL-PR5	E1 on-chip debugging emulator				LSSOP	LQFP (7 × 7)	LQFP (10 × 10)
—	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	5	1	2
SI/RxD	—	I/O	Transmit/receive signal				
SCK	—	Output	—	—	—	—	—
CLK	—	Output	—	—	—	—	—
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	6	2	3
/RESET	—	Output					
FLMD0	—	Output	Mode signal	—	—	—	—
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	12	8	11
GND		—	Ground	V_{SS}	11	7	10
				REGC Note	10	6	9
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	12	8	11

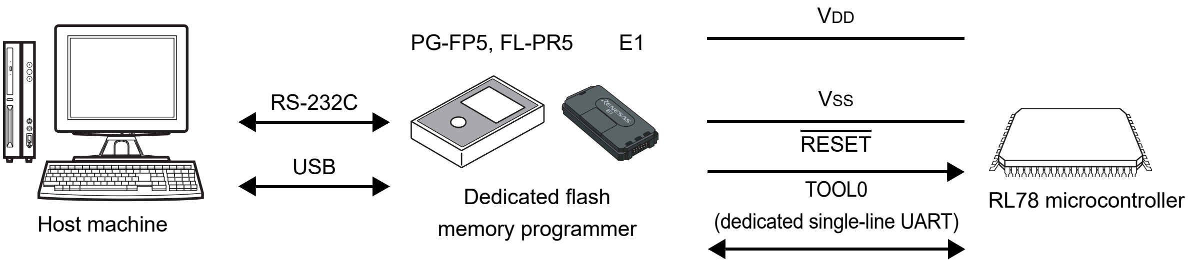
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

25.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 25 - 1 Environment for Writing Program to Flash Memory



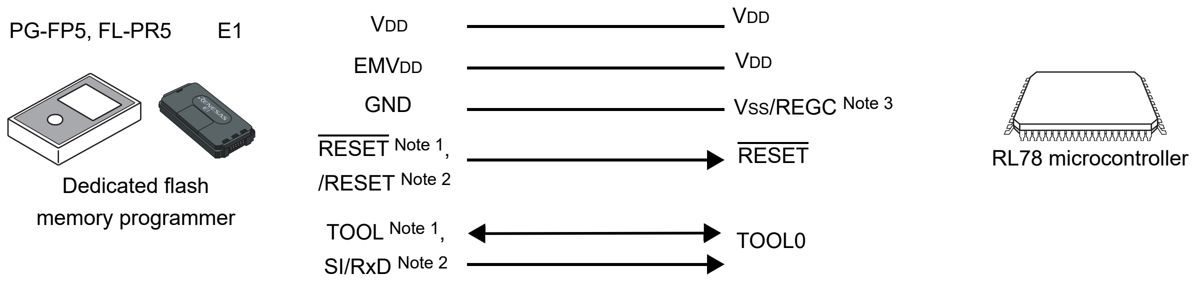
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

25.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 25 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1 on-chip debugging emulator.
Note 2. When using PG-FP5 or FL-PR5.
Note 3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 25 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name Note 2
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC Note 1
EMV_{DD}		—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	$\overline{\text{RESET}}$
—	$\overline{\text{RESET}}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Note 2. Pins to be connected differ with the product. For details, see **Table 25 - 1.**

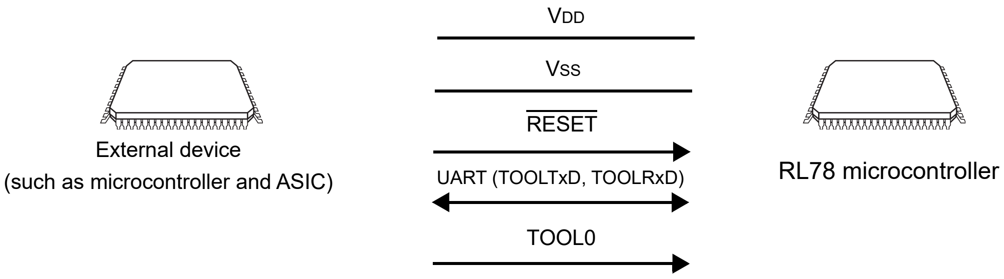
25.2 Serial Programming Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

25.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 25 - 3 Environment for Writing Program to Flash Memory



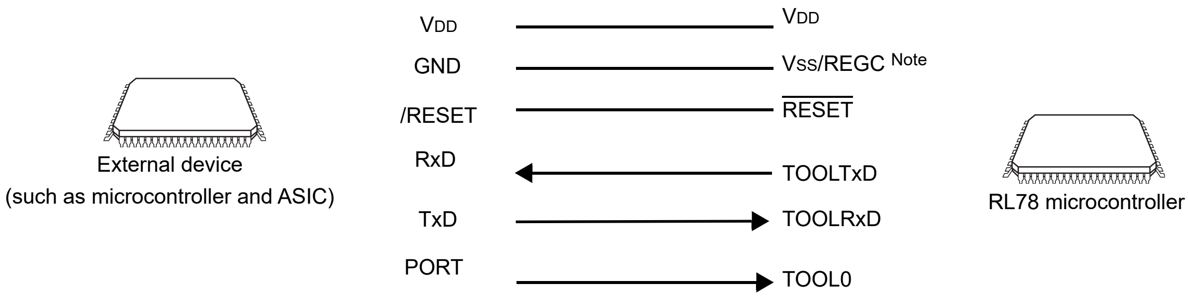
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

25.2.2 Communication mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 25 - 4 Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 25 - 3 Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD
GND	—	Ground	VSS, REGC Note
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

25.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark Refer to flash programming mode, see **25.4.2 Flash memory programming mode**.

25.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin:	Input of low-level is prohibited for t_{HD} period after pin reset release. Furthermore, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.
When used as an output pin:	When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remark 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **29.11 Timing for Switching Flash Memory Programming Modes**).

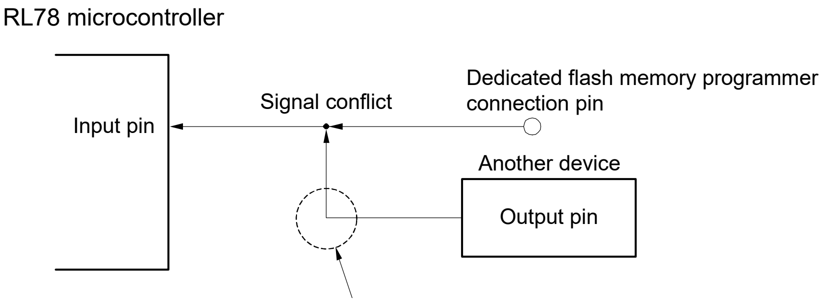
Remark 2. The SAU pin is not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

25.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 25 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

25.3.3 Port pins

Example When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either V_{DD} or V_{SS} via a resistor.

25.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

25.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

25.3.6 Power supply

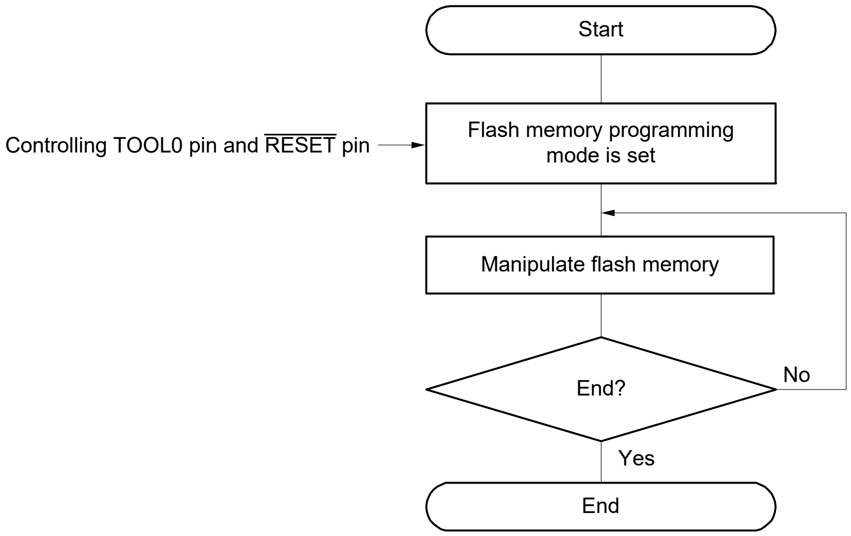
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

25.4 Programming Method

25.4.1 Controlling flash memory

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 25 - 6 Flash Memory Manipulation Procedure

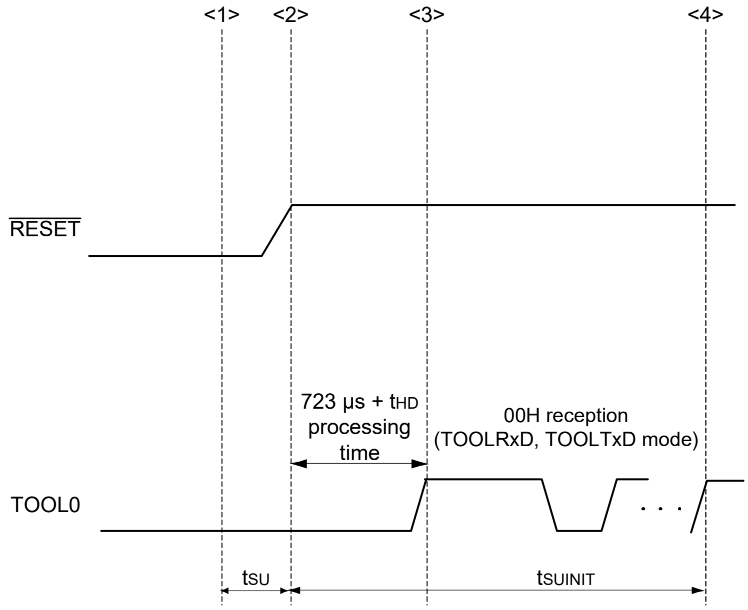


25.4.2 Flash memory programming mode

To rewrite the contents of the flash memory, set the RL78/G1G in the flash memory programming mode. To enter the mode, set as follows.

<When programming by using the dedicated flash memory programmer>
Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.
<When programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset. Keep the TOOL0 pin at the low level from the reset ends to 1 ms + software processing end, and then use UART communication to send the data “00H” from the external device. Finish UART communication within 100 ms after the reset ends.

Figure 25 - 7 Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded).

For details, see **29.11 Timing for Switching Flash Memory Programming Modes**.

Table 25 - 4 Relationship between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.
When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 25 - 5 Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified
Wide voltage mode	2.7 V to 5.5 V
Full speed mode Note	

Note This can only be specified if the CMODE1 and CMODE0 bits of the option byte 000C2H are 1.

Specify the mode that corresponds to the voltage range in which to write data. When programming by using the dedicated flash memory programmer, the mode is automatically selected by the voltage setting on GUI.

Remark 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.
Remark 2. For details about communication commands, see **25.4.4 Communication commands**.

25.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 25 - 6 Communication Modes

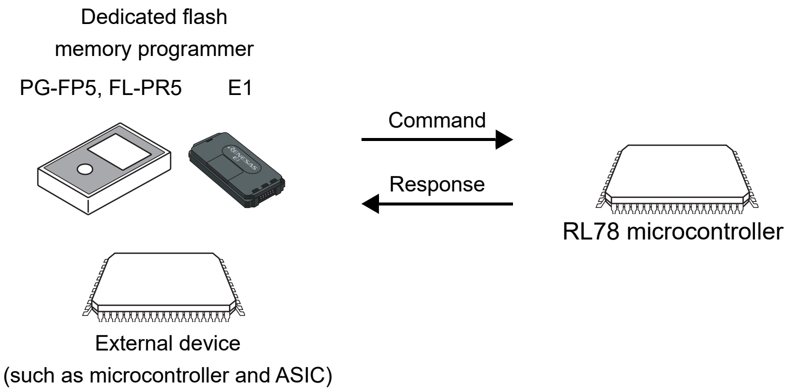
Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
UART0 (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for Standard settings on GUI of the flash memory programmer.
Note 2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

25.4.4 Communication commands

The RL78 microcontroller communicate with the dedicated flash memory programmer or external device by using commands. The signals sent from the flash memory programmer or external device to the RL78 microcontroller are called commands, and the signals sent from the RL78 microcontroller to the dedicated flash memory programmer or external device are called response.

Figure 25 - 8 Communication Commands



The flash memory control commands of the RL78 microcontroller are listed in the table below. All these commands are issued from the programmer or external device, and the RL78 microcontroller perform processing corresponding to the respective commands.

Table 25 - 7 Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory Note.
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

The RL78 microcontroller return a response for the command issued by the dedicated flash memory programmer or external device. The response names sent from the RL78 microcontroller are listed below.

Table 25 - 8 Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

25.4.5 Description of signature data

When the “silicon signature” command is performed, the product information (such as product name and firmware version) can be obtained.

Tables 25 - 9 and 25 - 10 show signature data list and example of signature data list.

Table 25 - 9 Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 03FFFH (16 KB) → FFH, 3FH, 00H)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 25 - 10 Signature Data List

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	Serial number	3 bytes	10 00 03
Device name	R5F11EF8	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 31 = “1” 45= “E” 46 = “F” 38 = “8” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 01FFFH (8 KB)	3 bytes	FF 1F 00
Firmware version	Ver.1.23	3 bytes	01 02 03

25.5 Security Settings

The RL78 microcontroller a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.
The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write command for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.
  After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set only by on-board/off-board programming. Each security setting can be used in combination.
Table 25 - 11 shows the relationship between the erase and write commands when the RL78 microcontroller security functions are enabled.

Caution The security function of the flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **25.6.1** for detail).

Table 25 - 11 Relationship between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **25.6.1** for detail).

Table 25 - 12 Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Execute security release command
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution **The security release command can be applied only when the security is set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area being blanks.**

25.6 Flash Memory Programming by Self-Programming

The RL78 microcontroller support a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Caution 1. Interrupts are prohibited during self-programming. Execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the RAM area to use + 10 bytes before overwriting.

Caution 3. The high-speed on-chip oscillator should be kept operating during self-programming.
If it is kept stopped, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed when the FRQSEL4 in the user option byte (000C2H) is 0, and after 80 μs have elapsed when the FRQSEL4 is 1.

Remark 1. For details of the self-programming function refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual (R01US0050E)**.

Remark 2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

Similar to when writing data by using the flash memory programmer, there are two flash memory programming modes.

Table 25 - 13 Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified	Writing Clock Frequency
Wide voltage mode	2.7 V to 5.5 V	8 MHz (MAX.)
Full speed mode Note		24 MHz (MAX.)

Note This can only be specified if the CMODE1 and CMODE0 bits of the option byte 000C2H are 1.

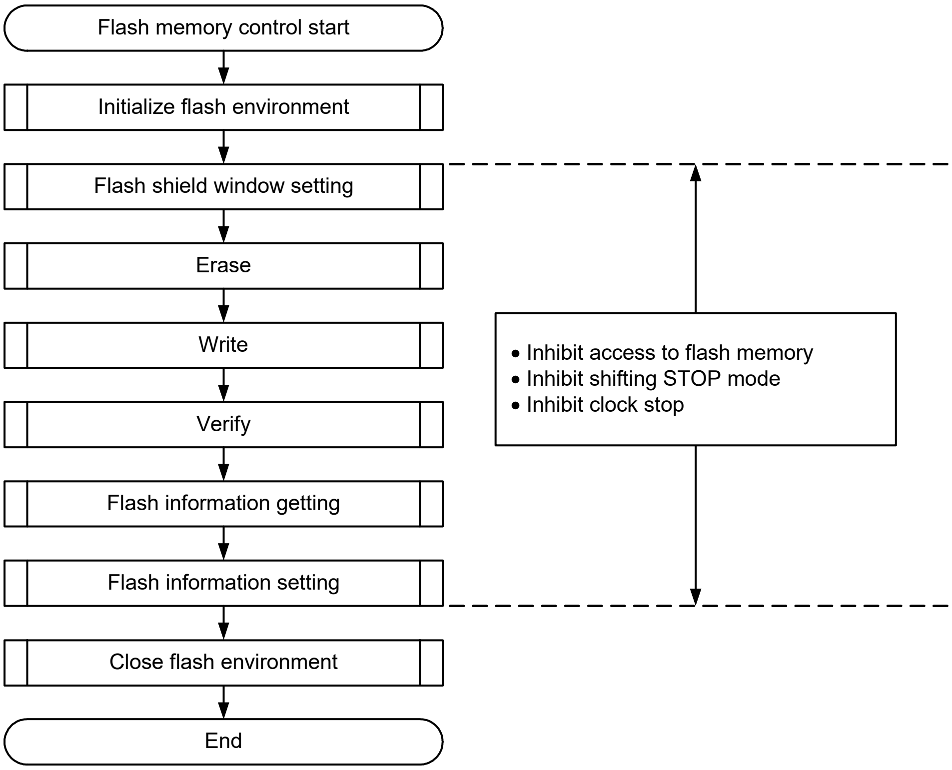
Specify the mode that corresponds to the voltage range in which to write data. If the argument fsl_flash_voltage_u08 is other than 00H when the FSL_Init function of the self programming library provided by Renesas Electronics is executed, wide-voltage mode is specified. If the argument is 00H, full-speed mode is specified.

Remark 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

Remark 2. For details of the self-programming function refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual (R01US0050E)**.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

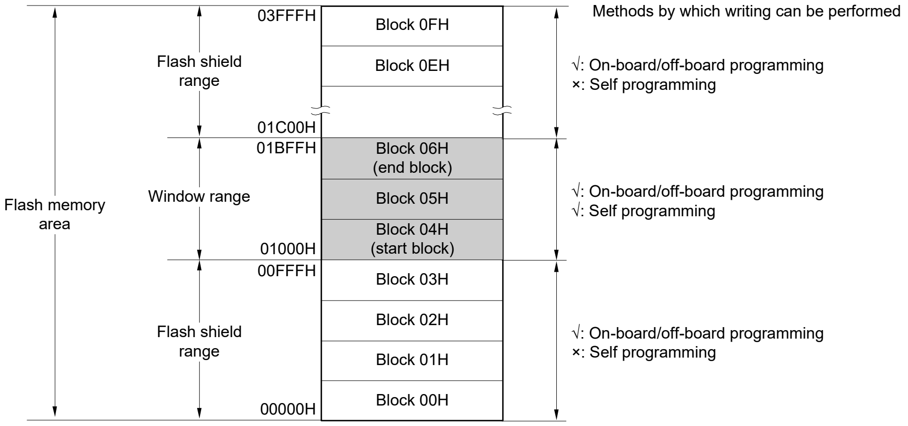
Figure 25 - 9 Flow of Self Programming (Rewriting Flash Memory)



25.6.1 Flash shield window function

The flash shield window function is provided as one of the security functions for self programming. It disables writing to and erasing areas outside the range specified as a window only during self programming.
The window range can be set by specifying the start and end blocks. The window range can be set or changed only during on-board/off-board programming.
Writing to and erasing areas outside the window range are disabled during self programming. The window range can be set or changed only during on-board/off-board programming.

Figure 25 - 10 Flash Shield Window Setting Example
(Target Devices: R5F11EAA, Start Block: 04H, End Block: 06H)



Caution 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

Caution 2. The flash shield window can only be used for the code flash memory.

Table 25 - 14 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming conditions	Window Range Setting/ Change Methods	Execution Commands	
		Block erase	Write
On-board/Off-board programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **25.5 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

25.7 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 25 - 15 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Port: TOOL0 (UART)	
	Speed: 1M bps	
	8 Kbytes	16 Kbytes
Erasing	1 s	1 s
Writing	1 s	1.5 s
Verification	1 s	1.5 s
Writing after erasing	1 s	1.5 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port:	TOOL0 (single-line UART)
Speed:	1,000,000 bps
Mode:	Full speed mode (flash operation mode: HS (high speed main) mode)

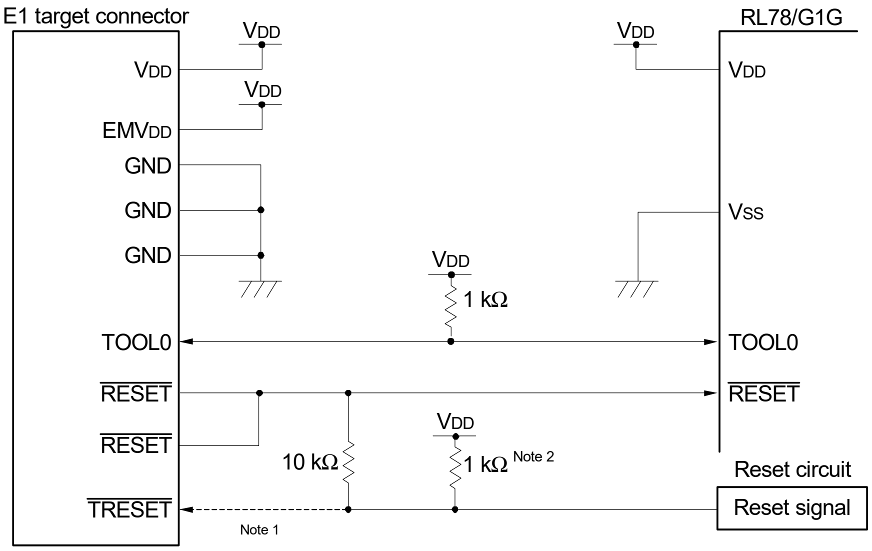
CHAPTER 26 ON-CHIP DEBUG FUNCTION

26.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 26 - 1 Connection Example of E1 On-chip Debugging Emulator



Note 1. Connecting the dotted line is not necessary during flash programming.

Note 2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

26.2 On-chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 24 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 26 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytesNote

Note The setting FFFFFFFFFFFFFFFFFFFFH for the ID code is not possible.

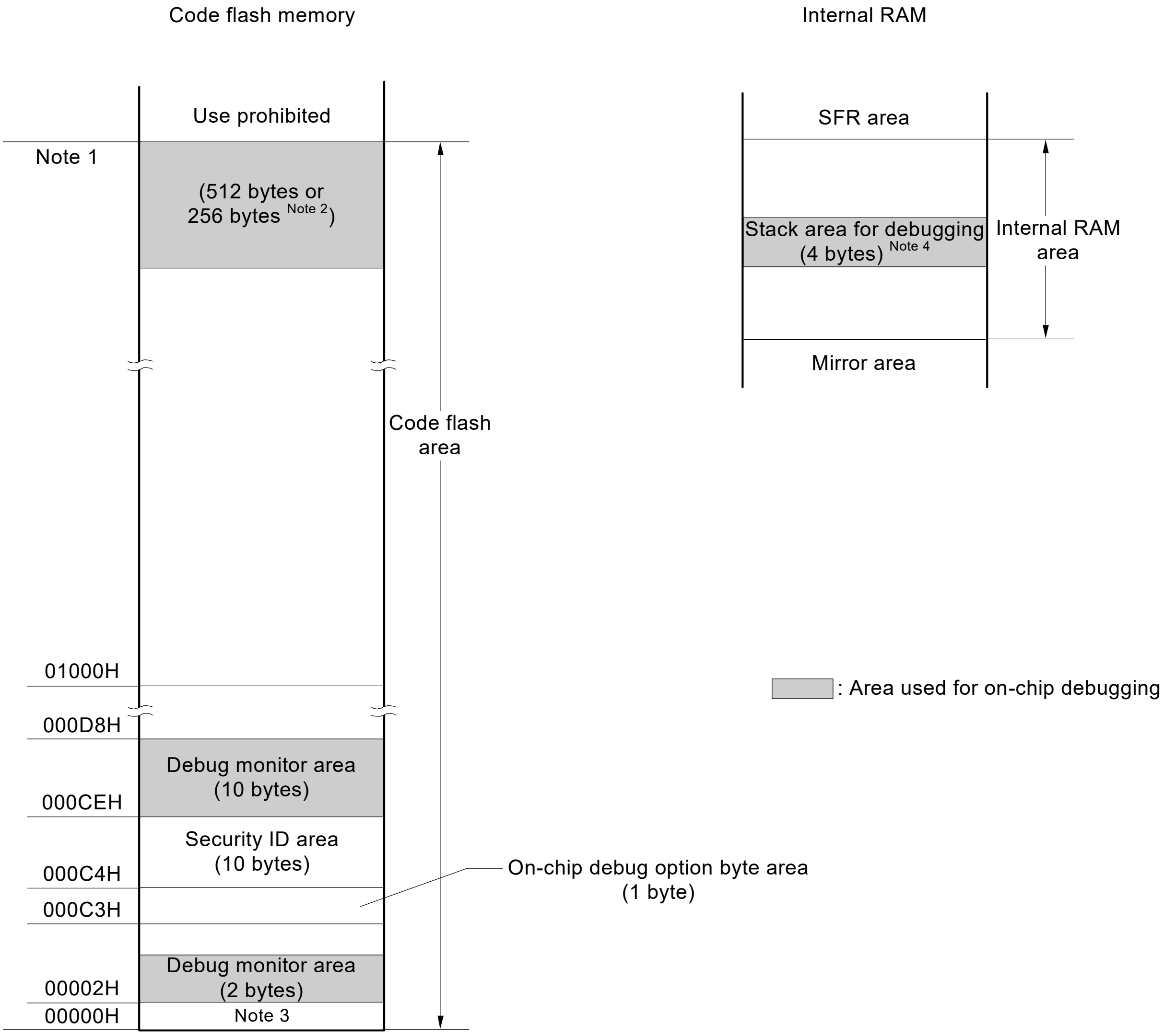
26.3 Securing of User Resources

To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.
If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 26 - 2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 26 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1.**
R5F11Ex8 (x = A, B, F)	01FFFH
R5F11ExA (x = A, B, F)	03FFFH

Note 2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 3. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 27 BCD CORRECTION CIRCUIT

27.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

27.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

(1) BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 27 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

27.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

The examples are shown below.

Example 1: 99 + 89 = 188

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #99H	; <1>	99H	—	—	—
ADD	A, #89H	; <2>	22H	1	1	66H
ADD	A, !BCDADJ	; <3>	88H	1	0	—

Example 2: 85 + 15 = 100

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #85H	; <1>	85H	—	—	—
ADD	A, #15H	; <2>	9AH	0	0	66H
ADD	A, !BCDADJ	; <3>	00H	1	1	—

Example 3: 80 + 80 = 160

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #80H	; <1>	80H	—	—	—
ADD	A, #80H	; <2>	00H	1	0	60H
ADD	A, !BCDADJ	; <3>	60H	1	0	—

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 - 52 = 39

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #91H	; <1>	91H	—	—	—
SUB	A, #52H	; <2>	3FH	0	1	06H
SUB	A, !BCDADJ	; <3>	39H	0	0	—

CHAPTER 28 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

28.1 Conventions Used in Operation List

28.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 28 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only Note) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only Note)
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Tables 3 - 5** to **3 - 7 Special Function Register (SFR) List**. The extended special function registers can be described to operand !addr16 as symbols. See **Tables 3 - 8** to **3 - 13 Extended Special Function Register (2nd SFR) List**.

28.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 28 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
⊻	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

28.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 28 - 3 Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

28.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.
A interrupt is not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 28 - 4 Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES: [HL]	11H	8BH	—	—	—

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

28.2 Operation List

Table 28 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	—	r ← byte			
		PSW, #byte	3	3	—	PSW ← byte	×	×	×
		CS, #byte	3	1	—	CS ← byte			
		ES, #byte	2	1	—	ES ← byte			
		!addr16, #byte	4	1	—	(addr16) ← byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16) ← byte			
		saddr, #byte	3	1	—	(saddr) ← byte			
		sfr, #byte	3	1	—	sfr ← byte			
		[DE+byte], #byte	3	1	—	(DE + byte) ← byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE) + byte) ← byte			
		[HL+byte], #byte	3	1	—	(HL + byte) ← byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL) + byte) ← byte			
		[SP+byte], #byte	3	1	—	(SP + byte) ← byte			
		word[B], #byte	4	1	—	(B + word) ← byte			
		ES:word[B], #byte	5	2	—	((ES, B) + word) ← byte			
		word[C], #byte	4	1	—	(C + word) ← byte			
		ES:word[C], #byte	5	2	—	((ES, C) + word) ← byte			
		word[BC], #byte	4	1	—	(BC + word) ← byte			
		ES:word[BC], #byte	5	2	—	((ES, BC) + word) ← byte			
		A, r Note 3	1	1	—	A ← r			
		r, A Note 3	1	1	—	r ← A			
		A, PSW	2	1	—	A ← PSW			
		PSW, A	2	3	—	PSW ← A	×	×	×
		A, CS	2	1	—	A ← CS			
		CS, A	2	1	—	CS ← A			
		A, ES	2	1	—	A ← ES			
		ES, A	2	1	—	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	—	(addr16) ← A			
		ES:!addr16, A	4	2	—	(ES, addr16) ← A			
		A, saddr	2	1	—	A ← (saddr)			
		saddr, A	2	1	—	(saddr) ← A			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 6 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, sfr	2	1	—	A ← sfr			
		sfr, A	2	1	—	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	—	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	—	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			
		[DE+byte], A	2	1	—	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE) + byte ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	—	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	—	A ← (SP + byte)			
		[SP+byte], A	2	1	—	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	—	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	—	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	—	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	—	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	—	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	—	((ES, BC) + word) ← A			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 7 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)			
		[HL+B], A	2	1	—	(HL + B) ← A			
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL+B], A	3	2	—	((ES, HL) + B) ← A			
		A, [HL+C]	2	1	4	A ← (HL + C)			
		[HL+C], A	2	1	—	(HL + C) ← A			
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL+C], A	3	2	—	((ES, HL) + C) ← A			
		X, !addr16	3	1	4	X ← (addr16)			
		X, ES:!addr16	4	2	5	X ← (ES, addr16)			
		X, saddr	2	1	—	X ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		B, saddr	2	1	—	B ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
		C, saddr	2	1	—	C ← (saddr)			
		ES, saddr	3	1	—	ES ← (saddr)			
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	—	A ←→ r			
		A, !addr16	4	2	—	A ←→ (addr16)			
		A, ES:!addr16	5	3	—	A ←→ (ES, addr16)			
		A, saddr	3	2	—	A ←→ (saddr)			
		A, sfr	3	2	—	A ←→ sfr			
		A, [DE]	2	2	—	A ←→ (DE)			
		A, ES:[DE]	3	3	—	A ←→ (ES, DE)			
		A, [HL]	2	2	—	A ←→ (HL)			
		A, ES:[HL]	3	3	—	A ←→ (ES, HL)			
		A, [DE+byte]	3	2	—	A ←→ (DE + byte)			
		A, ES:[DE+byte]	4	3	—	A ←→ ((ES, DE) + byte)			
		A, [HL+byte]	3	2	—	A ←→ (HL + byte)			
		A, ES:[HL+byte]	4	3	—	A ←→ ((ES, HL) + byte)			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 8 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	—	A ←→ (HL + B)			
		A, ES:[HL+B]	3	3	—	A ←→ ((ES, HL) + B)			
		A, [HL+C]	2	2	—	A ←→ (HL + C)			
		A, ES:[HL+C]	3	3	—	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	—	A ← 01H			
		X	1	1	—	X ← 01H			
		B	1	1	—	B ← 01H			
		C	1	1	—	C ← 01H			
		!addr16	3	1	—	(addr16) ← 01H			
		ES:!addr16	4	2	—	(ES, addr16) ← 01H			
		saddr	2	1	—	(saddr) ← 01H			
	CLRB	A	1	1	—	A ← 00H			
		X	1	1	—	X ← 00H			
		B	1	1	—	B ← 00H			
		C	1	1	—	C ← 00H			
		!addr16	3	1	—	(addr16) ← 00H			
		ES:!addr16	4	2	—	(ES,addr16) ← 00H			
		saddr	2	1	—	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	—	(HL + byte) ← X	×		×
		ES:[HL+byte], X	4	2	—	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	—	rp ← word			
		saddrp, #word	4	1	—	(saddrp) ← word			
		sfrp, #word	4	1	—	sfrp ← word			
		AX, rp Note 3	1	1	—	AX ← rp			
		rp, AX Note 3	1	1	—	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	—	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	—	(ES, addr16) ← AX			
		AX, saddrp	2	1	—	AX ← (saddrp)			
		saddrp, AX	2	1	—	(saddrp) ← AX			
		AX, sfrp	2	1	—	AX ← sfrp			
		sfrp, AX	2	1	—	sfrp ← AX			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 9 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	—	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	—	(HL) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE + byte)			
		[DE+byte], AX	2	1	—	(DE + byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	—	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	—	AX ← (SP + byte)			
		[SP+byte], AX	2	1	—	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	—	(B + word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	—	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	—	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	—	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	—	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	—	((ES, BC) + word) ← AX			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 10 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	—	BC ← (saddrp)			
		DE, saddrp	2	1	—	DE ← (saddrp)			
		HL, saddrp	2	1	—	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	—	AX ←→ rp			
	ONEW	AX	1	1	—	AX ← 0001H			
		BC	1	1	—	BC ← 0001H			
	CLRW	AX	1	1	—	AX ← 0000H			
		BC	1	1	—	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	—	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	—	A, CY ← A + r	×	×	×
		r, A	2	1	—	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, C ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the program memory area is accessed.

Note 3. Except rp = AX

Note 4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 11 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	—	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, rv Note 3	2	1	—	A, CY ← A + r + CY	×	×	×
		r, A	2	1	—	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, saddr	2	1	—	A, CY ← A + (saddr) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY ← A - byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r	×	×	×
		r, A	2	1	—	r, CY ← r - A	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A - ((ES, HL) + C)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 12 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	—	A, CY ← A - byte - CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte - CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r - CY	×	×	×
		r, A	2	1	—	r, CY ← r - A - CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16) - CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16) - CY	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr) - CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL) - CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL) - CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte) - CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte) - CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B) - CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B) - CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C) - CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A - ((ES:HL) + C) - CY	×	×	×
	AND	A, #byte	2	1	—	A ← A ∧ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	—	A ← A ∧ r	×		
		r, A	2	1	—	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 13 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	—	A ← A ∨ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	—	A ← A ∨ r	×		
		r, A	2	1	—	r ← r ∨ A	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∨ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	—	A ← A ⊻ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	1	—	A ← A ⊻ r	×		
		r, A	2	1	—	r ← r ⊻ A	×		
		A, !addr16	3	1	4	A ← A ⊻ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ⊻ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ⊻ (saddr)	×		
		A, [HL]	1	1	4	A ← A ⊻ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ⊻ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ⊻ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ⊻ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ⊻ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ⊻ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ⊻ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ⊻ ((ES:HL) + C)	×		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 14 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	—	A - byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) - byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) - byte	×	×	×
		saddr, #byte	3	1	—	(saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A - r	×	×	×
		r, A	2	1	—	r - A	×	×	×
		A, !addr16	3	1	4	A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A - (ES:addr16)	×	×	×
		A, saddr	2	1	—	A - (saddr)	×	×	×
		A, [HL]	1	1	4	A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A - (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A - ((ES:HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A - ((ES:HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A - ((ES:HL) + C)	×	×	×
	CMP0	A	1	1	—	A - 00H	×	0	0
		X	1	1	—	X - 00H	×	0	0
		B	1	1	—	B - 00H	×	0	0
		C	1	1	—	C - 00H	×	0	0
		!addr16	3	1	4	(addr16) - 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) - 00H	×	0	0
		saddr	2	1	—	(saddr) - 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X - (HL + byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X - ((ES:HL) + byte)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 15 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	—	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	—	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	—	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX + HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX + (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY ← AX - word	×	×	×
		AX, BC	1	1	—	AX, CY ← AX - BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX - DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX - HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX - ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX - word	×	×	×
		AX, BC	1	1	—	AX - BC	×	×	×
		AX, DE	1	1	—	AX - DE	×	×	×
		AX, HL	1	1	—	AX - HL	×	×	×
		AX, !addr16	3	1	4	AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX - ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	—	AX ← A × X			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 16 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumulate	MULU	X	1	1	—	AX ← A × X			
	MULHU		3	2	—	BCAX ← AX × BC (unsigned)			
	MULH		3	2	—	BCAX ← AX × BC (signed)			
	DIVHU		3	9	—	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	—	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	—	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	—	MACR ← MACR + AX × BC(signed)		×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Caution **Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.**

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 28 - 17 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	—	r ← r + 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) + 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) + 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	—	r ← r - 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) - 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) - 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1	×	×	
	INCW	rp	1	1	—	rp ← rp + 1			
		!addr16	3	2	—	(addr16) ← (addr16) + 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) + 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	—	rp ← rp - 1			
		!addr16	3	2	—	(addr16) ← (addr16) - 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) - 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1			
Shift	SHR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	—	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	—	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	—	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	—	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	—	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. cnt indicates the bit shift count.

Table 28 - 18 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX,1	2	1	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC,1	2	1	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
Bit manipulate	MOV1	CY, A.bit	2	1	—	CY ← A.bit			×
		A.bit, CY	2	1	—	A.bit ← CY			
		CY, PSW.bit	3	1	—	CY ← PSW.bit			×
		PSW.bit, CY	3	4	—	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	—	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	—	(saddr).bit ← CY			
		CY, sfr.bit	3	1	—	CY ← sfr.bit			×
		sfr.bit, CY	3	2	—	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	—	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	—	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	—	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	—	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∨ ∨ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 19 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	CY ← CY ⊻ bit			×
		CY, PSW.bit	3	1	—	CY ← CY ⊻ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ⊻ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	A.bit	2	1	—	A.bit ← 1			
		PSW.bit	3	4	—	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 1			
		saddr.bit	3	2	—	(saddr).bit ← 1			
		sfr.bit	3	2	—	sfr.bit ← 1			
		[HL].bit	2	2	—	(HL).bit ← 1			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	—	A.bit ← 0			
		PSW.bit	3	4	—	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 0			
		saddr.bit	3	2	—	(saddr.bit) ← 0			
		sfr.bit	3	2	—	sfr.bit ← 0			
		[HL].bit	2	2	—	(HL).bit ← 0			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 0			
	SET1	CY	2	1	—	CY ← 1			1
	CLR1	CY	2	1	—	CY ← 0			0
	NOT1	CY	2	1	—	CY ← $\overline{\text{CY}}$			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 20 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	CALL	rp	2	3	—	(SP - 2) ← (PC + 2)$_S$, (SP - 3) ← (PC + 2)$_H$, (SP - 4) ← (PC + 2)$_L$, PC ← CS, rp, SP ← SP - 4			
		$!addr20	3	3	—	(SP - 2) ← (PC + 3)$_S$, (SP - 3) ← (PC + 3)$_H$, (SP - 4) ← (PC + 3)$_L$, PC ← PC + 3 + jdisp16, SP ← SP - 4			
		!addr16	3	3	—	(SP - 2) ← (PC + 3)$_S$, (SP - 3) ← (PC + 3)$_H$, (SP - 4) ← (PC + 3)$_L$, PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	—	(SP - 2) ← (PC + 4)$_S$, (SP - 3) ← (PC + 4)$_H$, (SP - 4) ← (PC + 4)$_L$, PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	—	(SP - 2) ← (PC + 2)$_S$, (SP - 3) ← (PC + 2)$_H$, (SP - 4) ← (PC + 2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0000, addr5 + 1), PC$_L$ ← (0000, addr5), SP ← SP - 4			
	BRK	—	2	5	—	(SP - 1) ← PSW, (SP - 2) ← (PC + 2)$_S$, (SP - 3) ← (PC + 2)$_H$, (SP - 4) ← (PC + 2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0007FH), PC$_L$ ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	—	1	6	—	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), SP ← SP + 4			
	RETI	—	2	6	—	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	—	2	6	—	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 21 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	—	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2			
		rp	1	1	—	(SP - 1) ← rp$_H$, (SP - 2) ← rp$_L$, SP ← SP - 2			
	POP	PSW	2	3	—	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	—	rp$_L$ ← (SP), rp$_H$ ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	—	SP ← word			
		SP, AX	2	1	—	SP ← AX			
		AX, SP	2	1	—	AX ← SP			
		HL, SP	3	1	—	HL ← SP			
		BC, SP	3	1	—	BC ← SP			
		DE, SP	3	1	—	DE ← SP			
	ADDW	SP, #byte	2	1	—	SP ← SP + byte			
	SUBW	SP, #byte	2	1	—	SP ← SP - byte			
Unconditional branch	BR	AX	2	3	—	PC ← CS, AX			
		$addr20	2	3	—	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	—	PC ← PC + 3 + jdisp16			
		!addr16	3	3	—	PC ← 0000, addr16			
		!!addr20	4	3	—	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Note 1. Number of CPU clocks (f$_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f$_{CLK}$) when the program memory area is accessed.

Note 3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 28 - 22 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 3	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	—	2	1	—	Next instruction skip if CY = 1			
	SKNC	—	2	1	—	Next instruction skip if CY = 0			
	SKZ	—	2	1	—	Next instruction skip if Z = 1			
	SKNZ	—	2	1	—	Next instruction skip if Z = 0			
	SKH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SEL Note 4	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the program memory area is accessed.

Note 3. This indicates the number of clocks "when condition is not met/when condition is met".

Note 4. n indicates the number of register banks (n = 0 to 3)

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 29 ELECTRICAL SPECIFICATIONS

Caution 1. The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Caution 2. The pins mounted are as follows according to product.

29.1 Pins Mounted According to Product

29.1.1 Port functions

Refer to **2.1.1 30-pin products**, **2.1.2 32-pin products**, and **2.1.3 44-pin products**.

29.1.2 Non-port functions

Refer to **2.2.1 With functions for each product**.

29.2 Absolute Maximum Ratings

Absolute Maximum Ratings **(1/2)**

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to +2.8 and -0.3 to V_{DD} +0.3 $^{\text{Note 1}}$	V
Input voltage	V_{I1}	P00, P01, P10 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P121 to P124, P137, P146, P147, EXCLK, $\overline{\text{RESET}}$	-0.3 to V_{DD} +0.3 $^{\text{Note 2}}$	V
Output voltage	V_{O1}	P00, P01, P10 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	-0.3 to V_{DD} +0.3 $^{\text{Note 2}}$	V
Analog input voltage	V_{AI1}	ANI0 to ANI7, ANI16 to ANI19	-0.3 to V_{DD} +0.3 $^{\text{Notes 2, 3}}$ and -0.3 to AV_{REF} (+) +0.3	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Must be 6.5 V or lower.

Note 3. Do not exceed AV_{REF} (+) + 0.3 V in case of A/D conversion target pin.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. AV_{REF} (+): + side reference voltage of the A/D converter.

Remark 3. V_{SS}: Reference voltage

Absolute Maximum Ratings **(2/2)**

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	-40	mA
		Total of all pins -170 mA	P00, P01, P40, P41, P120	-70	mA
			P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P73, P146, P147	-100	mA
	I_{OH2}	Per pin	P20 to P27	-0.5	mA
		Total of all pins		-2	mA
Output current, low	I_{OL1}	Per pin	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	40	mA
		Total of all pins 170 mA	P00, P01, P40, P41, P120	70	mA
			P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P73, P146, P147	100	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			-65 to +150	°C

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

29.3 Oscillator Characteristics

29.3.1 X1 oscillator characteristics

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator**.

29.3.2 On-chip oscillator characteristics

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}		1		24	MHz
	f_{HOCO}		1		48	
High-speed on-chip oscillator clock frequency accuracy			-2		+2	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			-15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

29.4 DC Characteristics

29.4.1 Pin characteristics

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	2.7 V ≤ V_{DD} ≤ 5.5 V			-10.0 Note 2	mA
		Total of P00, P01, P40, P41, P120 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			-55.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			-10.0	mA
		Total of P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P73, P146, P147 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			-80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			-19.0	mA
		Total of all pins (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 5.5 V			-135.0	mA
	I_{OH2}	Per pin for P20 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V			-0.1 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 5.5 V			-1.5	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. Do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OH} = -10.0 mA
  Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Caution **P00, P10, P15, P17, P30, P50, P51 do not output high level in N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low [Note 1]	I_{OL1}	Per pin for P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147				20.0 Note 2	mA
		Total of P00, P01, P40, P41, P120 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			70.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
		Total of P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P73, P146, P147 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
		Total of all pins (When duty ≤ 70% [Note 3])				150.0	mA
	I_{OL2}	Per pin for P20 to P27				0.4 Note 2	mA
		Total of all pins (When duty ≤ 70% [Note 3])	2.7 V ≤ V_{DD} ≤ 5.5 V			5.0	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Note 2. However, do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OL} = 10.0 mA
  Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

  However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
  A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120 to P124, P146, P147	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P01, P10, P15 to P17, P30, P31, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	1.50		V_{DD}	V
	V_{IH3}	P20 to P27		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	EXCLK, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120 to P124, P146, P147	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P01, P10, P15 to P17, P30, P31, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27		0		0.3 V_{DD}	V
	V_{IL4}	EXCLK, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution The maximum value of V_{IH} of pins P00, P10, P15, P17, P30, P50, and P51 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 2.7 V $\leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	4.0 V $\leq V_{DD} \leq$ 5.5 V, I_{OH1} = -10.0 mA	V_{DD} - 1.5			V
			4.0 V $\leq V_{DD} \leq$ 5.5 V, I_{OH1} = -3.0 mA	V_{DD} - 0.7			V
			2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OH1} = -2.0 mA	V_{DD} - 0.6			V
			2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OH1} = -1.0 mA	V_{DD} - 0.5			V
	V_{OH2}	P20 to P27	2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OH2} = -100 μA	V_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	4.0 V $\leq V_{DD} \leq$ 5.5 V, I_{OL1} = 20.0 mA			1.3	V
			4.0 V $\leq V_{DD} \leq$ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OL1} = 0.3 mA			0.4	V
	V_{OL2}	P20 to P27	2.7 V $\leq V_{DD} \leq$ 5.5 V, I_{OL2} = 400 μA			0.4	V

Caution **P00, P10, P15, P17, P30, P50, and P51 do not output high level in N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P10 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P123, P124, P137, P146, P147, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00, P01, P10 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P123, P124, P137, P146, P147, $\overline{\text{RESET}}$	$V_I = V_{SS}$				-1	μA
	I_{LIL2}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{SS}$	In input port or external clock input			-1	μA
				In resonator connection			-10	μA
On-chip pull-up resistance	R_U	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P73, P120, P146, P147	$V_I = V_{SS}$, in input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

29.4.2 Supply current characteristics

(1) Flash ROM: 16 KB of 30- pin to 44-pin products

(T_A = -40 to +85°C, 2.7 V $\leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (high-speed main) mode Notes 3, 4	f_{HOCO} = 48 MHz, f_{IH} = 24 MHz	Basic operation	V_{DD} = 5.0 V		1.8		mA
						V_{DD} = 3.0 V		1.8		
			HS (high-speed main) mode Notes 3, 4	f_{HOCO} = 48 MHz, f_{IH} = 24 MHz	Normal operation	V_{DD} = 5.0 V		3.9	6.9	mA
						V_{DD} = 3.0 V		3.9	6.9	
				f_{HOCO} = 24 MHz, f_{IH} = 24 MHz	Normal operation	V_{DD} = 5.0 V		3.7	6.3	
						V_{DD} = 3.0 V		3.7	6.3	
				f_{HOCO} = 16 MHz, f_{IH} = 16 MHz	Normal operation	V_{DD} = 5.0 V		2.8	4.6	
						V_{DD} = 3.0 V		2.8	4.6	
			LS (low-speed main) mode Notes 3, 4	f_{IH} = 8 MHz	Normal operation	V_{DD} = 3.0 V		1.2	2.0	mA
			HS (high-speed main) mode Notes 2, 4	f_{MX} = 20 MHz, V_{DD} = 5.0 V	Normal operation	Square wave input		3.1	5.3	mA
						Resonator connection		3.3	5.5	
				f_{MX} = 20 MHz, V_{DD} = 3.0 V	Normal operation	Square wave input		3.1	5.3	
						Resonator connection		3.3	5.5	
				f_{MX} = 10 MHz, V_{DD} = 5.0 V	Normal operation	Square wave input		2.0	3.1	
						Resonator connection		2.0	3.2	
				f_{MX} = 10 MHz, V_{DD} = 3.0 V	Normal operation	Square wave input		2.0	3.1	
						Resonator connection		2.0	3.2	
			LS (low-speed main) mode Notes 2, 4	f_{MX} = 8 MHz, V_{DD} = 3.0 V	Normal operation	Square wave input		1.2	1.9	mA
						Resonator connection		1.2	2.0	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, comparator, programmable gain amplifier, watchdog timer, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors.

Note 2. When high-speed on-chip oscillator is stopped.

Note 3. When high-speed system clock is stopped.

Note 4. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (high speed main) mode: V_{DD} = 2.7 V to 5.5 V@1 MHz to 24 MHz
LS (low speed main) mode: V_{DD} = 2.7 V to 5.5 V@1 MHz to 8 MHz

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)

Remark 4. Temperature condition of the TYP. value is T_A = 25°C

(1) Flash ROM: 16 KB of 30-pin to 44-pin products

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode Notes 4, 6	f_{HOCO} = 48 MHz, f_{IH} = 24 MHz	V_{DD} = 5.0 V		0.60	2.40	mA
					V_{DD} = 3.0 V		0.60	2.40	
				f_{HOCO} = 24 MHz, f_{IH} = 24 MHz	V_{DD} = 5.0 V		0.40	1.83	
					V_{DD} = 3.0 V		0.40	1.83	
				f_{HOCO} = 16 MHz, f_{IH} = 16 MHz	V_{DD} = 5.0 V		0.38	1.38	
					V_{DD} = 3.0 V		0.38	1.38	
			LS (low-speed main) mode Notes 4, 6	f_{IH} = 8 MHz	V_{DD} = 3.0 V		260	710	μA
			HS (high-speed main) mode Notes 3, 6	f_{MX} = 20 MHz, V_{DD} = 5.0 V	Square wave input		0.28	1.55	mA
					Resonator connection		0.42	1.74	
				f_{MX} = 20 MHz, V_{DD} = 3.0 V	Square wave input		0.28	1.55	
					Resonator connection		0.42	1.74	
				f_{MX} = 10 MHz, V_{DD} = 5.0 V	Square wave input		0.19	0.86	
					Resonator connection		0.27	0.93	
				f_{MX} = 10 MHz, V_{DD} = 3.0 V	Square wave input		0.19	0.86	
					Resonator connection		0.27	0.93	
			LS (low-speed main) mode Notes 3, 6	f_{MX} = 8 MHz, V_{DD} = 3.0 V	Square wave input		95	550	μA
					Resonator connection		145	590	
	I_{DD3}	STOP mode Note 5	T_A = -40°C				0.18	0.51	μA
			T_A = +25°C				0.24	0.51	
			T_A = +50°C				0.29	1.10	
			T_A = +70°C				0.41	1.90	
			T_A = +85°C				0.90	3.30	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, comparator, programmable gain amplifier, watchdog timer, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors.

Note 2. During HALT instruction execution by flash memory.

Note 3. When high-speed on-chip oscillator is stopped.

Note 4. When high-speed system clock is stopped.

Note 5. When high-speed on-chip oscillator and high-speed system clock are stopped. When watchdog timer is stopped. The values below the MAX. column include the leakage current.

Note 6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (high speed main) mode: V_{DD} = 2.7 V to 5.5 V@1 MHz to 24 MHz
LS (low speed main) mode: V_{DD} = 2.7 V to 5.5 V@1 MHz to 8 MHz

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)

Remark 4. Temperature condition of the TYP. value is T_A = 25°C

(2) Peripheral Functions (Common to all products)

(T_A = -40 to +85°C, 2.7 V $\leq$ V_{DD} $\leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
12-bit interval timer operating current	I_{IT} Notes 1, 8					0.20		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2	f_{IL} = 15 kHz				0.22		μA
A/D converter operating current	I_{ADC} Note 3	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V			1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V			0.5	0.7	mA
A/D converter reference voltage current	I_{ADREF}					75		μA
Temperature sensor operating current	I_{TMPS}					75		μA
Comparator operating current	I_{CMP} Note 4	Per channel of comparator 1	When the comparator is operating			45.0	65.0	μA
			When the comparator is stopped			0.0	0.1	
Programmable gain amplifier operating current	I_{PGA} Note 5	When the programmable gain amplifier is operating				240.0	340.0	μA
		When the programmable gain amplifier is stopped				0.0	0.1	
LVD operating current	I_{LVI} Note 6					0.08		μA
SNOOZE operating current	I_{SNOZ}	ADC operation	The mode is performed Note 7			0.50	0.60	mA
			The A/D conversion operations are performed	Low voltage mode AV_{REFP} = V_{DD} = 3.0 V		1.20	1.44	mA
		Simplified SPI (CSI)/UART operation				0.70	0.84	mA

Note 1. When high speed on-chip oscillator and high-speed system clock are stopped.

Note 2. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontroller is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates in STOP mode.

Note 3. Current flowing only to the A/D converter. The current value of the RL78 microcontroller is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

Note 4. Current flowing only to the comparator. The current value of the RL78 microcontroller is the sum of I_{DD1} or I_{DD2} and I_{CMP} when the comparator operates in operating mode or HALT mode.

Note 5. Current flowing only to the programmable gain amplifier. The current value of the RL78 microcontroller is the sum of I_{DD1} or I_{DD2} and I_{PGA} when the programmable gain amplifier operates in operating mode or HALT mode.

Note 6. Current flowing only to the LVD circuit. The current value of the RL78 microcontroller is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

Note 7. For details on the transition time to SNOOZE mode, refer to **18.3.3 SNOOZE mode**.

Note 8. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontroller is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 3. Temperature condition of the TYP. value is T_A = 25°C

29.5 AC Characteristics

29.5.1 Basic operation

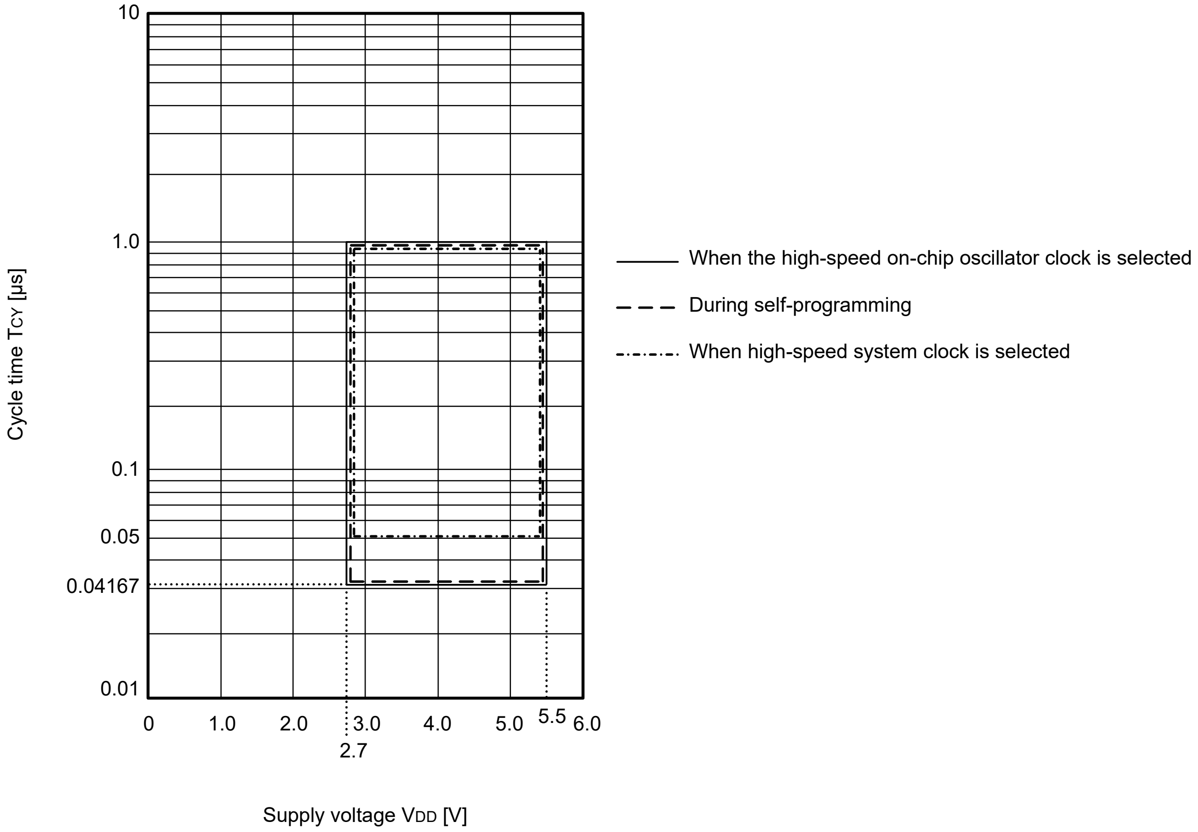
(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
			LS (low-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
		In the self programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
			LS (low-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns
Timer RJ input cycle	f_C	TRJIO		2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
Timer RJ input high-level width, low-level width	f_{WH}, f_{WL}	TRJIO		2.7 V ≤ V_{DD} ≤ 5.5 V	40			ns
TO00 to TO03, TRJIO0,TRJO, TRDIOA0/1, TRDIOB0/1, TRDIOC0/1,TRDIOD0/1 output frequency	f_{TO}	HS (high-speed main) mode		4.0 V ≤ V_{DD} ≤ 5.5 V			12	MHz
				2.7 V ≤ V_{DD} < 4.0 V			8	MHz
		LS (low-speed main) mode		2.7 V ≤ V_{DD} ≤ 5.5 V			4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		4.0 V ≤ V_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ V_{DD} < 4.0 V			8	MHz
		LS (low-speed main) mode		2.7 V ≤ V_{DD} ≤ 5.5 V			4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP5		2.7 V ≤ V_{DD} ≤ 5.5 V	1			μs
Key interrupt input low-level width	t_{KR}	KR0-KR3		2.7 V ≤ V_{DD} ≤ 5.5 V	250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

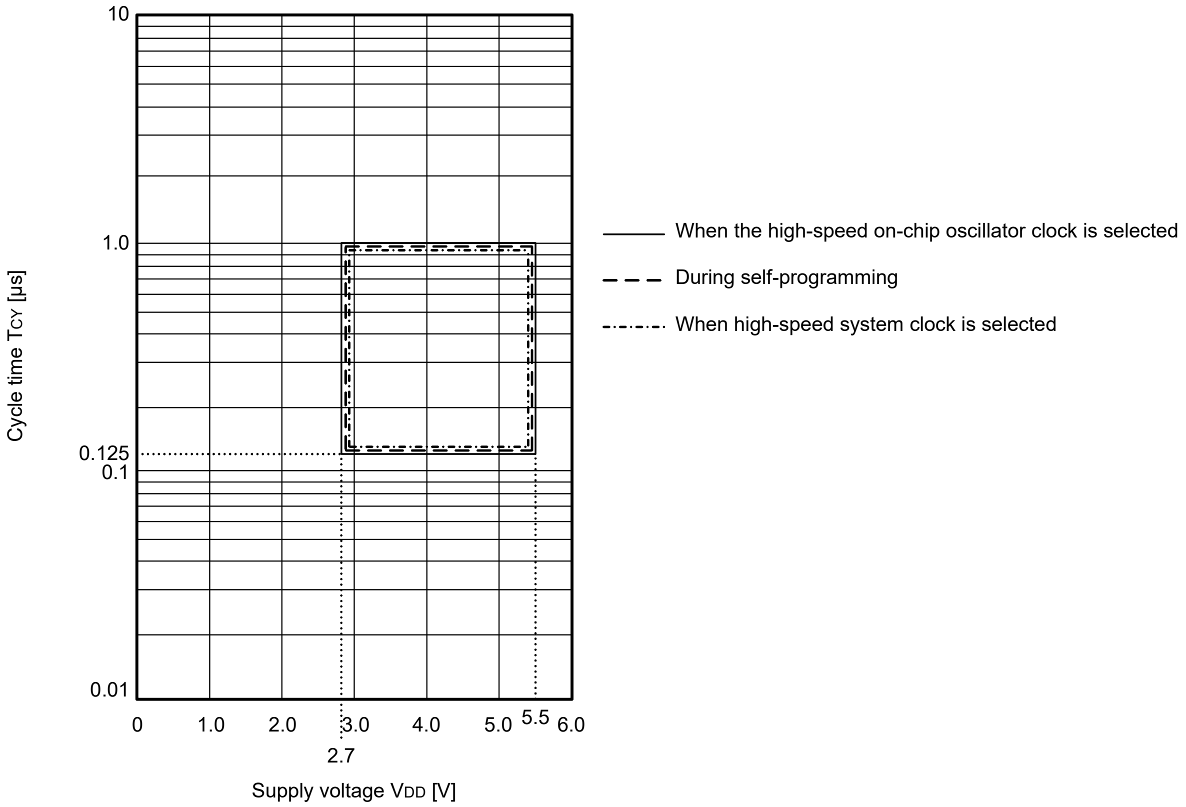
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 0 to 3))

Minimum Instruction Execution Time during Main System Clock Operation

TCY vs VDD (HS (high-speed main) mode)

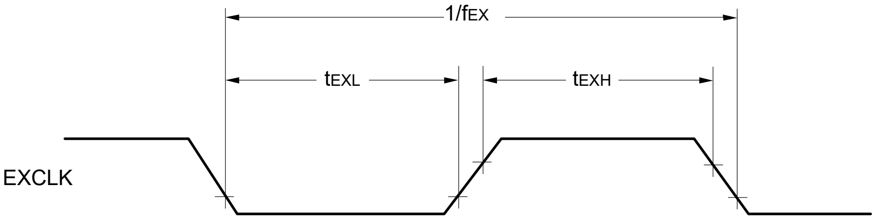


TCY vs VDD (LS (low-speed main) mode)

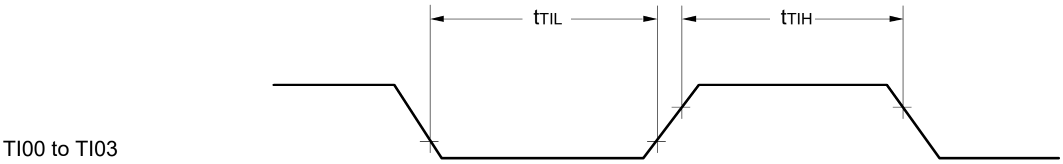


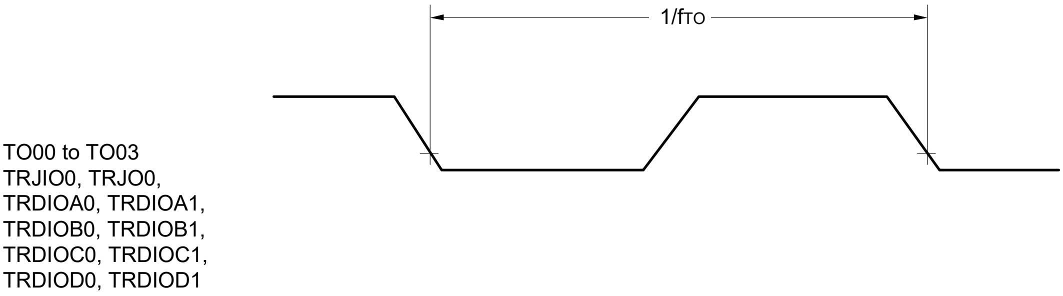
AC Timing Test Points

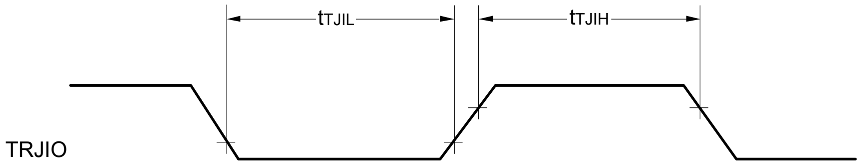
External System Clock Timing

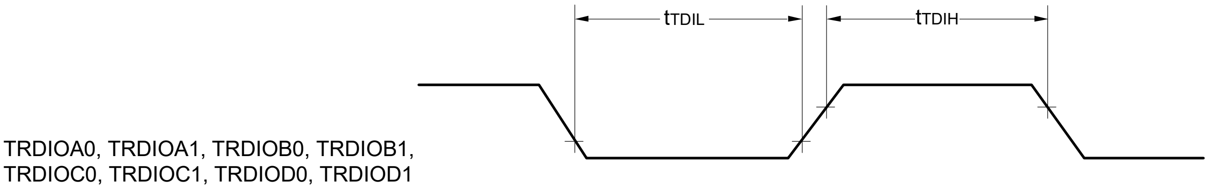


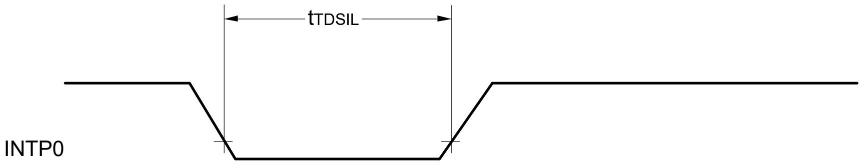
TI/TO Timing



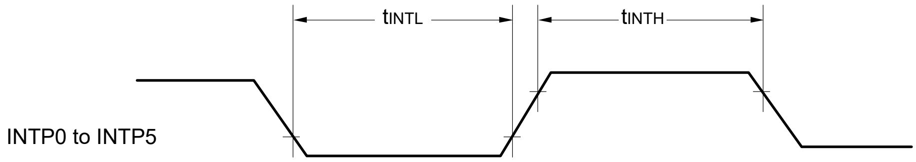



t_{TJIL}
t_{TJIH}
TRJIO

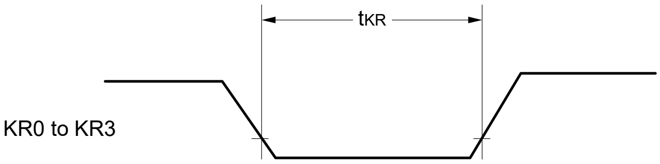

t_{TDIL}
t_{TDIH}
TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1,
TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1


t_{TDSIL}
INTP0

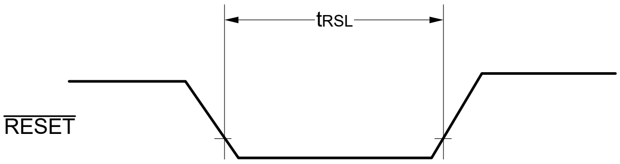
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



29.6 Peripheral Functions Characteristics

AC Timing Test Points

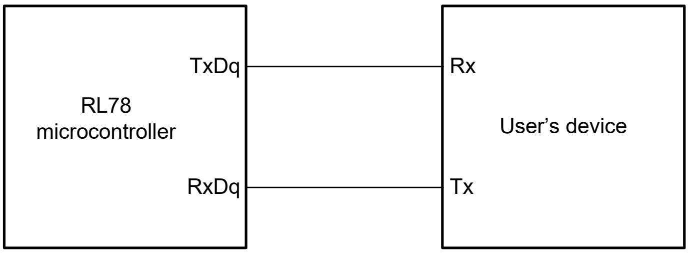
29.6.1 Serial array unit

(1) During communication at same potential (UART mode)

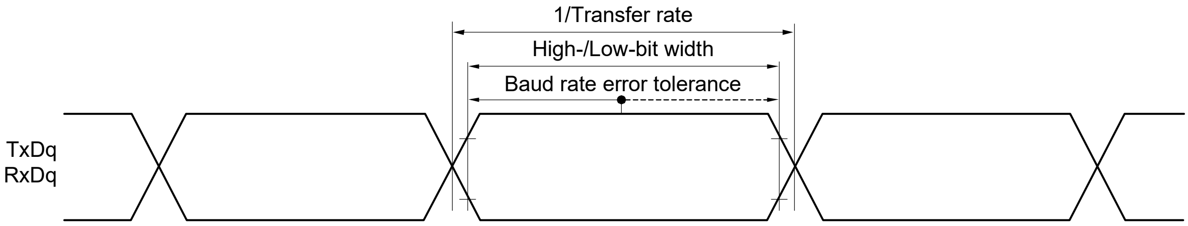
(T_A = -40 to +85°C, 2.7 V ≤ 5.5 V, Vss = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
Transfer rate [Note 1]		2.7 V ≤ V_{DD} ≤ 5.5 V		$f_{MCK}/6$		$f_{MCK}/6$	bps
		Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ [Note 2]		4.0		1.3	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.
However, the SNOOZE mode cannot be used when FRQSEL4 = 1.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 8 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03))

(2) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 2/f_{CLK}$	2.7 V ≤ V_{DD} ≤ 5.5 V	83.3		250		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ - 7		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ - 10		$t_{KCY1}/2$ - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		23		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		33		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 5.5 V		10		10		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 20 pF Note 4			10		10	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. This value is valid only when CSI00's peripheral I/O redirect function is not used.

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0),
g: PIM and POM numbers (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00))

(3) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	167		500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 12		t_{KCY1}/2 - 50		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 18		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		44		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		44		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 5.5 V		19		19		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 5.5 V C = 30 pF Note 4			25		25	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM number (g = 3, 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note 5	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		—		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 44		2/f_{MCK} + 110	ns
$\overline{\text{SSI00}}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	120		120		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 120		1/f_{MCK} + 120		ns
$\overline{\text{SSI00}}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 120		1/f_{MCK} + 120		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	120		120		ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SOp output lines.

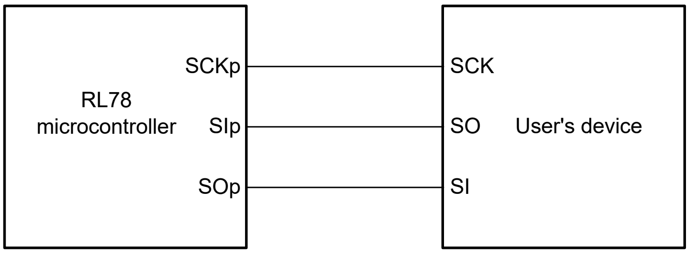
Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

Caution **Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

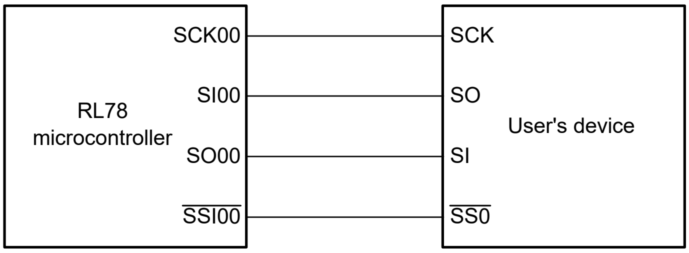
Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM number (g = 3, 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00))

Simplified SPI (CSI) mode connection diagram (during communication at same potential)



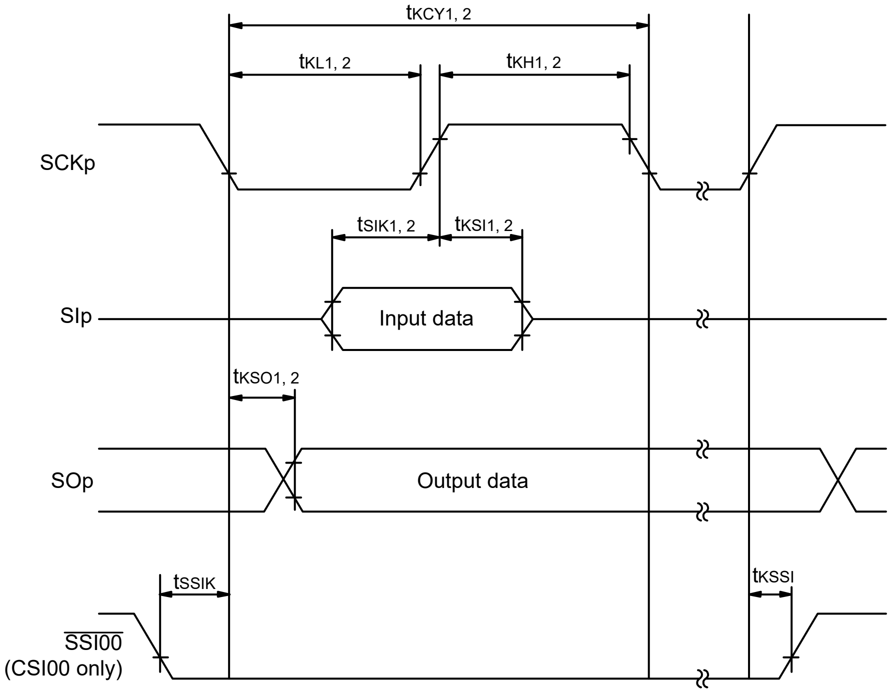
Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))



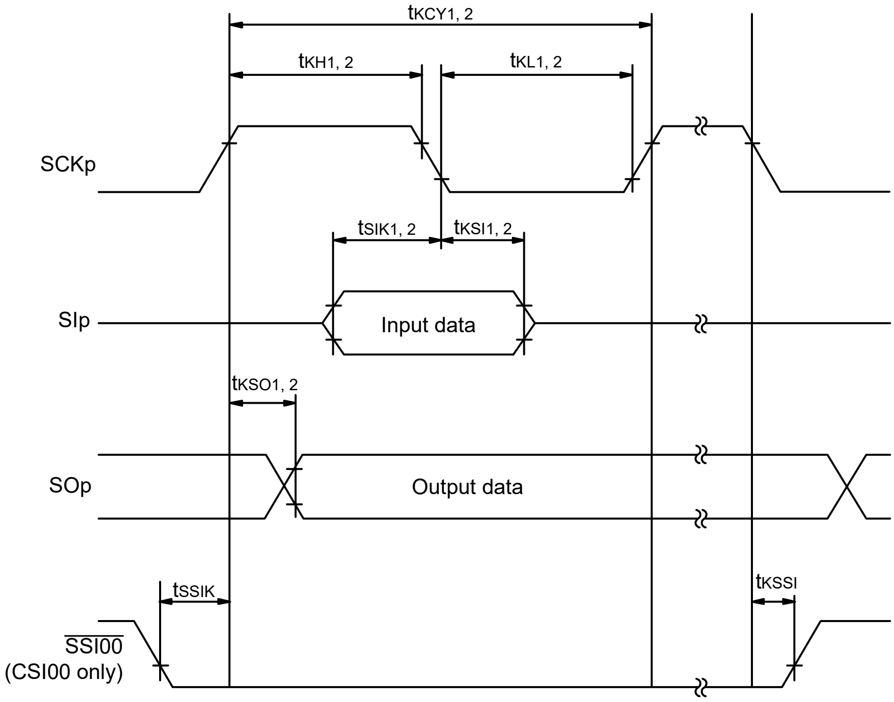
Remark 1. p: CSI number (p = 00)

Remark 2. m: Unit number, n: Channel number (mn = 00)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00)

Remark 2. m: Unit number, n: Channel number (mn = 00)

(5) During communication at same potential (simplified I^{2}C mode)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

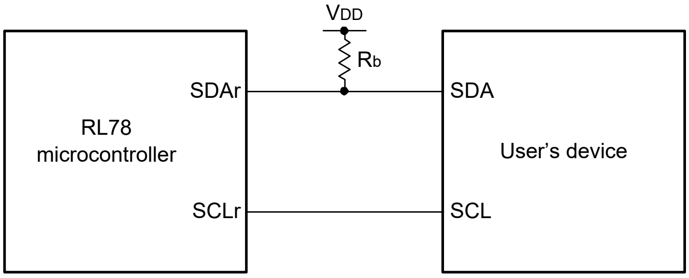
Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		400 Note 1	kHz
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1		400 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 85 Note 2		$1/f_{MCK}$ + 145 Note 2		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 Note 2		$1/f_{MCK}$ + 145 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	0	355	ns

Note 1. The value must also be equal to or less than $f_{MCK}/4$.

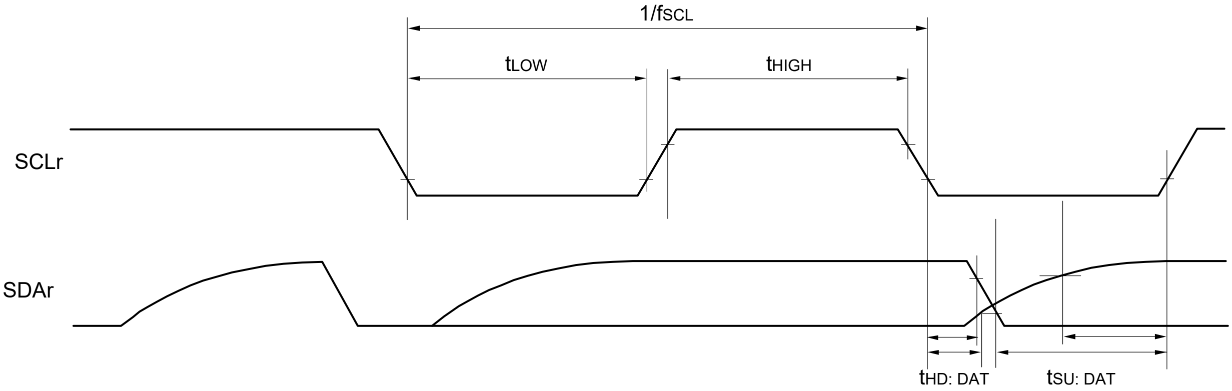
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

(Remarks are listed on the next page.)

Simplified I²C mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Caution **Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).**

Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00), g: PIM number (g = 3, 5), h: POM number (h = 3, 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0),
n: Channel number (n = 0), mn = 00)

(6) Communication at different potential (2.5 V, 3 V) (UART mode)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		4.0		1.3	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		4.0		1.3	Mbps
			2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 Notes 1, 2		f_{MCK}/6 Notes 1, 2	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		4.0		1.3	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.
However, the SNOOZE mode cannot be used when FRQSEL4 = 1.

Note 2. Use it with V_{DD} ≥ V_b.

Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 8 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03)

Remark 4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V
2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V: V_{IH} = 1.50 V, V_{IL} = 0.32 V

(6) Communication at different potential (2.5 V, 3 V) (UART mode)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions			HS (high-speed main) mode		LS (low-speed main) mode		Unit
					MIN.	MAX.	MIN.	MAX.	
Transfer rate		transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			Note 1		Note 1	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8 Note 2		2.8 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			Note 3		Note 3	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note 4		1.2 Note 4	Mbps
			2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Note 5, 6		Note 5, 6	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 7		0.43 Note 7	Mbps

Note 1. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

Note 3. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. Use it with V_{DD} ≥ V_b.

Note 6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when $2.7\text{ V} \leq V_{DD} < 3.3\text{ V}$ and $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 7. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

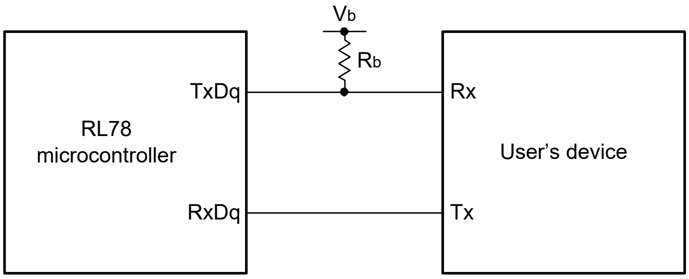
Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance,
C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 5)

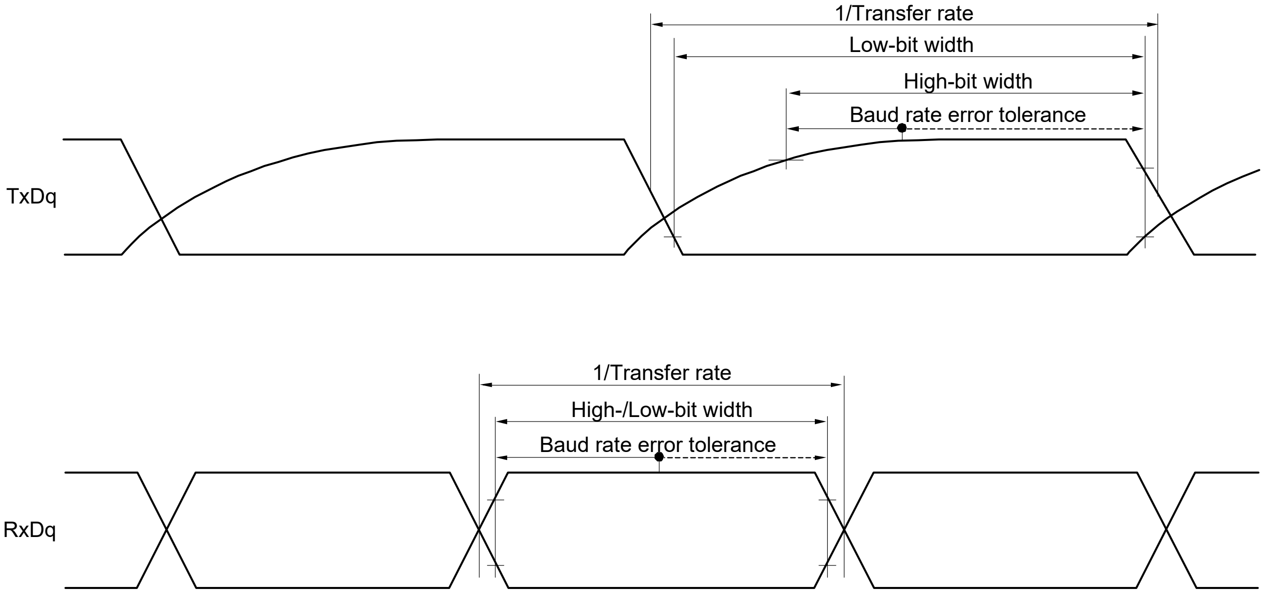
Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03))

Remark 4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}$: $V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}$: $V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
$2.7\text{ V} \leq V_{DD} < 3.3\text{ V}, 1.6\text{ V} \leq V_b \leq 2.0\text{ V}$: $V_{IH} = 1.50\text{ V}, V_{IL} = 0.32\text{ V}$

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 5)

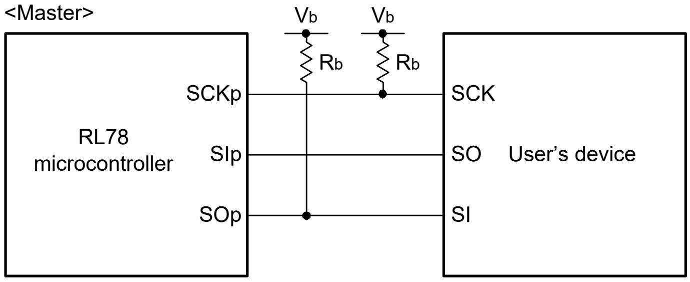
(7) Communication at different potential (2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 2/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 120		$t_{KCY1}/2$ - 120		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 7		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 10		$t_{KCY1}/2$ - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		23		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			10		10	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10		10	ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution **Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

Remark 3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified SPI (CSI) mode.

$4.0\ V \leq V_{DD} \leq 5.5\ V$, $2.7\ V \leq V_b \leq 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$

$2.7\ V \leq V_{DD} < 4.0\ V$, $2.3\ V \leq V_b \leq 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

Remark 4. This value is valid only when CSI00's peripheral I/O redirect function is not used.

(8) Communication at different potential (2.5 V, 3 V) (f_{MCK}/4) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		ns
			2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 458		t_{KCY1}/2 - 458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 12		t_{KCY1}/2 - 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 18		t_{KCY1}/2 - 50		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns

Caution 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Caution 2. Use it with $V_{DD} \geq V_b$.

Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

Remark 3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified SPI (CSI) mode.

4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

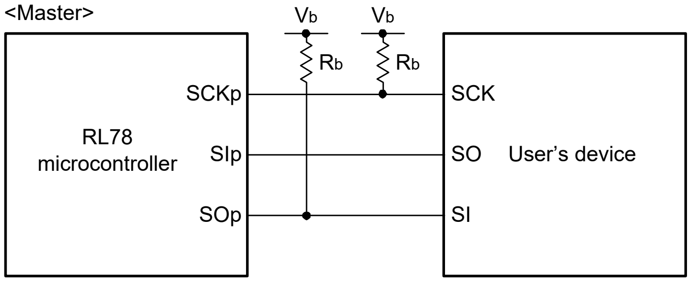
(8) Communication at different potential (2.5 V, 3 V) (f_{MCK}/4) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		479		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	19		19		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195	ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		483		483	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	110		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	19		19		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25	ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		25		25	ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Caution 2. Use it with $V_{DD} \geq V_b$.

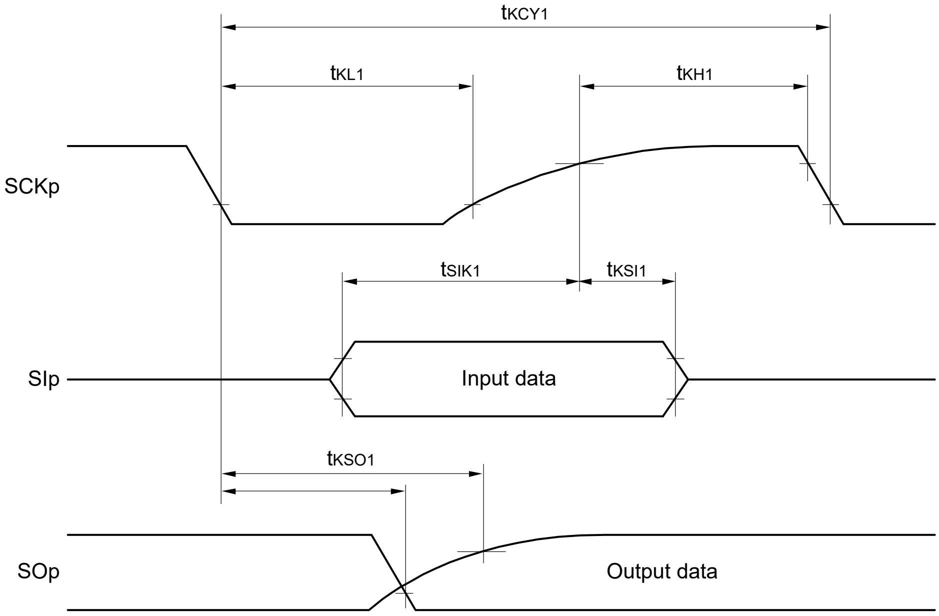
Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

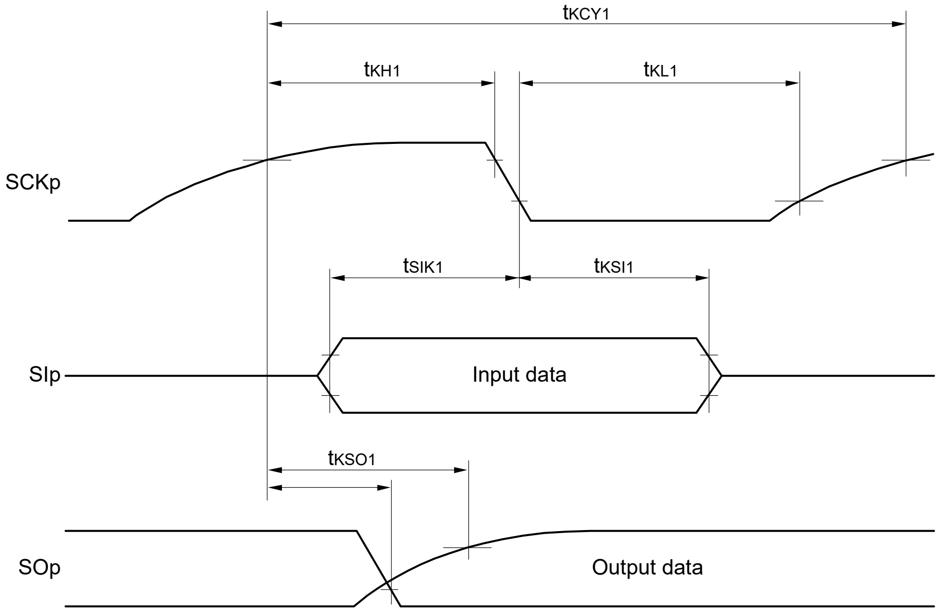
Remark 3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified SPI (CSI) mode.

4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

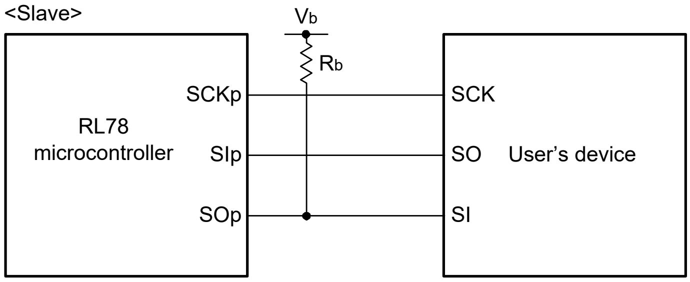
(9) Communication at different potential (2.5 V, 3 V) (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 - 12		t_{KCY2}/2 - 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 18		t_{KCY2}/2 - 50		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 120		2/f_{MCK} + 573	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 214		2/f_{MCK} + 573	ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_v = 5.5 kΩ			2/f_{MCK} + 573		2/f_{MCK} + 573	ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution **Select the TTL input buffer for the SIp pin and SCKp pin, and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance,
V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

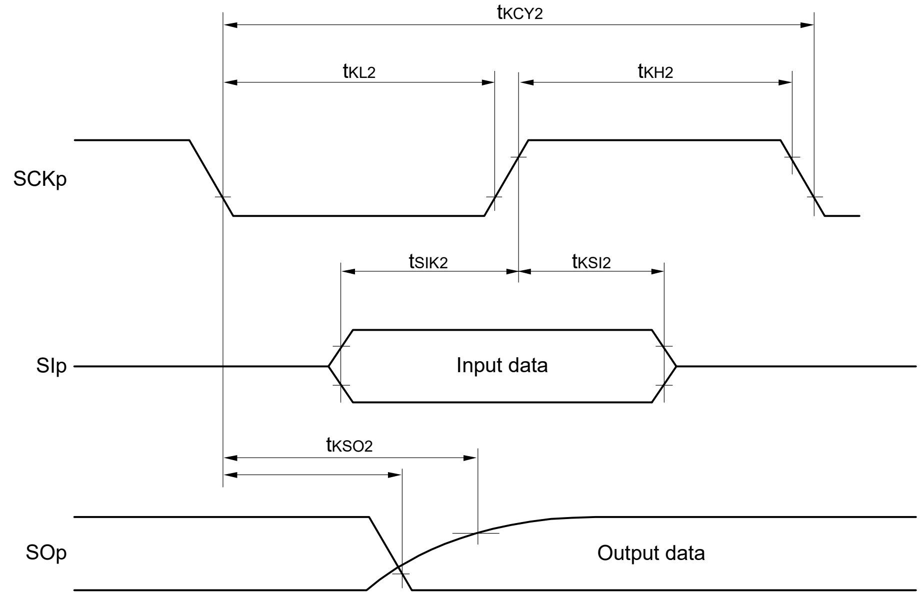
Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

Remark 4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified SPI (CSI) mode.

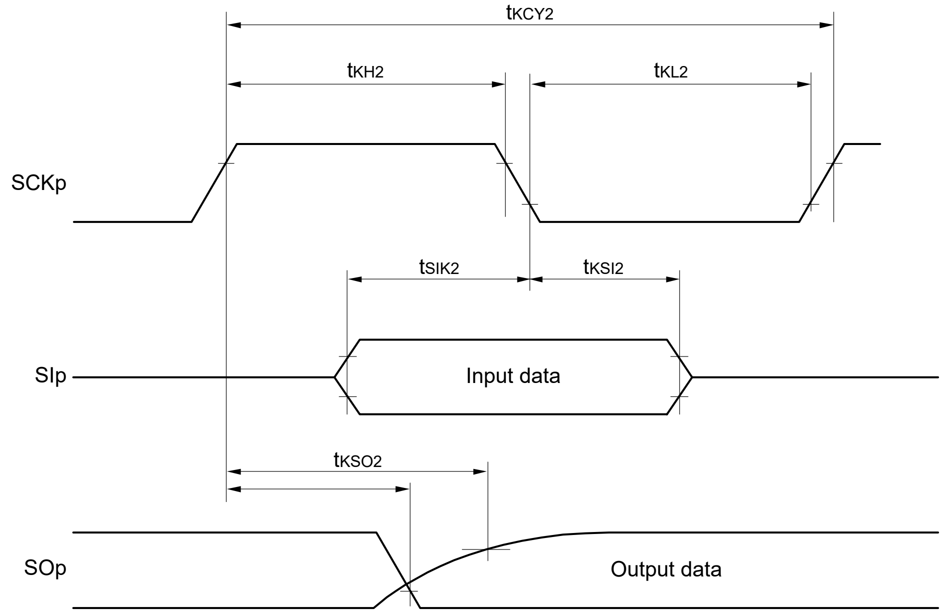
$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}, 2.7 \text{ V} \leq V_b \leq 4.0 \text{ V}: V_{IH} = 2.2 \text{ V}, V_{IL} = 0.8 \text{ V}$
$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}, 2.3 \text{ V} \leq V_b \leq 2.7 \text{ V}: V_{IH} = 2.0 \text{ V}, V_{IL} = 0.5 \text{ V}$

Remark 5. Communication at different potential cannot be performed during clock synchronous serial communication with the slave select function.

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

Remark 2. Communication at different potential cannot be performed during clock synchronous serial communication with the slave select function.

(10) Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		300 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		300 Note 1	kHz
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 Note 1		300 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note 1		300 Note 1	kHz
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b < 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		300 Note 1		300 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b < 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1550		1550		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		610		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		610		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		610		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		610		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b < 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	610		610		ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

(10) Communication at different potential (2.5 V, 3 V) (simplified I²C mode)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 Note 3		1/f_{MCK} + 190 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 Note 3		1/f_{MCK} + 190 Note 3		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b < 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	ns
		2.7 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b < 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	405	0	405	ns

Note 1. The value must also be equal to or less than f_{MCK}/4.

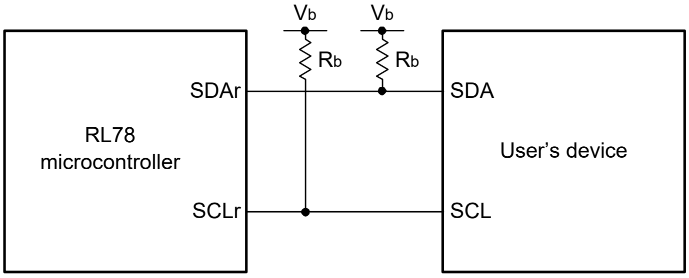
Note 2. Use it with V_{DD} ≥ V_b.

Note 3. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

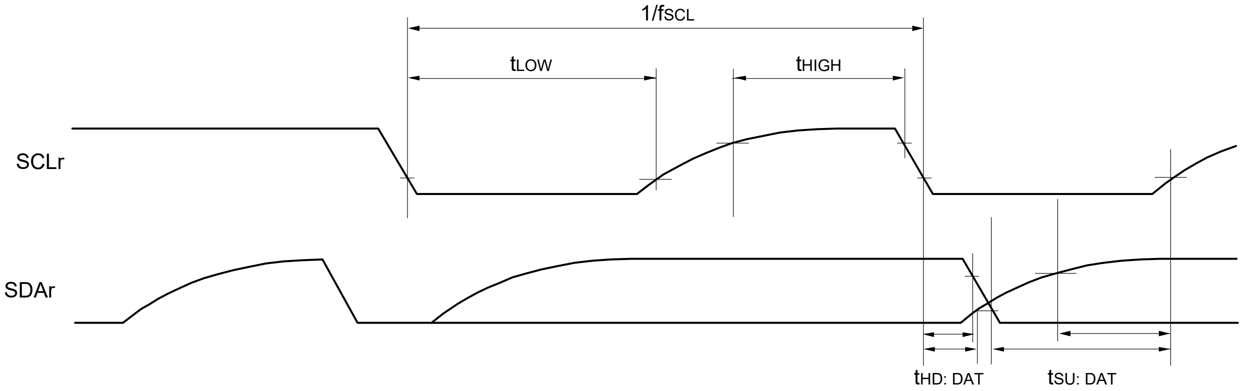
Caution **Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

(**Remarks** are listed on the next page.)

Simplified I2C mode connection diagram (during communication at different potential)



Simplified I2C mode serial transfer timing (during communication at different potential)



Remark 1. R_b[Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

Remark 2. r: IIC number (r = 00), g: PIM, POM number (g = 3, 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0), mn = 00)

Remark 4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^{2}C mode.

$4.0\ V \leq V_{DD} \leq 5.5\ V$, $2.7\ V \leq V_b \leq 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \leq V_{DD} < 4.0\ V$, $2.3\ V \leq V_b \leq 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

29.7 Analog Characteristics

29.7.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage Input channel	Reference voltage (+) = AVREFP Reference voltage (-) = AVREFM	Reference voltage (+) = VDD Reference voltage (-) = VSS	Reference voltage (+) = VBGR Reference voltage (-) = AVREFM
ANI0 to ANI7	Refer to **29.7.1 (1)**.	Refer to **29.7.1 (3)**.	Refer to **29.7.1 (4)**.
ANI16 to ANI19	Refer to **29.7.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **29.7.1 (1)**.		—

(1) When AVREF (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), AVREF (-) = AVREFM/ANI1 (ADREFM = 1), target ANI pin: ANI2 to ANI7

(TA = -40 to +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V		1.2	±3.5	LSB
Conversion time	tCONV	10-bit resolution AVREFP = VDD	3.6 V ≤ VDD ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ VDD ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±0.25	% FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±0.25	% FSR
Integral linearity error Note 1	ILE	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±2.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±1.5	LSB
Reference voltage (+)	AVREFP			2.7		VDD	V
Analog input voltage	VAIN			0		AVREFP	V
	VBGR	Select internal reference voltage output, 2.7 V ≤ VDD ≤ 5.5 V, HS (high-speed main) mode		1.38	1.45	1.5	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

(2) When AVREF (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), AVREF (-) = AVREFM/ANI1 (ADREFM = 1), target ANI pin: ANI16 to ANI19

(TA = -40 to +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V		1.2	±5.0	LSB
Conversion time	tCONV	10-bit resolution AVREFP = VDD	3.6 V ≤ VDD ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ VDD ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±0.35	% FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±0.35	% FSR
Integral linearity error Note 1	ILE	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution AVREFP = VDD	2.7 V ≤ VDD ≤ 5.5 V			±2.0	LSB
Reference voltage (+)	AVREFP			2.7		VDD	V
Analog input voltage	VAIN			0		AVREFP	V
	VBGR	Select internal reference voltage output, 2.7 V ≤ VDD ≤ 5.5 V, HS (high-speed main) mode		1.38	1.45	1.5	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

(3) When AV_{REF} (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), AV_{REF} (-) = V_{SS} (ADREFM = 0), target ANI pin: ANI0 to ANI7, ANI16 to ANI19

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.60	% FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.60	% FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI7		0		V_{DD}	V
		ANI16 to ANI19		0		V_{DD}	V
	V_{BGR}	Select internal reference voltage output, 2.7 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode		1.38	1.45	1.5	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

(4) When AVREF (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), AVREF (-) = AVREFM/ANI1 (ADREFM = 1), target ANI pin: ANI0 to ANI7, ANI16 to ANI19

(TA = -40 to +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V, Reference voltage (+) = VBGR, Reference voltage (-) = AVREFM = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES				8		bit
Conversion time	tCONV	8-bit resolution	2.7 V ≤ VDD ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	8-bit resolution	2.7 V ≤ VDD ≤ 5.5 V			±0.60	% FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.7 V ≤ VDD ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.7 V ≤ VDD ≤ 5.5 V			±1.0	LSB
Reference voltage (+)	VBGR			1.38	1.45	1.5	V
Analog input voltage	VAIN			0		VBGR	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

29.7.2 Temperature sensor characteristics

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Reference output voltage	V_{CONST}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			µs

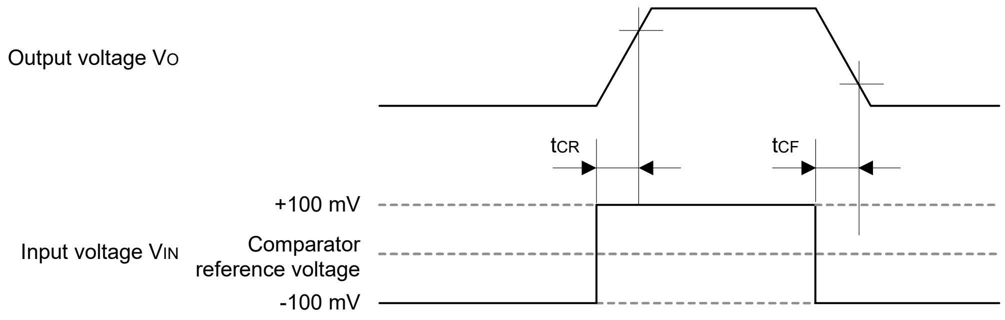
29.7.3 Comparator

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}				±5	±40	mV
Input voltage range	V_{ICMP}			0		V_{DD}	V
Internal reference voltage deviation	ΔV$_{IREF}$	CmRVM register value: 7FH to 80H (m = 0, 1)				±2	LSB
		Other than above				±1	LSB
Response time	t_{CR}, t_{CF}	Input amplitude = ±100 mV			70	150	ns
Operation stabilization time [Note 1]	t_{CMP}	CMPnEN = 0→1	V_{DD} = 3.3 to 5.5 V			1	µs
			V_{DD} = 2.7 to 3.3 V			3	
Reference voltage stabilization wait time	t_{VR}	CVRE: 0→1 [Note 2]				20	µs

Note 1. Time required after the operation enable signal of the comparator has been changed (CMPnEN = 0 → 1) until a state satisfying the DC and AC characteristics of the comparator is entered.

Note 2. Enable operation of internal reference voltage generation (CVREm bit = 1; m = 0, 1) and wait for the operation stabilization wait time before enabling the comparator output (CnOE bit = 1; n = 0, 1).



29.7.4 Programmable gain amplifier

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOPGA}				±5	±10	mV
Input voltage range	V_{IPGA}			0		0.9 × V_{DD}/gain	V
Response time	V_{OHPGA}			0.9 × V_{DD}			V
	V_{OLPGA}					0.1 × V_{DD}	
Gain error	—	4, 8 times				±1	%
		16 times				±1.5	
		32 times				±2	
Slew rate	SR_{RPGA}	Rising edge	4.0 V ≤ V_{DD} ≤ 5.5 V	1.4			V/μs
			2.7 V ≤ V_{DD} ≤ 4.0 V	0.5			
	SR_{FPGA}	Falling edge	4.0 V ≤ V_{DD} ≤ 5.5 V	1.4			
			2.7 V ≤ V_{DD} ≤ 4.0 V	0.5			
Operation stabilization wait time Note	t_{PGA}	4, 8 times				5	μs
		16, 32 times				10	

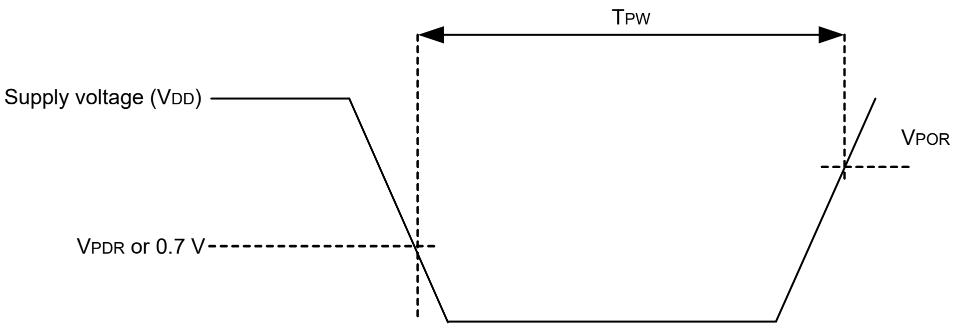
Note Time required after the PGA operation has been enabled (PGAEN = 1) until a state satisfying the DC and AC specifications of the PGA is entered.

29.7.5 POR circuit characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.47	1.51	1.55	V
	V_{PDR}	Power supply fall time	1.46	1.50	1.54	V
Minimum pulse width Note	t_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



29.7.6 LVD circuit characteristics

(T_A = -40 to +85°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.98	4.06	4.14	V
			Power supply fall time	3.90	3.98	4.06	V
		V_{LVD1}	Power supply rise time	3.68	3.75	3.82	V
			Power supply fall time	3.60	3.67	3.74	V
		V_{LVD2}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD3}	Power supply rise time	2.96	3.02	3.08	V
			Power supply fall time	2.90	2.96	3.02	V
		V_{LVD4}	Power supply rise time	2.86	2.92	2.97	V
			Power supply fall time	2.80	2.86	2.91	V
		V_{LVD5}	Power supply rise time	2.76	2.81	2.87	V
			Power supply fall time	2.70	2.75	2.81	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time		t_{LD}				300	μs

Remark $V_{LVD (n - 1)} > V_{LVDn}$: n = 1 to 5

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = -40 to +85°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage: 2.7 V			2.70	2.75	2.81	V
	V_{LVD4}		LVIS1, LVIS0 = 1, 0 (+0.1 V)	Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVD3}		LVIS1, LVIS0 = 0, 1 (+0.2 V)	Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V
	V_{LVD0}		LVIS1, LVIS0 = 0, 0 (+1.2 V)	Rising release reset voltage	3.98	4.06	4.14	V
				Falling interrupt voltage	3.90	3.98	4.06	V

29.7.7 Power supply voltage rising slope characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

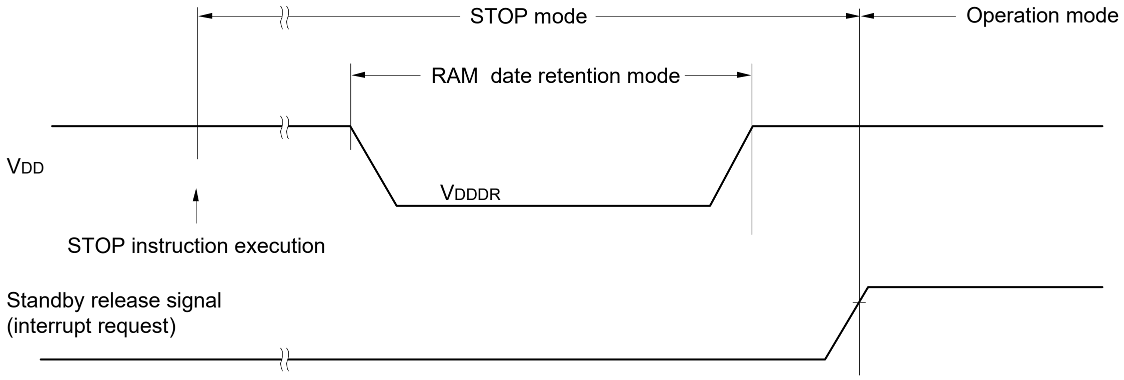
Caution **Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 29.5 AC Characteristics.**

29.8 RAM Data Retention Characteristics

(T_A = -40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



29.9 Flash Memory Programming Characteristics

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V		1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years	T_A = 85°C Note 3	1,000			Times

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite.
The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self programming library.

Note 3. These specifications show the characteristics of the flash memory and the results obtained from Renesas Electronics reliability testing.

Remark When updating data multiple times, use the flash memory as one for updating data.

29.10 Dedicated Flash Memory Programmer Communication (UART)

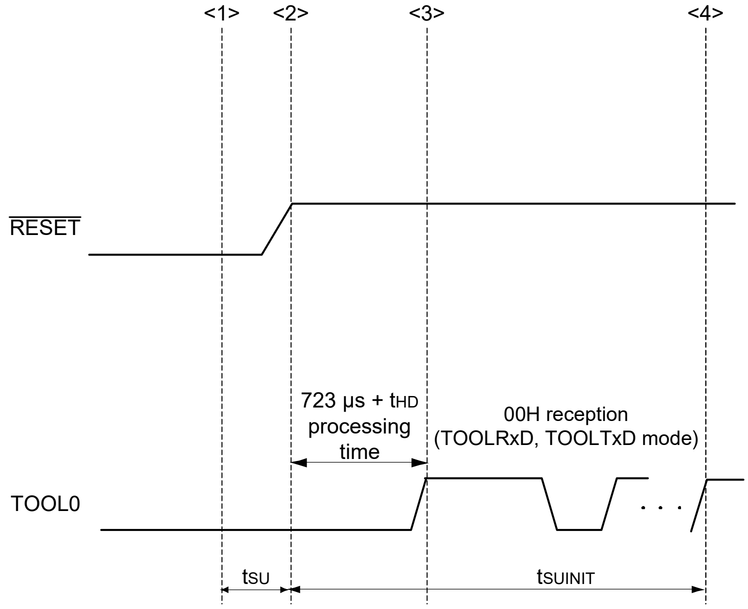
(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115.2 k		1 M	bps

29.11 Timing for Switching Flash Memory Programming Modes

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.

t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends

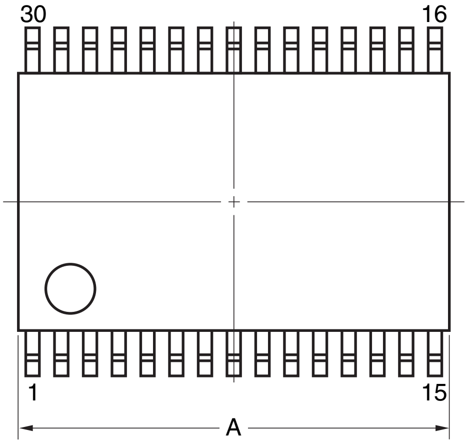
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded)

CHAPTER 30 PACKAGE DRAWINGS

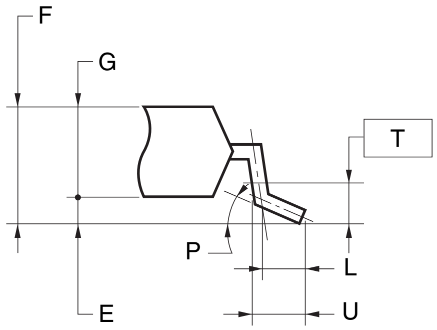
30.1 30-pin Products

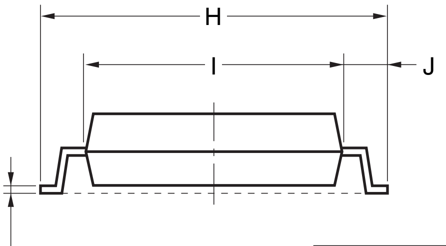
R5F11EA8ASP, R5F11EAAASP

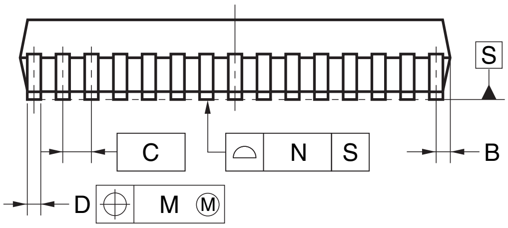
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



detail of lead end







NOTE

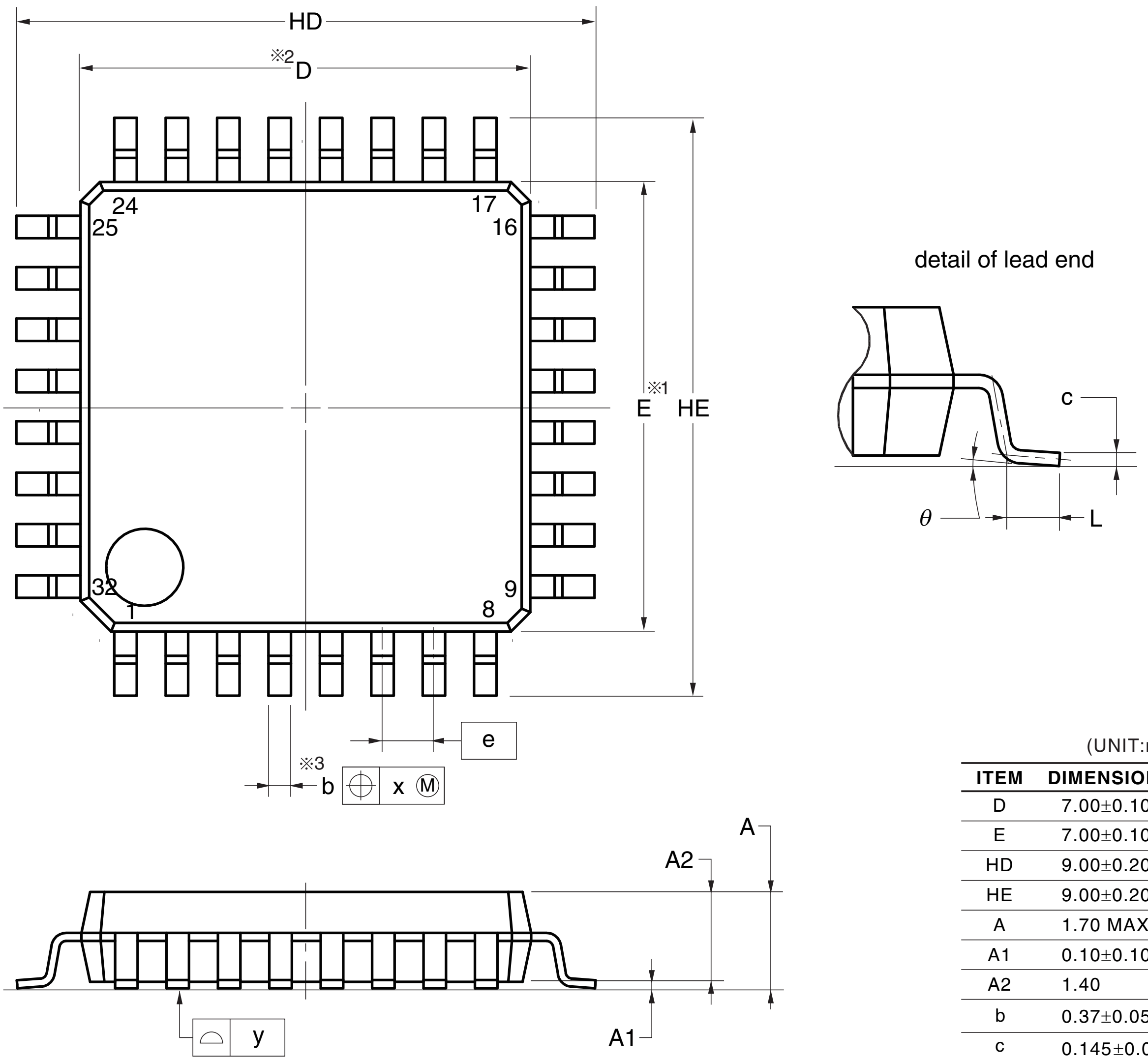
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3°^{+5°}_{-3°}$
T	0.25
U	0.6±0.15

30.2 32-pin Products

R5F11EB8AFP, R5F11EBAAFP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

NOTE

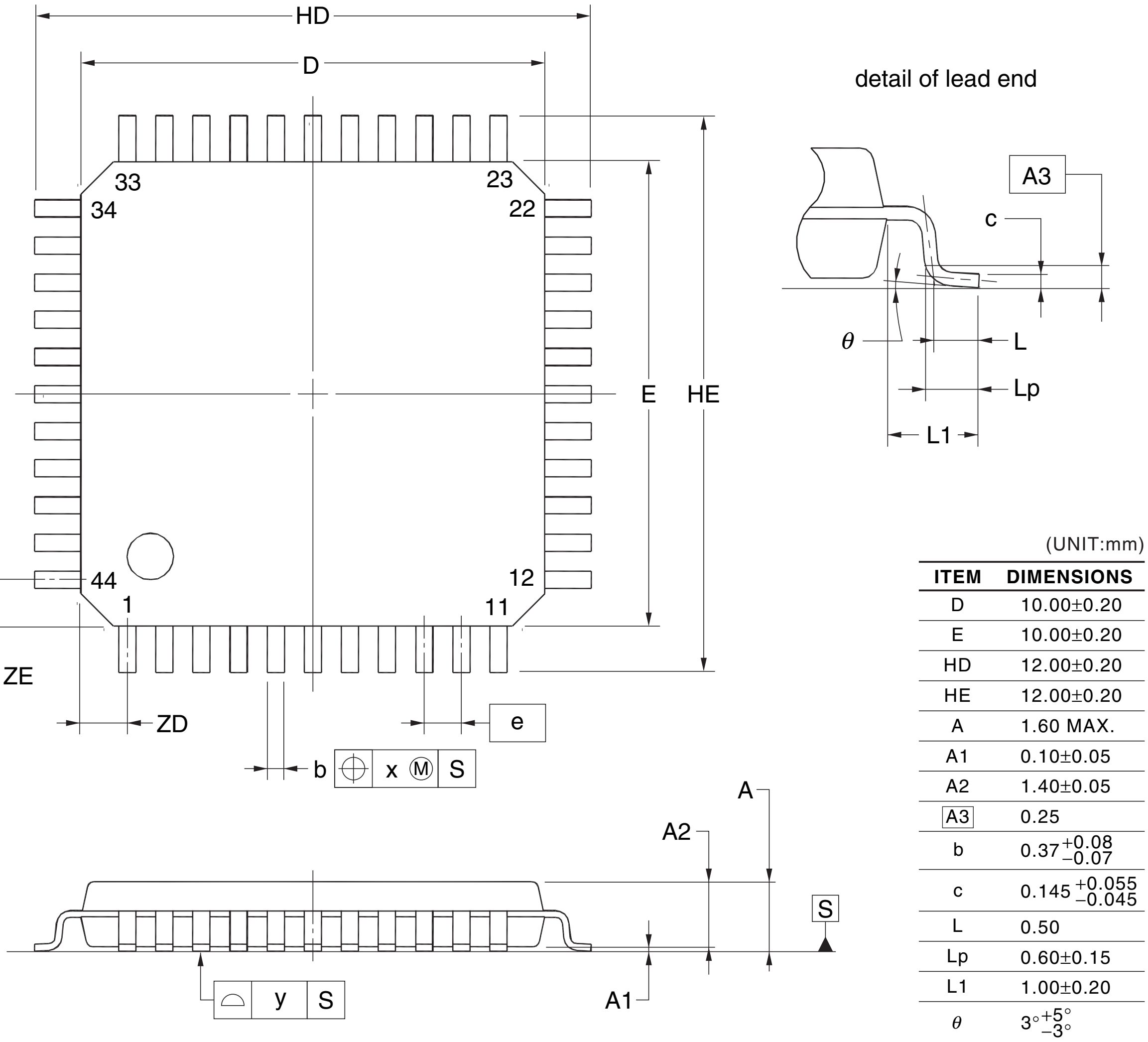
1.Dimensions “※1” and “※2” do not include mold flash.

2.Dimension “※3” does not include trim offset.

30.3 44-pin Products

R5F11EF8AFP, R5F11EFAAFP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP44-10x10-0.80	PLQP0044GC-A	P44GB-80-UES-2	0.36



(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.37^{+0.08}_{-0.07}$
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
e	0.80
x	0.20
y	0.10
ZD	1.00
ZE	1.00

NOTE
Each lead centerline is located within 0.20 mm of its true position at maximum material condition.

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/1)

Page	Description	Classification
CHAPTER 1 OUTLINE		
P.3	Modification of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G1G	(d)
P.3	Modification of Table 1 - 1 Orderable Part Numbers	(d)
CHAPTER 11 WATCHDOG TIMER		
P.406	Addition of Note in Table 11 - 4 Setting Window Open Period of Watchdog Timer	(c)
CHAPTER 22 SAFETY FUNCTIONS		
P.752	Modification of 22.1 Overview of Safety Functions	(c)
P.756	Modification of 22.4 CRC Operation Function (General-Purpose CRC)	(c)
P.760	Modification of 22.6 RAM Guard Function	(c)
P.761	Modification of 22.7 SFR Guard Function	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

(1/6)

Edition	Description	Chapter
Rev.1.31	“3-wire serial I/O” and “3-wire serial” were modified to “simplified SPI”.	All
	“CSI” was modified to “simplified SPI (CSI)”.	
	1.1 Features: Note 1 was added.	CHAPTER 1 OUTLINE
	1.2 List of Part Numbers: Ordering part numbers were added.	
	4.4.4 Handling different potential (2.5 V, 3 V) by using I/O buffers: Note was added.	CHAPTER 4 PORT FUNCTIONS
	CHAPTER 14 SERIAL ARRAY UNIT: Note was added.	CHAPTER 14 SERIAL ARRAY UNIT
	14.1.3 Simplified I^{2}C (IIC00): “Wait detection functions” was modified to “clock stretch detection”.	
	14.8 Operation of Simplified I^{2}C (IIC00) Communication: “Wait detection functions” was modified to “clock stretch detection”.	

(2/6)

Edition	Description	Chapter
Rev.1.30	Addition of Note 1 to 1.1 Features	CHAPTER 1 OUTLINE
	Modification of Pin configuration in 1.3.1 30-pin products	
	Modification of Pin configuration in 1.3.2 32-pin products	
	Modification of Pin configuration in 1.3.3 44-pin products	
	Modification of description of (1) Program counter in 3.2.1 Control registers	CHAPTER 3 CPU ARCHITECTURE
	Addition of Caution to Figure 4-1 Block Diagram of P00	CHAPTER 4 PORT FUNCTIONS
	Addition of Caution to Figure 4-2 Block Diagram of P01	
	Addition of Cautions 1 and 2 to Figure 4-3 Block Diagram of P10	
	Modification of Figure 4-7 Block Diagram of P14	
	Addition of Cautions 1 and 2 to Figure 4-8 Block Diagram of P15	
	Addition of Caution to Figure 4-9 Block Diagram of P16	
	Addition of Cautions 1 and 2 to Figure 4-10 Block Diagram of P17	
	Addition of Cautions 1 and 2 to Figure 4-12 Block Diagram of P30	
	Modification of Figure 4-13 Block Diagram of P31 and addition of Caution	
	Addition of Cautions 1 and 2 to Figure 4-16 Block Diagram of P50	
	Addition of Caution to Figure 4-17 Block Diagram of P51	
	Modification of Figure 5-1 Block DIagram of Clock Generator	CHAPTER 5 CLOCK GENERATOR
	Addition of description to Caution 5 of Figure 5-4 Format of Clock operation status control register (CSC)	
	Addition of Caution to 5.6.2 Example of setting X1 oscillation clock	
	Modification of Table 5-6 Changing CPU Clock	
	Addition of description to 5.6.6 Conditions before clock oscillation is stopped	
	Modification of Figure 6-12 Format of Timer mode register mn (TMRmn) (1/4)	CHAPTER 6 TIMER ARRAY UNIT
	Modification of Note 3 of Figure 6-15 Format of Timer mode register mn (TMRmn) (4/4)	
	Modification of description of (2) and (7) in 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)	
	Deletion of Caution from Figure 6-41 TO0n Pin Statuses bu Collective Manipulation of TO0n Bit	
	Modification of description in 6.8.2 Operation as external event counter	
	Modification of Figure 6-65 Operation Procedure When Input Signal High-/Low-level Width Measurement Function Is Used	
	Modification of Caution in 6.9.1 Operation as one-shot pulse output function	
	Addition of Note to Figure 6-75 Operation Procedure of One-Shot Pulse Output Function (2/2)	
	Modification of Figure 7-11 Operation Example in TImer Mode	CHAPTER 7 TIMER RJ
	Modification of Figure 7-12 Operation Example in Pulse Output Mode	
	Modification of Figure 7-13 Operation Example 1 in Event Counter Mode	
	Modification of Figure 7-15 Operation Example in Pulse Width Measurement Mode	
	Modification of Figure 7-16 Operation Example in Pulse Period Measurement Mode	

(3/6)

Edition	Description	Chapter
Rev.1.30	Addition of Figure 8-12 Format of Timer RD output control register (TRDOCR) [Reset Synchronous PWM Mode, Complementary PWM Mode]	CHAPTER 8 TIMER RD
	Modification of Figure 8-19 Format of Timer RD control register i (TRDCRi) (i = 0 or 1) [Complementary PWM Mode]	
	Modification of Figure 8-32 Format of Timer RD counter 0 (TRD0) [Reset Synchronous PWM Mode and PWM3 Mode]	
	Modification of Figure 8-33 Format of Timer RD counter 0 (TRD0) [Complementary PWM Mode (TRD0)]	
	Modification of Figure 8-34 Format of Timer RD counter 1 (TRD1) [Complementary PWM Mode (TRD1)]	
	Modification of Table 8-7 TRDGRji Register Functions in Complementary PWM Mode	
	Modification of Figure 8-46 Block Diagram of Pulse Output Forced Cutoff	
	Addition of 8.5 Timer RJ Operation	
	Modification of Figure 8-51 Block Diagram of Output Compare Function	
	Modification of Figure 8-55 Block Diagram of PWM Function	
	Modification of Figure 8-58 Block Diagram of Reset Synchronous PWM Mode	
	Modification of Figure 8-60 Block Diagram of Complementary PWM Mode	
	Modification of Figure 8-63 Block Diagram of PWM3 Mode	
	Modification of Figure 9-1 Block Diagram of 12-bit Interval Timer	CHAPTER 9 12-BIT INTERVAL TIMER
	Modification of Figure 12-31 Setting Up Software Trigger Mode	CHAPTER 12 A/D CONVERTER
	Modification of Figure 12-32 Setting Up Hardware Trigger No-Wait Mode	
	Modification of Figure 12-33 Setting Up Hardware Trigger Wait Mode	
	Modification of Figure 12-34 Setup when temperature sensor output voltage/internal reference voltage is selected	
	Modification of Figure 12-35 Setting Up Test Mode	
	Modification of Figure 12-39 Flowchart for Setting up SNOOZE Mode	
	Modification of description of (2) Input range of ANI0 to ANI7 and ANI16 to ANI19 pins in 12.10 Cautions for A/D Converter	
	Modification of Note 2 in 14.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	CHAPTER 14 SERIAL ARRAY UNIT
	Modification of Figure 14-20 Example of Reverse Transmit Data	
	Modification of description in 14.5.7 SNOOZE mode function	
	Change of Figure 14-73 Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0) and modification of Note 1	
	Modification of Figure 14-74 Flowchart of SNOOZE Mode Operation (Once Startup)	
	Modification of Note of Figure 14-75 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DSPmn = 0, CKPmn = 0)	
	Modification of Figure 14-76 Flowchart of SNOOZE Mode Operation (Continuous Startup)	
	Modification of Figure 14-112 Flowchart of UART Transmission (in Single-Transmission Mode)	
	Modification of description in 14.7.2 UART reception	
	Modification of Note of Figure 14-115 Example of Contents of Registers for UART Reception of UART (UART0, UAERT1) (1/2)	

(4/6)

Edition	Description	Chapter
Rev.1.30	Modification of Figure 14-119 Procedure for Resuming UART Reception	CHAPTER 14 SERIAL ARRAY UNIT
	Modification of description in 14.7.3 SNOOZE mode function and addition of Caution 5	
	Change of Figure 14-122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)	
	Change of Figure 14-123 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)	
	Modification of Figure 14-124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 =1, SSECm = 0)	
	Change of Figure 14-125 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Modification of Figure 14-126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Modification of Table 16-1 Interrupt Source List	CHAPTER 16 INTERRUPT FUNCTIONS
	Modification of Table 19-2 Hardware Statuses After Reset Acknowledgment (1/4)	CHAPTER 19 REST FUNCTION
	Modification of title of Figure 19-6 Example of Procedure for Checking Reset Source	
	Change of all	CHAPTER 20 POWER-ON-RESET CIRCUIT
	Change of all	CHAPTER 21 VOLTAGE DETECTOR
	Deletion of Note 2 from Figure 22-10 Format of Invalid memory access detection control register (IAWCTL)	CHAPTER 22 SAFETY FUNCTIONS
	Addition of description to (3) 000C2H in 24.1.1 User option byte (000C0H to 000C2H)	CHAPTER 24 OPTION BYTE
	Modification of Figure 24-4 Format of Option Byte (000C2H)	
	Modification of Table 25-1 Writing between RL78/G1G and Dedicated Flash Memory Programmer	CHAPTER 25 FLASH MEMORY
	Addition of Note to Table 26-1 On-Chip Debug Security ID	CHAPTER 26 ON-CHIP DEBUG FUNCTION
	Modification of title of 29.8 RAM Data Retention Characteristics	CHAPTER 29 ELECTRICAL SPECIFICATIONS
Rev.1.20	Change of description in 1.1 Features	CHAPTER 1 OUTLINE
	Change of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G1G	
	Change of Table 1 - 1 Orderable Part Numbers	
	Change of 1.6 Outline of Functions	
	Change of 3.1 Overview	CHAPTER 3 CPU ARCHITECTURE
	Change of description in 4.5 Register Setting for Used Port and Alternate Functions	CHAPTER 4 PORT FUNCTIONS
	Change of description in 8.1 Functions of Timer RD	CHAPTER 8 TIMER RD
	Change of description in 8.4.4 Pulse Output Forced Cutoff	
	Change of Figure 8 - 57 Block Diagram of Reset Synchronous PWM Mode	
	Change of Figure 8 - 58 Block Diagram of Complementary PWM Mode	

(5/6)

Edition	Description	Chapter
Rev.1.20	Change of Figure 12 - 4 Timing Chart When A/D Voltage Comparator Is Used	CHAPTER 12 A/D CONVERTER
	Change of Figure 12 - 12 Format of Analog input channel specification register (ADS) (1/2)	
	Change of Figure 12 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 12 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 12 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 12 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 26 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 12 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 12 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 12 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of description in 13.1 Overview	CHAPTER 13 COMPARATORS/PROGRAMMABLE GAIN AMPLIFIERS
	Change of Figure 13 - 5 Format of Comparator internal reference voltage control register (CVRCTL)	
	Change of Figure 13 - 6 Format of Comparator internal reference voltage selection register i (CiRVM)	
	Change of Table 13 - 3 Procedure for Setting Comparator Associated Registers	
	Change of description in 13.3.5 Using comparator output to forcibly cut off timer RD output	
	Deletion of Figure 13 - 17 Schematic Diagram of Timer RD Hi-Z Control Using Comparator Output	
	Addition of caution in 16.4.4 Interrupt servicing during division instruction	CHAPTER 16 INTERRUPT FUNCTIONS
	Change of description in 16.4.5 Interrupt request hold	
	Change of Table 18 - 2 Operating Statuses in STOP Mode	CHAPTER 18 STANDBY FUNCTION
	Addition of caution in Table 28 - 16 Operation List (12/18)	CHAPTER 28 INSTRUCTION SET

(6/6)

Edition	Description	Chapter
Rev.1.20	Change of 29.7.3 Comparator	CHAPTER 29 ELECTRICAL SPECIFICATIONS
Rev.1.10	Addition of the oscillation evaluation result.	CHAPTER 5 CLOCK GENERATOR
Rev.1.00	First Edition issued.	Throughout

RL78/G1G User’s Manual: Hardware

Publication Date: Rev. 1.40 Apr 26, 2024

Published by: Renesas Electronics Corporation

RL78/G1G

Renesas Electronics Corporation